RIJKSUNIVERSITEIT GRONINGEN

Vector Quantization based Learning Algorithms for Mixed Data Types and their Application in Cognitive Support Systems for Biomedical Research

Proefschrift

ter verkrijging van het doctoraat in de
Wiskunde en Natuurwetenschappen
aan de Rijksuniversiteit Groningen
op gezag van de
Rector Magnificus, dr. E. Sterken,
in het openbaar te verdedigen op
maandag 22 oktober 2012
om 11.00 uur

door

Waltraut Dietlind Zühlke

geboren op 17 november 1983
te Schkeuditz, Duitsland

Promotores:	Prof. dr. M. Biehl
	Prof. dr. T. Villmann
Beoordelingscommissie:	Prof. dr. N. Petkov
	Prof. dr. G. Bhanot
	Prof. dr. G. Palm

ISBN: 978-90-9027048-7

Contents

Acknowledgments

First of all I give my acknowledgments to the University of Groningen, Professor Michael Biehl and Professor Nicolai Petkov. I really appreciate the opportunity to conduct my PhD in your Intelligent Systems group at the Faculty of Mathematics and Natural Sciences of the University Groningen.

Thanks to the professors that agreed to become members of the reading committee: Professor Nicolai Petkov, Professor Gyan Bhanot and Professor Günther Palm.

Des Weiteren möchte ich mich beim Bundesministerium für Bildung und Forschung (BMBF) in Deutschland bedanken, welches das Exprimage-Projekt (Fkz 13N9873) – und damit auch zu großen Teilen meine Arbeit an der Dissertation – gefördert hat.

Ich danke dem Lehrstuhl Informatik V der RWTH Aachen, sowie dem Fraunhofer FIT, insbesondere Professor Matthias Jarke und Professor Thomas Berlage, die mir die Möglichkeit gaben, in und an dem Projekt zu wachsen, wissenschaftlich wie professionell. Ich danke allen Freunden, Kollegen, ehemaligen Kollegen und Mitarbeitern beim Fraunhofer FIT und bei Localite.

Mein Dank gilt Dr. Kerstin Röser für die pathologisch biologische Unterstützung bei der Konzeption und Validierung der verschiedenen Bausteine des Exprimage-Systems. Ich freu mich auf meinen nächsten Hamburg-Besuch bei dir. Ein Dank geht auch an die Pathologie Hamburg-West, die als pathologischer Partner in Exprimage die Proben aufbereitet und zur Verfügung gestellt haben.

Ich danke Professor Michael Biehl, dass er durch seine Unterstützung und Einwilligung die Durchführung der Promotion an der Universität in Groningen möglich gemacht hat. Michael, ich danke dir für die konstruktiven Gespräche, die wir auch schon hatten, lange bevor du mein “Promotor” warst. Du hast mir Mut gemacht, dass die Anwendung von “unseren” prototyp-basierten Verfahren im Kontext von echten Problemstellungen in der Medizin und Biologie Wissenschaft genug ist.

Mein Dank gilt Professor Barbara Hammer, die vor allem zu Beginn der Arbeit an der Promotion wichtige Denkanstöße und Präzisierungen in die ersten Vorstellungen

bezüglich einer geeigneten Thematik und deren Ausgestaltung gebracht hat. Barbara, ich danke dir für die offenen Gespräche, die mir im Laufe der Promotion immer wieder wichtige Orientierungshilfe waren.

Ganz besonders möchte ich mich bei Professor Thomas Villmann bedanken. Villy, du bist mit mir den langen und manchmal beschwerlichen Weg zu meiner Diss gegangen. Du wurdest nicht müde, mir meine Schwächen in der mathematischen Ausdrucksweise anzukreiden, sie mir zu verzeihen und sie gemeinsam mit mir auszumerzen. Deine Anregungen, Provokationen und Fragen haben mich und meine Arbeit wissenschaftlich reifen lassen. Und ganz nebenbei sind wir ziemlich gute Freunde geworden. Ich danke dir für alle Energie, Geduld und Zuneigung, die ich über die Zeit von dir für mich und meine Arbeit erfahren habe!

Ich danke meiner Disssis, Tina Geweniger, dass wir gemeinsam ein gutes Stück unseres Weges zur Dissertation gehen konnten! Danke für die langen und immer spannenden Gespräche in Nächten von Konferenzen, Workshops und Seminaren. Es war mir immer ein Vergnügen das Zimmer mit dir zu teilen. Ich wünsche dir, dass du dein Ziel weiterhin wissenschaftlich zu arbeiten umgesetzt bekommst und wünsche mir, dass wir noch oft in diesem Sinne zusammenarbeiten werden! Übrigens, auch vielen Dank an deine Familie, dass ihr mich immer wieder herzlich in eurem Haus als Gast aufgenommen habt.

Meinen "kleinen" Promotionsgeschwistern Sven Haase, Marika Kästner und Mandy Lange danke ich für die gute Zeit und wünsche Ihnen für ihre Diss alles Gute.

Ich danke meiner Familie, insbesondere meinen Eltern, Dr. Hubertus Zühlke und Gudrun Zühlke, und meiner Schwester, Dagmar Raab, mit ihrer Familie. Ihr habt mich immer wieder liebevoll energisch daran erinnert, dass das Leben nicht allein aus Wissenschaft besteht. Ich liebe meine beiden Neffen, Torben und Espen Raab, und ihre Überzeugungskraft, dass das Spielen mit ihnen wichtiger ist, als ein Titel ☺

Ein ganz lieber Dank geht an Sybille Grunst und Barbara Widua. Danke, dass wir unsere ganz eigene, jedoch nicht weniger liebevoll echte Familie sein können. Ich weiß, dass auch ihr zeitweise unter meiner Dissertation gelitten habt. Danke, dass ihr das ausgehalten und mir nicht nachgetragen habt!

Meinen Freundinnen Anna Schumacher, Janette Kawaschinski und Dr. Wiebke Hilbert danke ich, dass sie mich im richtigen Zeitraum vollkommen in Ruhe an meiner Arbeit haben schreiben lassen und jederzeit sofort für mich da waren, wenn ich das partout mal nicht wollte oder konnte. Ihr ward im echten Sinn immer für mich da.

Ein besonderer Dank gilt Jan Bornemeier, auf dessen außergewöhnlicher Diplomarbeit ich meine Analysen aufbauen durfte. Du hast mehr erreicht, als wir zu hoffen gewagt hatten und damit meine Ergebnisse möglich gemacht. Ich danke auch

Jans Freundin Joanna, dass sie Jan immer unterstützt hat und für unsere schönen gemeinsamen Abende.

Stefan Borbe, meinem Kollegen und gutem Freund, möchte ich aus tiefstem Herzen danken. Als mein Raumteiler hast du eine Menge der Höhen und Tiefen meiner Doktorarbeit hautnah mitbekommen und mir geholfen, sie durchzustehen. Ich wünsche dir Stefan, dass deine Dissertation in den nächsten zwei Jahren zu deiner Zufriedenheit gedeiht und zu einem guten Abschluss kommt.

Mit Dr. Hendrik Hoeth danke ich einem für mich sehr besonderen Menschen, der mich über die Zeit der Dissertation begleitet hat. Du hast mein Englisch immer wieder korrigiert und damit letztlich tatsächlich besser gemacht. Du hast in Phasen der intensiveren Kommunikation die Inhalte meiner Arbeit aus deiner fachlichen Sicht, der Teilchenphysik, hinterfragt und mich vorangebracht, was die Vermittlung meiner Arbeit betrifft. Gemeinsam mit unserem Freund Marc Schäfer hast du außerdem dafür gesorgt, dass die Dissertation auch ästhetisch ein kleines Schmuckstück geworden ist. Danke!

Dem wichtigsten Menschen in meinem Leben, Dr. Gernoth Grunst, kann ich nicht besser danken, als ich es mit dieser Dissertation tue. Sie ist unser Kind, das du, mein Wi, mit deinen Ideen und Überlegungen gezeugt hast. Du hast mir, als "werdender Mutter", immer wieder die Kraft gegeben, mit der ich das Kind ernähren und pflegen konnte. Auch, wenn ich schon längst hätte aufgeben wollen. Deine konstruktive Kritik, dein umfassendes konzeptuelles Urteilsvermögen und deine Sprachgewalt haben mich manchmal an den Rand der Verzweiflung gebracht. Mit deiner Hilfe habe ich gelernt dieser Verzweiflung nicht anheim zu fallen, sondern ihre Energie zum Größeren, zum Ausbau und zur Verwirklichung meiner Talente zu nutzen. Ich freu mich darauf, noch viele weitere "Kinder" mit dir zu bekommen.

Dietlind Zühlke
Bonn
August 27, 2012

Nomenclature

$[j]$ Indices of those feature dimensions belonging to the j^{th} feature group

$[v]_m$ Value of the feature of the m^{th} dimension of vector v

$[w]_{[j]}, [v]_{[j]}$ Feature components of prototype w and data point v respectively belonging to feature group j

$\cdot \circ \cdot$ Hadamard (entry-wise) product of two matrices

$\cdot^{\top}$ Transpose of a vector or matrix

ϵ_w, ϵ_α, ϵ_Λ Learning rate of the prototypes, the dissimilarity parameter vector $\vec{\alpha}$ and the dissimilarity parameter matrix Λ respectively, controlling the adaption strength

Λ, Λ_n Weighting factor matrix for dissimilarity functions, weighting factor matrix for dissimilarity functions with respect to prototype w_n

$\langle \cdot, \cdot \rangle$ Inner product of two vectors

$|\cdot|$ Cardinality of the set

$\mathbb{1}$ Column vector of all ones

$\mathbb{F}$ Embedding or feature space (Hilbert or Euclidean from context)

$\mathbb{I}$ Index set of the prototype set W

$\mathfrak{f}$ Fuzzifier according to Bezdek (Bezdek 1981) in Fuzzy C-Means

$\mathfrak{K}$ Mercer kernel matrix

$\mathfrak{k}_{\Phi}$ Mercer kernel function for the mapping function $\Phi(\cdot)$

$\mathfrak{V}$ Visualization space (for evaluation visualization)

μ_k, μ_{α}^{k}, μ_{Λ}^{k} Function expressing the relative difference distance for a data point v_k for the given model, relative difference distance for v_k and dissimilarity d_{α} or d_{Λ} respectively

ν Convergence threshold for relative cost function difference

Ψ, $\Psi^{\mathcal{F}}$ Crisp assignment function, fuzzy assignment function

$\psi_{w_n}(v_k)$, $\psi_{k,n}$ Assignment degree (possibility or probability) with which input v_k is represented by prototype w_n

Υ_n,$\tilde{\Upsilon}_{l,r}$,$\Upsilon_n^{\mathcal{F}}$ Receptive field of prototype w_n, intersection of the receptive fields of prototypes w_l and w_r, fuzzy receptive field of prototype w_n

$\vec{\alpha}$, $\vec{\alpha}_n$, α_j^n Weighting factor vector for dissimilarity functions, weighting factor vector for dissimilarity functions with respect to prototype w_n, single weighting factor for dissimilarity function d_j^n

$\vec{\beta}_n$, $[\beta_n]_k$ Coefficient vector for the vector representation X representing prototype $w_n^{\mathbb{F}}$, k^{th} component of the coefficient vector $\vec{\beta}_n$

$\vec{d}(k,n)$ Vector of dissimilarities $d_j(v_k, w_n)$ with $j = 1, \dots, J$

$\Xi_{\Psi(v_k)}(n)$ Characteristic function of the winner index $\Psi(v_k) = s$

ζ_{v_k}, $\vec{\zeta}_{v_k}$ Crisp class assignment for input vector $v_k \in V$, fuzzy class assignment for input vector $v_k \in V$ with $\vec{\zeta}_{v_k} = (\zeta_{v_k}(1), \dots, \zeta_{v_k}(C))$

$\zeta_{v_k}(c)$ Membership degree of input vector $v_k \in V$ corresponding to class $z_c \in Z$

C, c Number of classes, iteration index over the set of class labels $c = 1, \dots, C$

C_K Centering matrix of specific dimension K

D, $D^{\mathbb{F}}$ Dissimilarity matrix of data dissimilarities $D = (d_{k,l}) = (d(v_k, v_l))$, dissimilarity matrix in the embedding space

d, d_j^n Dissimilarity function, dissimilarity function for the j^{th} feature group with respect to prototype w_n

D_{α}, D_{Λ} Vector integrated overall dissimilarity and matrix integrated overall dissimilarity for mixed data, respectively

d_α, d_Λ Dissimilarity function parameterized by the vector $\vec{\alpha}$, dissimilarity function parameterized by the matrix Λ

$d^{\mathbb{F}}_{k,n}$ Dissimilarity between the projected data point $\Phi\left(v_k\right)$ and the prototype $w^{\mathbb{F}}_n$ in the embedding space $\mathbb{F}$

E, E_k Cost function, cost function according to the k^{th} data point

G_X Gram matrix for the vector representation X

h_σ Neighborhood function in Neural Gas based algorithms with range parameter σ^2

J, j Number of feature groups, iteration index of one of the J feature groups describing a data point or prototype

K, k Number of input feature vectors, iteration index over the input feature vector set $k = 1, \ldots, K$

K' Number of positive eigenvalues of G

L Sigmoid loss function in Generalized Learning Vector Quantization

M, m Dimension of the feature vectors, iteration index over the feature vector dimensions $m = 1, \ldots, M$

M' Dimensionality of the embedding space

N, n Number of prototypes, iteration index over the set of prototypes $n = 1, \ldots, N$

NEC Non-Euclidean coefficient (Pękalska and Duin 2009)

$p\left(v\right)$, $P\left(v\right)$ Probability density with $v \in V$, probability distribution with $v \in V$

$r_{k,n}$ Dissimilarity rank of prototype w_n with respect to input data point v_k, dissimilarity d and set of prototypes W

s, $s\left(v_k\right)$ Index of the winning prototype, index of the winning prototype according to data point v_k

V, V_{Tr} Set of input feature vectors with $|V| = K$, training data set with $V_{\text{Tr}} \subseteq V$

v, v_k Input vector, k^{th} of K data points (input vectors) $\in V$

W, $W^{\mathbb{F}}$ Set of prototype feature vectors, set of prototype vectors in an embedding space $\mathbb{F}$

w, w_n, w_+, w_- Prototype vector, n^{th} of N prototypes $\in W$, prototype vector of the best matching unit of the same class as the presented input, prototype vector of the best matching unit of a class different from the class of the presented input

$W^+_{v_k}$, $W^-_{v_k}$ Set of prototype vectors $w \in W$ that have the same class label as the input vector v_k, set of prototype vectors that have a different class label than the input vector

X Vector representation in the embedding space $\mathbb{F}$

Z Set of class labels with $|Z| = C$

z, z_c class labels, c^{th} of C possible class labels $\in Z$

z_{v_k} Crisp class label of data point v_k

z_{w_n}, $\vec{z}_{w_n}$ Crisp class label of prototype w_n, fuzzy class label of prototype w_n with $\vec{z}_{w_n} = (z_{w_n}(1), \ldots, z_{w_n}(C))$

$z_{w_n}(c)$ Membership degree of prototype $w_n \in W$ to class $z_c \in Z$

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

This thesis is embedded into a field of research and development that aims at systems for the cognitive support of modern research in medicine and biology. A current development in biology and medicine is the application of multi-layer models for the description of pathological objects and relations. For example to characterize a disease there are numerous, heterogeneous and growing clues in pathological research. Computational support via machine learning is needed as the number and heterogeneity of these clues is no longer cognitively manageable by human domain experts.

Concentrating on the medical insight interest the main task is the differentiation between diverse pathologies, i.e. subtypes of diseases. This is even more important regarding the upcoming personalized medicine. Human domain experts cognitively discriminate the pathological subtypes of a disease by means of typical representatives, i.e. patient cases. The computational concept of prototypes – as they are used in machine learning – presents a functional equivalent to these cognitive structures that seems to be adequate to built graspable disease models.

The typical patient cases are described by heterogeneous feature patterns mapping the multi-layer model of a patient's situation. For example in breast cancer research this includes the following feature groups:

- the status of the lymph nodes represented by a category,
- the distribution of inflammation over the different probe tissues,
- the heterogeneity of the expression level of a hormone receptor in different tumor regions.

A conceptual and mathematical valid measure of dissimilarity between different patient samples is needed. Often the suited measures are different for each of the feature groups. Hastie et al. state: "Although simple generic prescriptions for choosing the individual attribute dissimilarities [. . .] can be comforting, there is no substitute for careful thought in the context of each individual problem. Specifying an appropriate dissimilarity measure is far more important in obtaining success with clustering than choice of clustering algorithm. This aspect of the problem is emphasized less in the clustering literature than the algorithms themselves, since it depends on domain knowledge specifics and is less amenable to general research." (Hastie, T. et al. 2003, p. 506)

The relevance of the feature pattern is given by its power to discriminate different subtypes of the considered disease. Traditional prototype based methods are not able to provide computational support in these pathological research fields. They work on homogeneous vectors of describing features using one single similarity measure for all features. We extend these traditional prototype approaches for mixed data, i.e. for structured object descriptions that comprise heterogeneous feature groups with different dissimilarity measures.

Concerning the data in biomedical applications that is used for the training of the prototype based methods we face a twofold problem. We aim at differentiation between types within one disease but there is no biomedical knowledge about the number and characteristics of these different pathologies. Frequently suitable data for the conceptual discrimination into disease types are missing. In our application example, the breast cancer project Exprimage, the medical aim was the typing of breast tumors for the control of adjuvant therapies after tumor surgery. The information about the course of therapy was not given in the data. We had to approximate the typing by relying on the information about the clinical follow-up of the patients.

Another problem concerning the data is that often data is rare and the data sets in the applications are small and unrepresentative. From the statistical point of view these data sets are not suited for the generation of reliable models. To cope with this data-poor situation in the development of the prototype based methods we focus on the confirmation of interim results. This confirmation is provided in a twofold manner: by mathematical evaluation and by projecting the interim results back into the biomedical domain. This coupling to the domain experts enables the evaluation of the ecological validity[1] of the learned model. In this thesis we show a stepwise approach to reliable propositions on medical relations using the developed prototype based methods.

In the following chapter we introduce basic computational concepts, strategies, and necessities in the development of machine learning for cognitive support systems. Chapter 3 considers different dissimilarity measures that are suitable for comparing different feature groups as parts of an object description. The traditional prototype based algorithms are introduced in chapter 4.

In chapter 5 we introduce the framework of extended approaches that allows the handling of mixed data in prototype based methods. Our application scenario is described in chapter 6. Chapter 7 presents the results of the framework in the application scenario and explains the research process using the framework as cognitive

[1] Ecological validity is the extent to which research results can be applied to real life situations.

support system. In chapter 8 we discuss these results and draw corresponding conclusions. Additionally we will discuss mathematical as well as medical subsequent research aims that are based on the developed approaches.

Chapter 2

Basic concepts

This chapter has two objectives. The first is to introduce the underlying concepts of vector quantization and learning to the readers that do not have a background in this area. The second objective is to give the basic mathematical definitions corresponding to these basic concepts.

2.1 Nomenclature

For the first part of this chapter we assume that a set of real world objects is given. These objects are encoded as real-valued vectors $v_k \in V$, with $k \in \{1, \ldots, K\}$. The vectors are called *input vectors* and $V \subseteq \mathbb{R}^M$ is the *input vector set* where M is called the *dimension* of the vectors. The input vectors v_k are distributed according to the probability density $p(v)$, with $v \in V$. The probability distribution $P(v)$, with $v \in V$ is obtained from the probability density $p(v)$ by integrating. In most cases only examples for the input vectors are given and the probability density as well as the probability distribution are unknown.

The input vectors are also called points in the input space. In later sections and chapters an extended idea of object representations, going beyond simple real-valued vectors, is presented. We refer to these representations as *data points* instead of input vectors but still use v_k in the mathematical formulation. In each of the cases single dimensions of the object encoding are called *feature* or *attribute* of the object.

2.2 Crisp and fuzzy data representation using prototypes

In prototype based approaches the input vectors $v_k \in V \subseteq \mathbb{R}^M$ are approximated by a set of codebook or prototype vectors $w_n \in W = \{w_1, \ldots, w_N\} \subseteq \mathbb{R}^M$. The prototype vectors are usually defined over the same space as the input vectors. The traditionally *crisp assignment* is a *winner takes all rule*, mapping an input vector v_k

to the most similar prototype vector according to some (dis-)similarity measure $d\left(v_k, w_n\right)$:

$$\Psi : V \to \mathbb{I} : v_k \mapsto s = \arg\min_{n \in \mathbb{I}} \big(d\left(v_k, w_n\right)\big) \tag{2.2.1}$$

where $s = s\left(v_k\right)$ is called the *winner index* and $\mathbb{I} = \{1, \ldots, N\}$ is the index set of W. Instead of using the winner takes all rule, a *fuzzy assignment* from the input vector to membership degrees for all prototype vectors is given by:

$$\Psi^{\mathcal{F}} : V \to [0,1]^N : v_k \mapsto \vec{\psi}_W\left(v_k\right) = \big(\psi_{w_1}\left(v_k\right), \ldots, \psi_{w_N}\left(v_k\right)\big) \tag{2.2.2}$$

where $\psi_{w_n}\left(v_k\right) \in [0,1]$ is the possibility (or probability if $\sum_{n=1}^{N} \psi_{w_n}\left(v_k\right) = 1$) with which the input v_k is represented by prototype w_n. There are different definitions for this assignment function $\Psi^{\mathcal{F}}$, e.g. (Bezdek 1981) formulated this assignment function in Fuzzy C-Means algorithm for the determination of fuzzy membership degrees $\psi_{w_n}\left(v_k\right)$ as

$$\psi_{w_n}\left(v_k\right) = \frac{1}{\sum_{l=1}^{N} \left(\frac{d(v_k, w_n)}{d(v_k, w_l)}\right)^{\frac{2}{\mathfrak{f}-1}}} \tag{2.2.3}$$

including the fuzzifier value $\mathfrak{f}$. In the limit of $\mathfrak{f} \to 1$ the assignments become crisp. They can be either interpreted as probabilistic or as possibilistic. In crisp as well as in fuzzy assignment functions, the dissimilarity function $d = d\left(v_k, w_n\right)$ over the input vectors $v_k \in V$ and the prototype vectors $w_n \in W$ is often defined as the *Euclidean metric*.

Given the crisp assignment function in equation (2.2.1) the *crisp receptive field* of a prototype w_n is defined as those vectors in the input vector space V for which the winner is w_n according to the mapping function Ψ. This is formalized as:

$$\Upsilon_n = \big\{v \in V : \Psi\left(v\right) = n\big\}. \tag{2.2.4}$$

Two crisp receptive fields are called neighboring if their intersection is not empty, i.e. $\tilde{\Upsilon}_{l,r} = \Upsilon_l \cap \Upsilon_r \neq \emptyset$. Then $\tilde{\Upsilon}_{l,r}$ is a hyperplane in $\mathbb{R}^M$ according to the dissimilarity measure d. Furthermore, if d is the Euclidean metric all receptive fields are convex and the border of a receptive field is piecewise linear. Using the fuzzy assignment function in equation (2.2.2), a *fuzzy receptive field* of a prototype w_n is defined as:

$$\Upsilon_n^{\mathcal{F}} = \big\{v_k \in V : \psi_{w_n}\left(v_k\right) > 0\big\} \tag{2.2.5}$$

If not stated otherwise we always refer to crisp receptive fields when using the term receptive fields.

It is preferable to make fuzzy assignments in situations where the data are known to be overlapping as every crisp decision in such a situation would be artificial. Also in applications where the assignment is of interest for data analysis and exploration, fuzzy approaches can be more informative and thus more suitable.

2.3 Classification and clustering

This section is based on the standard work "Pattern Recognition and Machine Learning" of Bishop (Bishop 2007), its definitions and citations are used. Exceptions are clearly marked.

2.3.1 Basic definition for classification

Assume a set of *class labels* $Z = \{z_1, \ldots, z_C\}$. The vectorial crisp classification problem aims at assigning to each input vector $v_k \in V$ one $\zeta_{v_k} \in Z$. In vectorial fuzzy classification to each input vector $v_k \in V$ a vector of membership degrees $\vec{\zeta}_{v_k} = \left(\zeta_{v_k}(1), \ldots, \zeta_{v_k}(C)\right)$ is assigned with $\zeta_{v_k}(c) \in [0,1]$ giving the membership degree of $v_k \in V$ to the corresponding class $z_c \in Z$.

For classification problems two stages can be distinguished: the inference and the decision stage. In the inference stage the training data set V is used to build a model for the discrimination, which in the decision stage is used for creating class assignments for unlabeled vectors. There are at least two main kinds of models for discrimination:

1. *Class typical models*: They explicitly model typical properties of the different classes that e.g. in vector quantization approaches are represented by a prototypical vector. Learning class typical models often involves modeling the class-conditional probability densities.

2. *Class discriminating models*: They emphasize properties that discriminate the classes.

Examples for class typical models are Markov Models (MM) and Learning Vector Quantizers (LVQ). Linear Discriminant Analysis (LDA) and derivatives belong to the class discriminating models as well as decision trees and Support Vector Machines (SVM).

In prototype based classification approaches the prototypes w_n carry class labels that are either crisp or fuzzy. The crisp class label of a prototype w_n is referred to as $z_{w_n} \in Z$. The fuzzy class labels of the prototypes are vectors of class memberships $\vec{z}_{w_n} = (z_{w_n}(1), \ldots, z_{w_n}(C))$ with $z_{w_n}(c) \in [0,1]$ being the class membership degree for prototype w_n to class z_c. In the probabilistic variant of fuzzy classification the class memberships sum up to one, which is not necessarily the case in the possibilistic variant.

In crisp classification often the prototype class labels and the assignment function Ψ are crisp. The class label z_{w_s} of the winner prototype $w_s : s = \Psi(v_k)$ is used as class prediction ζ_{v_k} for input vector v_k and thus

$$\zeta_{v_k}(Z) = z_{w_s}. \tag{2.3.1}$$

In fuzzy classification approaches the label prediction is frequently calculated from the labels of the prototypes according to the fuzzy assignments

$$\Psi^{\mathcal{F}}(v_k) = \big(\psi_{w_1}(v_k), \ldots, \psi_{w_N}(v_k)\big)$$

yielding a class label assignment vector

$$\vec{\zeta}_{v_k}(Z) = \big(\zeta_{v_k}(1), \ldots, \zeta_{v_k}(C)\big) \tag{2.3.2}$$

with

$$\zeta_{v_k}(c) = \sum_{n=1}^{N} \big\{\psi_{w_n}(v_k) \,\big|\, z_{w_n} = z_c\big\} \qquad \text{for prototypes with crisp labels} \tag{2.3.3}$$

$$\zeta_{v_k}(c) = \sum_{n=1}^{N} \psi_{w_n}(v_k) \cdot z_{w_n}(c) \qquad \text{for prototypes with fuzzy labels} \tag{2.3.4}$$

Also other configurations, e.g. using crisp assignments with fuzzy prototype labels for fuzzy classification are possible and can be handled accordingly.

2.3.2 Basic definition for vector quantization and clustering

In vectorial clustering the aim is to discover the inherent structure of the input vectors $v_k \in V$ and group similar samples together. In the strict definition clustering also determines the number of clusters itself. Frequently the term clustering is also used for approaches that try to find a corresponding structure in the data for a given number of clusters. This realizes an encoding or compression of the input vectors. In clustering approaches most of the models are *cluster typical models*, explicitly modeling the data structure with cluster typical property descriptions. Examples for class typical models in the context of vectorial clustering are k-Means, Affinity Propagation and Growing Neural Gas.

Clustering approaches can be interpreted as processing only the inference stage of classification. In the case where an assignment function is learning, a newly incoming input vector v_k can be mapped – according to the assignment functions in equation (2.2.1) or (2.2.2) respectively – to the cluster index as a functional equivalent of a label:

$$\zeta_{v_k}(Z) = \Psi(v_k) \qquad \text{for crisp assignments} \qquad (2.3.5)$$

$$\vec{\zeta}_{v_k}(Z) = \big(\zeta_{v_k}(1), \ldots, \zeta_{v_k}(N)\big) = \Psi^{\mathcal{F}}(v_k) \qquad \text{for fuzzy assignments} \qquad (2.3.6)$$

2.4 Static and adaptive models

For building models of the data in clustering or classification there are two main approaches:

- *Static models* are calculated in advance from a fixed set of data and remain static afterwards. Newly incoming data can not be integrated into the model without completely recalculating it.
- *Adaptive models* remain adaptive for newly arriving data within a given structural frame.

Examples for static models are the Linear Discriminant Analysis (LDA) or hierarchical clustering approaches. Often the calculation of static models is efficient. Adaptive models, e.g. the Self Organizing Map (SOM), are robust and fault-tolerant and can change with newly incoming data. Because of their cognitive motivation adaptive models are also called learning approaches.

2.5 Unsupervised and supervised learning

If the focus is put on learning approaches there is another important concept to consider: the question whether to use supervised or unsupervised learning in the given application. This question does not exactly coincide with the question of classification or clustering, as we will point out in the following. In general: "In supervised learning, the goal is to predict the value of an outcome measure based on a number of input measures; in unsupervised learning, there is no outcome measure, and the goal is to describe the associations and patterns among a set of input measures."(Hastie, T. et al. 2003, p. xi)

There is also much ongoing work on combinations of supervised and unsupervised learning used in information processing especially in biomedical applications where information about the inputs is rare or expensive. (Corsini et al. 2006) introduce an example where the pairwise dissimilarities were known only for some of the input vectors. To effectively use this rare information they first used supervised learning to generate a general dissimilarity measure for the given input vectors. Using this dissimilarity measure they then could conduct the clustering.

Also in cases where the data are known to be multimodal, but the number and kind of modes are unknown, unsupervised learning can be used to find data inherent clusters. This provides a way of splitting the data into different subclasses, that can increase classification power. This concept was used e.g. by (Yang and Qu Yang 2006) who had the aim to create a tree-based classifier for functional protein classes. In every step of splitting data for the decision tree they used unsupervised methods to find the best split concerning the considered variable.

2.5.1 Issues of labeling data in biomedical domains

In biomedical classification problems the class labels can either be biomedical facts (e.g. whether the patient is alive or dead of disease) or annotations the experts gave to an encoded object. The later annotations are needed e.g. when different tissue types are analyzed or different kinds of cells have to be recognized. The experts have to annotate a set of inputs as training data for a classification approach. This process is also called manual labeling.

Especially in biomedical domains manual labeling of the training data for supervised learning can pose several severe problems. In some cases data sets are so huge that a human scanning of all data points is unacceptable. One possibility in this case is to use a small training set to train a preliminary classifier π in a first iteration. Ideally this classifier provides a reliability measure for every single data point's classification. This classifier can then be used to label another small set of test data points. If the experts are content with the performance of the classifier and the reliability measure is valid all other data points can be labeled using this classifier. Only the data labeled with low reliability is presented again to the experts to validate and eventually correct the label. If the classifier π does not perform good enough yet, it is retrained with the training and test set as new training data set. This process can be iterated until good classifier performance and thus probably good labeling quality is achieved.

Another approach to cope with very large data sets is to use so called semi-supervised learning. We refer to (Chapelle et al. 2006) for an overview of the state of the art. In these methods typically small amounts of labeled data are used together with large amounts of unlabeled data to train a classifier. Inherent structural information about the data is inferred from the unlabeled data and combined with the external structure information given by the labeled data.

Another issue is that the human experts' knowledge is complex and often implicit. If confronted with a question broken down to a machine learning task, the experts often have difficulties to constrain on a simplified or abstracted task. In our application example – the breast cancer project Exprimage – a tissue type characterization based on pathological tissue images had to be conducted. We asked the experts to annotate images of tissue samples (see section 6.5.1 on page 128 for details), a task that is not part of their normal diagnostic work. The experts marked very small areas and annotated a large number of tissue types. They did not want to loose any possibly relevant medical detail for the model that had to be built up from the tissue type representations.

As manual labeling in all biomedical domains is heavily biased by experience. It is likely that different experts highlight different tissue details as relevant. This causes the problem of bad intra- and inter-coder reliability. Intra-coder reliability refers to the observation that one expert may code the same fact differently at different points in time. Especially in pathology this is a very common phenomenon as the decisions are biased towards recent cases. In addition inter-coder reliability expresses the amount of agreement between different experts' codings. From the technical point of view the detailed manual coding can raise the problem of generalization as a sufficient number of training examples for all classes are needed that may not be available in any case.

It is possible to tackle this problem of manual labeling by computational approaches using a combination of unsupervised and supervised methods. In our example application of tissue type characterization we first applied unsupervised learning using a subset of the input vectors – as representations of pixels. The pixels where colored according to the corresponding cluster index of their representations. The resulting false color image was overlaid to the contributing pathological images with adjustable transparency. This way the experts evaluated the results and marked wrongly colored pixels additionally giving the right coloring. As the experts discussed their evaluation of the computational labeling, a consensus process was achieved. They agreed on cluster labels that are more likely to represent the examined biomedical reality (see also section "Post-labeling" below).

For computationally labeling the remaining data there are two possibilities:

- The trained unsupervised method together with the cluster labels is used to label also the remaining data points.
- The labeled training data from the first step is used to train a classifier that is evaluated again and used to label the remaining data accordingly.

The first approach is preferable if the clustering and its labels are totally accepted by the experts. In contrast to this the supervised retraining approach is able to meet requirements of editing the cluster membership of single input vectors.

Current developments in learning approaches also show possibilities to model the uncertainty in expert labeling directly, e.g. (Bouveyron and C. 2011) who introduced a probabilistic version of the Fisher Discriminant Analysis and (Bootkrajang and Kabán 2011) extending multi-class quadratic normal discriminant analysis with a model of the mislabeling process.

2.6 Post-labeling

The possibility of post-labeling is tightly connected to the question of unsupervised and supervised learning. It is well suited for prototype based, class typical learning. First an unsupervised learning is executed on the whole input vector set. If the input vectors already carry class labels the labeling of the prototypes can be conducted as follows. If the prototypes ought to have crisp labels they can be chosen by the majority vote from the class labels of the input vectors in the respective receptive field of the prototype. For fuzzy prototype class labels the relative amount of different class labels can be used as fuzzy labels.

If the given data points are unlabeled, there is also the possibility to generate labels after unsupervised learning by using the domain experts' judgment of the unsupervised learning results. In a suitable visual evaluation process (a choice of these processes is introduced in the sections 4.6 and 6.5.1) the domain experts are asked to assign class labels to the identified clusters. This can be done by labeling the cluster defining prototypes by hand and automatically classifying all input vectors in the receptive field. Then this classification also has to be evaluated. Another variant is to label every input vector by hand. This may be very time consuming and impracticable.

In the case of unbalanced data sets using crisp post-labeling with majority vote can lead to unrepresented classes. This occurs for data clouds where the small quantity of data points for a small class is outvoted by the large quantity of data points of a larger class. If that occurs for all data clouds in which the small class is present, there is no prototype representing the small class with its label.

2.7 Online and batch learning

Adaptation in the learning approaches can be done in different schemes. One difference is their adaptation interval, i.e. how many data points are used for one adaptation step. The adaptation schemes that we will focus on are:

- *Online learning*: data can come in over time and the adaptation takes place after every single input vector. This approach is often used for adaptive models.

- *Batch learning*: data is assumed to be independent and identically distributed over time and adaptation takes place using the complete set of input vectors. This approach is preferably used for static models and the adaptation is often done using an Expectation Maximization principle (see e.g. (Hastie, T. et al. 2003)). This does not necessarily require differentiable similarities in the determination of the update rules for the prototypes.

Online learning often converges slower than batch approaches but is not that prone to instability with respect to different initializations as compared to batch-learning, see e.g. (Cottrell et al. 2006). If it is known that the underlying concepts of the data drift over time, online learning is chosen and it is useful to integrate some kind of memory that is able to forget older samples (see e.g. (Biehl and Schwarze 1993), (Widmer and Kubat 1996) or (Vovk 2005)).

2.8 Learning for different dissimilarity types

More or less parallel to the question of online or batch learning, it is possible to distinguish different forms of learning in class typical model approaches according to the dissimilarities that are appropriate for comparing the chosen object encodings:

- *Learning using dissimilarity functions*: In these approaches the dissimilarity between object encodings can be determined according to a given function or procedure d. So for every possible point in the input space we can calculate the dissimilarity to any other possible point in the input space. Thus, it is possible for the prototypes to be placed on arbitrary points in the input space. If the dissimilarity measure is differentiable gradient based methods can be applied.

- *Relational learning*: In some applications only dissimilarity values between pairs of object encodings are given. A isometric projection of the object encodings into another, potentially high dimensional, space is assumed. That means that the given dissimilarities are Euclidean distances of these projected encodings. The embedding is called (pseudo) Euclidean embedding, see section 3.4. It is possible to describe the prototypes in the projection space.

- *Median learning*: If only pairwise data dissimilarities are available that allow no (pseudo) Euclidean embedding, no other dissimilarities can be approximated reliably. The prototypes can only be selected among the training object encodings themselves. Such prototypes are called exemplars.

A special case of learning is *learning with adaptive dissimilarities*. It is also called *relevance learning* as introduced by (Bojer et al. 2001) for Learning Vector Quantization.

In these approaches a dissimilarity function is specified only by its structure and is adaptable according to special parameters (see chapter 3 for examples). The first relevance learning approaches used metrics as basic structures and thus were also called *metric adaptation*. Popular relevance learning approaches use dissimilarity functions for learning. We will also introduce variants incorporating relational data in chapter 5.

In the case where the input is vectorial data the dissimilarity measure structure can for example be a weighted Euclidean distance. In this dissimilarity measure the difference in every vectorial dimension m is additionally weighted by parameter α_m. In metric adaptation approaches, the dissimilarity parameters are inferred from the training data during learning. For the parameter adaptation different strategies can be applied:

- *Global dissimilarity adaptation*: In this case only one set of parameters (in our example α) is adapted, which defines a global dissimilarity measure for all inputs (in the example $d = d_\alpha$).

- *Local dissimilarity adaptation*: In local adaptation several dissimilarity measures are optimized in terms of their parameters. In prototype based approaches this is done e.g. by learning the parameters $\vec{\alpha}_n$ for one dissimilarity measure for each prototype w_n.

Under specific restrictions the parameters learned for the dissimilarity can be interpreted as relevance factors for features or combinations of features. The restrictions are detailed in chapter 4 for example approaches.

As data are often encoded as real-valued vectors, traditionally the Euclidean metric is used as dissimilarity function in learning. Metric adaptation is commonly applied for further insights in the relevance of single feature dimensions. With more complex data and dissimilarity representations the need for approaches using pairwise dissimilarities arose. In section 3.4 we show the approach to ascertain whether a pseudo Euclidean embedding is available for the given dissimilarities. In the use of median learning a sufficiently large number of training data points is required to receive stable and suitable results. This precondition was e.g. not met in our application example Exprimage.

2.9 Integration of mixed data

Biomedical objects are often encoded by an ensemble of feature groups from different data modalities. This goes beyond the encoding of objects as input vectors in a vectorial feature input space. The question for classification or clustering is how the different feature groups for one object could be integrated to gain a description of the object that is more appropriate as it contains more information. As mentioned before, the individual feature groups are often of different types, e.g. numerical or categorical features. There are several specialized approaches to cope with mixed data[1], especially in classification. One popular approach is the integration of mixed data into the class discriminating model of a Support Vector Machine via mixed kernels. The integration of simple mixed kernels was originally developed for multivariate data analysis. The algorithmic foundation was laid by (Aitchison and Aitken 1976), who parameterized this integration. The parameters used in the integration can either be set to a fixed value or can, just as in metric adaptation, be adapted during the learning or inference stage.

There are also methods for incorporating mixed data into basic statistic or clustering approaches, e.g. in General Linear Statistics (Levasseur et al. 2009), Similarity-Based Agglomerative Clustering (Li and Biswas 2002) or Evidence-Based Spectral Clustering (Luo et al. 2006). Furthermore simple prototype based, class typical models were equipped with the possibility to handle mixed data, e.g. the Nearest Neighbor Approach (García-Borroto and Ruiz-Shulcloper 2005) or the K-Means approach (Huang 1998). Most of these methods are based on Euclidean dissimilarities. Extensions for more complex prototype based methods exist but are restricted to integral dissimilarity measures for all numerical and all categorical features. They cannot cope with arbitrary dissimilarities or more groups of features in the data (see e.g. for Learning Vector Quantization (Chen and Marques 2010)).

2.10 Evaluation of learning results

For the evaluation of learning results it is relevant to distinguish between two different views:

The algorithmic view is concerned with the general evaluation of a method. Here the computer science experts evaluate an algorithm with focus on aspects like sensitivity against initialization, suitable choice of parameters, convergence criteria, or algorithm complexity.

The application view is focused on the evaluation of a method for a concrete application. Here the computer science experts as well as the domain experts

[1] Processing of mixed data types has to be distinguished from the processing of heterogeneous data. In heterogeneous data each object can be of a different type (see e.g. (Globerson et al. 2005)), whereas in mixed or structured data all objects have the same structure that can contain different types of feature groups.

evaluate the accuracy of the results and the consistency with earlier findings, expert knowledge and with results obtained by other methods.

In both views there are two main evaluation approaches:

Numerical evaluation This evaluation uses measures like the recognition rate or Cohen's kappa values. It is often understood as a performance evaluation with respect to some *ground truth*, e.g. labels or the assumed presence of compact clusters.

Visual inspection In one perspective visual inspection allows the judgment of the learning quality by the back projection of the learning results to the specific learning task, e.g. in the case of image analysis. Furthermore cluster qualities like compactness, separation or density are evaluated using visualizations of the data space.

The methods for visual inspection depend more on the learning task, on the data and the application than the numerical evaluation methods. We introduce the evaluation methods in relation to the basic learning methods (see chapter 4).

When evaluating with respect to ground truth, labels given for the data are interpreted as the true class membership. This assumption has to be checked carefully in the biomedical domains. We refer to section 2.5.1 for general considerations concerning labeling and section 6.5.1 for the concrete experiences in the application example. If there is uncertainty in the class labeling, ways to reduce this uncertainty are the application of unsupervised learning and post-labeling or semi-supervised methods.

Visual inspection relies on an appropriate visualization quality. Faithful visualization can be used for data exploration. The presentation of the results as well as the conditions and data that lead to these results have to be accessible for the domain experts in a plausible way. As far as possible, the presentation should be oriented towards the experts' cognitive abilities and common working environment. This is related to the research field of suitable knowledge representation and cognitive ergonomics.

The evaluation of learning results is necessary in the context of selecting an appropriate training algorithm. Some algorithms are based on assumptions, e.g. about the convexity of a problem, that cannot be proven easily by only looking at the application or the data. In this case a pertinent approach is to apply several methods with different underlying assumptions to the given problem. The comparison and plausibility discussion of the results is a suitable way to evaluate the validity of the different algorithms' assumptions in the given application.

Many evaluation approaches especially in numerical evaluation yield more representative results the more data points are available in the data set, that is used for evaluation. Especially in the biomedical domain the number of data points is often limited. For most evaluation approaches workarounds or specialized evaluation strategies exist that are able to cope with small data sets. We will detail them in the context of the corresponding evaluation methods.

2.11 Discussion of suitable algorithm types

The choice of an algorithm is often affected by personal preferences that were formed in previous work with the considered algorithms. This is not necessarily bad practice as the personal experience and intuition in working with the preferred algorithms can be beneficial for the parameter adaption of the algorithm to the current task.

There are fundamental advantages and disadvantages of different algorithm types in cognitive support systems for biomedical research. In biomedical applications, enabling systems are often used to identify interesting groups of data or patterns defining such groups. From the definition of the different basic approaches in sections 2.3.1 and 2.3.2 it follows that class discriminating models describe the borders between the groups rather than the groups themselves. Thus it appears advantageous to use class or cluster typical models for learning more about the groups in the data.

The class or cluster typical models also show advantages in evaluating the results. In class or cluster typical models it is easy to explore representatives of groups and patterns e.g. by displaying the representatives directly or choosing for every representative the closest training data point. The data needed for pertinent evaluation visualization are inherent results of the algorithms.

In enabling systems unusual events are of interest and a system for the detection of outliers or novelty is indispensable. Class typical models allow an intuitive introduction of such systems, see e.g. (Vovk 2005). If class conditional properties are modeled, the reliability with which each data point is classified in this model can be calculated. The classification of data points that yields a reliability under a certain threshold is rejected. The data point is assumed to be too different from the model. Often biomedical experts favor this option over a system trying to classify all data. A display of rejected objects can show the necessity of more representative training data, e.g. if the class of the object is not involved yet or this subtype of the class is not adequately represented. It can also show the occurrence of a concept drift in the data or a problem in biomedical probe preparation.

The choice of suitable algorithms is further influenced by known data properties. Biomedical data are often subject to noise or errors. In this case the application of adaptive models or learning approaches is recommended. Most of the adaptation algorithms that are commonly used are based on dissimilarities. We will detail this concept in chapter 3. Many of these algorithms use the Euclidean metric to measure how similar two object encodings are. Metric adaptation is frequently used as it helps to identify the relevance of single features or combinations of them for the classification or clustering at hand. It is especially suited to yield insight into the data set and problem.

There are data for which the Euclidean metric or adaptive versions of it seem not appropriate, e.g. in the case of categorical data[2]. For these data some kind of relation between the objects should be used for defining their pairwise dissimilarities. As detailed in section 2.8 there are dedicated learning approaches available for the different kinds of dissimilarities. As the choice of the dissimilarity measure is highly domain specific, it is not possible to give a general advise for choosing between learning using dissimilarity functions, relational and median learning or metric adaptation. The same holds for the question whether online or batch learning should be used. The specification of the preconditions and the suitable algorithms in our application example Exprimage is given in section 7.1.1.

Biomedical data are often represented by mixed data. That means they comprise feature groups of different data types. It is possible that these different data types adequately would be handled using corresponding different dissimilarities. As the single feature groups are conceptually and semantically different, it may be inadequate to just concatenate them and use some Euclidean overall measure of dissimilarity. Neglecting single feature groups is just as inappropriate as the features might express their relevance only in the combination and give a context for each other. The relevance of the single groups should be reflected in the construction of the overall dissimilarity measure used for comparing two encoded objects. The context and relevance of the feature groups can be different for single prototypes of classes or clusters. A local metric with different relevance parameters for the dissimilarities in the single feature groups is required in this scenario. As setting these relevance parameters by hand would be complex and prone to errors, we suggest to adapt them during training in a dissimilarity adaption approach.

Another question about known data properties is whether classes in the data are likely to contain multiple separated modes or data clouds. If this is the case, an algorithm is needed that is able to either

- find the centers of the different data clouds and assign all of them to a common class or

[2] Examples for categorical data are the gender or the blood group.

- project the data into some space where the different data clouds for one class are not distinct any more.

An approach like Linear Discriminant Analysis is not able to cope with such multimodal data as it projects only one mean per class. In approaches based on representatives or prototypes that problem does not arise if more than one representative per class is used.

Furthermore, we have to decide on the mapping that is used in the clustering or classification approach. If the focus is on gaining information about the data and the data space, fuzzy mappings are preferred as they do not draw artificial borders in the case of overlapping classes or clusters. These borders are needed if the clustering or classification is embedded into some work flow or procedure that needs a crisp decision to proceed.

2.12 Conclusion

Summarizing we advise to use prototype based methods that build class or cluster typical models. These models should be built in an adaptive manner, where the adaption scheme is chosen according to the time constraints and the data properties as well as the resulting dissimilarity properties. It is preferable to use approaches that are able to cope with multimodal data. In this thesis we chose Vector Quantization as a base for all further developments as it fulfills these properties. In the following chapter we discuss a choice of algorithms based on Vector Quantization (VQ) for different time constraints, data and dissimilarity properties. We show extensions of the crisp median algorithms to yield fuzzy mappings instead. In chapter 3 we introduce different kinds of dissimilarities used for comparing object encoding feature groups.

We extend the GLVQ approach as well as the BNG approach in this thesis to object encodings that are composed of feature groups that need different dissimilarity measures. Inspired by the metric adaptation methods introduced in the Vector Quantization chapter (see chapter 4), we show two basically different ways to integrate adaptive dissimilarities into learning:

1. Calculate the overall dissimilarity as weighted sum of dissimilarities in the single feature groups. This idea is based on Generalized Relevance Learning Vector Quantization (Hammer and Villmann 2002).

2. Calculate the overall dissimilarity as weighted sum additionally integrating combinations of feature groups in a first order, an idea based on Generalized Matrix Learning Vector Quantization (Schneider et al. 2009).

We detail and formalize these conceptual ideas in chapter 5. Furthermore, we extend the integrated dissimilarity adaptation to data comprising relational feature groups. We did not extend median approaches as in our application example there are not enough data samples available to expect reliable and stable results in median learning.

2.13 Related publications and authors' contributions

The work presented in this thesis was partly already published. The author collaborated in the fuzzy extensions of Median C-Means (Geweniger et al. 2010) and Affinity Propagation (Geweniger et al. 2009) discussed in sections 4.3.2 and 4.3.2 respectively. The author was the main contributor in the extension of the evaluation measure Fleiss' kappa to fuzzy mappings (Zühlke et al. 2009) (see section 4.5.4). In (Zühlke et al. 2010) we describe the extension of the traditional prototype based approach GLVQ to handle mixed data that was developed mainly by the author. The other algorithms introduced in chapter 5 as framework for the handling of mixed data in prototype based methods were derived by the author and not published yet. Concerning the application example discussed in chapter 6 the author initiated and supervised the Master thesis of Khabirova (Khabirova 2011) and the Diplomarbeit of Bornemeier (Bornemeier 2011).

Chapter 3

Dissimilarities

In this chapter we want to detail the concept of *dissimilarity*. This concept is associated with a variety of terms, including distance, similarity, measures, metrics. For sake of clarity we will first give the mathematical formalization starting with the definition of a metric space. The definitions are in accordance with those given in (Pękalska and Duin 2006).

Assume a set $\mathbb{V}$[1], then a mapping $d(x, y) : \mathbb{V} \times \mathbb{V} \to \mathbb{R}^+$ is called a *metric* if it fulfills the following axioms:

1. $d(x, x) = 0$ (*reflexivity*),
2. $d(x, y) = d(y, x)$ (*symmetry*),
3. $d(x, y) = 0 \Longrightarrow x = y$ (*definiteness*) and
4. $d(x, z) \leq d(x, y) + d(y, z)$ (*triangle inequality*)

Axiom (1) together with axiom (3) form the axiom of the so called *identity of indiscernibles*.

It may be conceptually more adequate in a considered application to drop one or more of the axioms. Then we get the following categorization for mappings $d(x, y) : \mathbb{V} \times \mathbb{V} \to \mathbb{R}^+$

1. d is called a *(dis-)similarity measure* if it is reflexive.
2. d is called a *definite (dis-)similarity measure* if it is reflexive and definite.
3. d is called a *pre-(dis-)similarity measure* if it is reflexive and symmetric and $\mathbb{V}$ is then *premetric*.
4. d is called a *quasimetric* if it is premetric and satisfies the definiteness constraint and $\mathbb{V}$ is then *quasimetric*.
5. d is called a *semimetric* if it is premetric and satisfies the triangle inequality and $\mathbb{V}$ is then *semimetric*.

[1] Usually in our applications this set $\mathbb{V}$ is the input vector space and a subset of or equal to $\mathbb{R}^M$.

Often the input data are encoded as real-valued vectors or – to put it into other words – points in an input vector space. For such data the interpretation of similarity as distance in the input vector space is intuitive. Thus the Euclidean distance is an appropriate measure of dissimilarity. It fulfills all metric axioms. Many adaptation schemes in learning algorithms are based on the Euclidean metric. Some of them can also be used with other metrics – relying on the metric axioms.

With growing complexity of the applications the idea of input data being simple vectors becomes more and more inappropriate. To model biomedical objects a variety of other representations become interesting, e.g. spectra (being actually some functions of wavelength or mass) or clinical data (often expressed in terms of membership to some category). There are at least two possibilities to cope with these object representations. Either these representations are recoded into a vectorial representation or other dissimilarity measures are needed. As recoding often cannot be done without loosing information, it is more suitable to use an appropriate dissimilarity measure. This may also mean that a suitable learning approach for this measure has to be found or that the learning approach has to be adapted to the chosen dissimilarity measure.

The choice of the dissimilarity measure is based on the object encodings – expressed in terms of *features*, *attributes* or groups thereof – and their mathematical properties as well as their conceptual meaning. Hastie et al. state: "Although simple generic prescriptions for choosing the individual attribute dissimilarities [...] can be comforting, there is no substitute for careful thought in the context of each individual problem. Specifying an appropriate dissimilarity measure is far more important in obtaining success with clustering than choice of clustering algorithm. This aspect of the problem is emphasized less in the clustering literature than the algorithms themselves, since it depends on domain knowledge specifics and is less amenable to general research." (Hastie, T. et al. 2003, p. 506)

Furthermore, in the context of the choice of an adequate dissimilarity measure the question of an adequate scaling of the features arises naturally. In the Euclidean distance all feature dimensions considered are of equal weight and their order has no influence on the distance calculation. In this case a scaling that normalizes every feature dimension independently from the others is often adequate and commonly used. For example for features representing functions this is not the case. Strickert et al. introduced a general method for the assessment of data attribute variability that "allows a mathematically rigorous characterization of attribute sensitivity given not only Euclidean distance between data points but very general similarity measures" (Strickert et al. 2011, p.105). An approach towards scaling the data attributes accordingly is an issue of current research.

In the following sections we focus on the dissimilarity measures that are needed later in the specific application (see section 7.1.4): metrics (especially the Euclidean metric), dissimilarity measures (especially divergences) and relational dissimilarities (in terms of pairwise dissimilarities). For these dissimilarity measures we will also discuss adequate scaling approaches. A selection of other dissimilarity measures will be mentioned and sketched roughly.

3.1 Metrics

As formalized before, metrics are a strictly defined category of dissimilarity measures. An important special case of a metric is the Euclidean metric as it is the most intuitive measure in terms of the psychology of perception and cognition. Humans in every day life think space in terms of the Euclidean distance on an orthogonal coordinate system. For example the triangle inequality satisfies the intuition that the distance cannot be shorter when going from point a to point b via point c instead of going directly from a to b. In this section we will also introduce other metrics with non-Euclidean behavior.

3.1.1 Euclidean metric and relatives thereof

The *Euclidean metric* (also called Euclidean distance) is a special case of a metric induced by the *L^p-norm*. The L^p-norm is given by:

$$|v|_p = \left(\left|[v]_1\right|^p + \left|[v]_2\right|^p + \cdots + \left|[v]_M\right|^p\right)^{\frac{1}{p}}$$

where $[v]_m$ is the m^{th} dimension of the vector $v \in \mathbb{R}^M$.

With $p \geq 1$ a general metric – called *Minkowski metric* – is induced by this norm[2]. It is defined as:

$$d_{p \text{ or Minkowski}}\left(v, w\right) = \left(\sum_{m=1}^{M} \left|[v]_m - [w]_m\right|^p\right)^{\frac{1}{p}}$$

As the sum in the definition of the norm and the metric is commutative, the order of the coordinates i is meaningless in the calculation. These metrics are therefore suitable for vector representations of objects where the order of the dimensions is not informative.

Important special cases of this L^p-norm-induced metric are:

[2] For $0 < p < 1$ the norm induces a quasimetric that violates the triangle inequality.

Manhattan metric with $p = 1$:

$$d_{\text{Manhattan}}\left(v, w\right) = \sum_{m=1}^{M} \left|[v]_m - [w]_m\right|$$

It sums up the absolute differences in the single coordinates. This metric is also known as *taxicab metric* as it follows axis-aligned directions.

Euclidean metric with $p = 2$:

$$\begin{aligned} d_{\text{Euclidean}}\left(v, w\right) &= \sqrt{\sum_{m=1}^{M} \left([v]_m - [w]_m\right)^2} \\ &= \sqrt{\left(v - w\right)^{\top} \left(v - w\right)} \end{aligned} \tag{3.1.1}$$

The Euclidean metric is the metric most widely used. In some applications the squared Euclidean metric

$$d_{\text{Euclidean}}\left(v, w\right) = \left(v - w\right)^{\top} \left(v - w\right) \tag{3.1.2}$$

is used instead.

Chebyshev metric with $p = \infty$:

$$d_{\text{Chebyshev}}\left(v, w\right) = \max_m \left(\left|[v]_m - [w]_m\right|\right)$$

It assumes only this dimension as relevant that has the largest absolute difference between the single coordinates. This metric is associated to chess descriptions, e.g. (van der Heijden et al. 2004), and sometimes used in warehouse logistics, e.g. (Langevin and Riopel 2005).

These L^p-norm-induced metrics are commonly used for vectorial data. As the dimensions are weighted equally in these metrics and there is no information in the order of the dimensions, the features in the single dimensions are often independently normalized. A common strategy is to test the values within one feature dimension in the data set whether they are distributed according to the normal distribution by applying the Jarque-Bera-test (Jarque and Bera 1980). If the feature dimension values are distributed accordingly they are normalized to have zero mean and a standard deviation of one – called zero-mean-normalization. If that is not the case often a linear scaling to a range of -1 to 1 is applied.

There are metrics extending the abilities of the Euclidean metric to account for different variances in the specific feature dimensions. The *Mahalanobis distance* is defined as a metric between two vectors v and w of the same distribution with the covariance matrix M_{cov}:

$$d_{\text{Mahalanobis}}\left(v,w\right)=\sqrt{\left(v-w\right)^{\top}M_{cov}^{-1}\left(v-w\right)} \tag{3.1.3}$$

If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance.

Metrics cannot only be induced by a norm. A positive definite, symmetric bilinear form on a real vector space $\mathbb{V}\subseteq\mathbb{R}^{M}$ induces a metric (see e.g. (Pękalska and Duin 2006) for definitions). If $\mathbb{V}$ is finite dimensional the symmetric bilinear form $\mathfrak{b}$ with respect to a special vector space base $\mathfrak{B}=\{\mathfrak{v}_i\}_{i=1}^{M}$ can be defined completely by the *defining matrix* Λ of $\mathfrak{b}$ with

$$\Lambda\left(\mathfrak{b}\right)_{i,l}=\left(\mathfrak{b}\left(\mathfrak{v}_i,\mathfrak{v}_l\right)\right) \tag{3.1.4}$$

for all $i,l=1,\ldots,M$. The bilinear form $\mathfrak{b}$ is symmetric if and only if Λ is symmetric. Assume $\mathfrak{b}$ to be a real, symmetric and non-degenerate bilinear form. The latter means that the defining matrix Λ of $\mathfrak{b}$ has full rank of M. If we further assume, that Λ is positive definite then $\mathfrak{b}$ defines a traditional inner product in $\mathbb{V}$. It is denoted as $\langle\mathfrak{x},\mathfrak{y}\rangle_{\Lambda}=\mathfrak{x}^{\top}\Lambda\mathfrak{y}$ where $\mathfrak{x}$ and $\mathfrak{y}$ are expressed with respect to the basis $\mathfrak{B}$ (Pękalska and Duin 2006).

An important special case of a dissimilarity measure induced by such a bilinear form is the quadratic form

$$d_{\Lambda_{\mathfrak{b}}}\left(v,w\right)=\left(v-w\right)^{\top}\Lambda\left(v-w\right). \tag{3.1.5}$$

where Λ is the defining matrix of a symmetric bilinear form $\mathfrak{b}$ and v and w are expressed in the same basis as Λ. For $\Lambda=\mathbb{1}$ this yields the squared Euclidean distance.

To ensure the metric properties for the dissimilarity measure given in (3.1.5), the matrix Λ must be symmetric and positive definite. It holds that for every matrix Ω, the matrix Λ given by

$$\Lambda=\Omega^{\top}\Omega \tag{3.1.6}$$

is positive semi-definite and symmetric. The other way round, each positive definite, symmetric matrix Λ is decomposable according to equation (3.1.6), but in this case Ω is not unique. Using equation (3.1.6), equation (3.1.5) and the definition $u=v-w$ we get $d_{\Lambda}=u^{\top}\Lambda u=u^{\top}\Omega^{\top}\Omega u=\left(\Omega u\right)^{2}\geq 0$ for all u and thus guaranties positive semi-definiteness. To guarantee strong positive definiteness additionally $\det\Lambda\neq 0$ is required. If $\Omega=(j\times i)$ it can be seen as a map $\Omega:\mathbb{R}^{i}\rightarrow\mathbb{R}^{j}$. Then $\left(\Omega u\right)^{2}$ is the quadratic Euclidean distance in $\mathbb{R}^{j}$. This mapping property of the matrix Ω can be used to improve the conceptual validity of the comparison of different objects and thus to enhance the quality of the learned model. In this sense, the matrix Λ or Ω respectively can be used to integrate prior knowledge about the relevances of single vector dimensions or combinations thereof.

The restriction to diagonal matrices yields:

$$d_\alpha\left(v, w\right) = \sum_{m=1}^{M} \alpha_m \big(\left[v\right]_m - \left[w\right]_m \big)^2. \tag{3.1.7}$$

with $\vec{\alpha} = (\alpha_1, \ldots, \alpha_M)$, $\alpha_m \geq 0$ and $\sum_{m=1}^{M} \alpha_m = 1$. This function in general is no metric. Here again prior knowledge about the vector dimensions can be introduced to enhance the conceptual validity of object comparisons.

As we will see in chapter 4, the matrix Λ and the vector $\vec{\alpha}$ can be adapted during the learning to create a discrimination optimal dissimilarity. To interpret these parameters as a relevance weighting, it is necessary to normalize the single feature dimension beforehand.

The Mahalanobis distance and its generalizations are used for vectorial data where the order of the dimensions does not provide any information. These dissimilarities yield a kind of intrinsic normalization and thus are suitable if the single dimensions have variable ranges that should be equalized in distance calculation.

3.1.2 Other metrics

(Lee and Verleysen 2005) introduced a generalization of the L^p-norm for time series, inducing the *functional metric* that takes the order of the vector dimensions into account. (Villmann and Hammer 2009) used the *Sobolev metric*, that is induced by the Sobolev norm, to represent the dissimilarity of vectors that have a function like character. Examples of biomedical applications where such functional vector representations are used are: specified parts of an EKG curve or of an EEG curve, absorption or reflection spectra or action potentials as well as fluorescence distributions around cell membranes.

Based on information theory the *Kolmogorov metric* is also suited for (normalized) spectral data, see (Paclík and Duin 2003) for a detailed description. There are also metrics that are neither induced by a norm nor by a real bilinear form, see (Pękalska and Duin 2006).

3.2 Dissimilarity measure functions

As formalized at the beginning of this chapter every reflexive mapping $d(x, y)$: $\mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R}^+$ is a dissimilarity measures. This is a rather loose restriction. In this section we will focus on divergences and shortly introduce a choice of other dissimilarity functions afterwards. Measures like the quasimetric induced by the L_p-norm for $0 < p < 1$ introduced before, are also examples for dissimilarity measure functions.

3.2.1 Divergences

Divergences are a group of dissimilarity measures originally proposed for the comparison between density functions or positive measures (Villmann and Haase 2011). Often used in physics divergences were introduced into learning approaches some years ago. They are said to have a great potential in the representation of dissimilarities in many real world applications. The following paragraph and definitions were adapted from (Villmann and Haase 2011) who give an overview and the mathematical framework for using divergences in gradient based approaches.

We assume p and q to be positive measures in x which means that they are positive functions for the support $x \in \mathbb{X}$ with finite weight, i.e. $\int_{\mathbb{X}} p(x)\,\mathrm{d}x < \infty$. If further $\int_{\mathbb{X}} p(x)\,\mathrm{d}x = 1$ holds, $p(x)$ is called a density measure, or simply density. In the following we abbreviate $p(x)$ by p for readability.

(Cichocki et al. 2009) classified the large variety of divergences into three, partially overlapping, main classes:

- Bregman divergences
- Csiszár's f-divergences
- γ-divergences

A choice of important divergences for the application in prototype based learning are:

The standard Kullback-Leibler-divergence given by

$$d_{\mathrm{KL}} = \int p \log \frac{p}{q}\,\mathrm{d}x. \tag{3.2.1}$$

It can be attributed to all three divergence classes mentioned before.

The generalized Rényi-divergences that are defined as

$$d_{\alpha}^{\mathrm{GR}}(p,q) = \frac{1}{\alpha - 1} \log \left(\int \left[p^{\alpha} q^{1-\alpha} - \alpha \cdot p + (\alpha - 1)\, q + 1 \right] \mathrm{d}x \right). \tag{3.2.2}$$

The γ-divergences as a class of very outlier-robust divergences that are defined according to

$$d_{\gamma}\left(p,q\right) = \frac{1}{\gamma+1}\log\left[\left(\int p^{\gamma+1}\mathrm{d}x\right)^{\frac{1}{\gamma}} \cdot \left(\int q^{\gamma+1}\mathrm{d}x\right)\right] \tag{3.2.3}$$

$$- \log\left[\left(\int p \cdot q^{\gamma}\mathrm{d}x\right)^{\frac{1}{\gamma}}\right]. \tag{3.2.4}$$

In the limit $\gamma \to 0$ $d_{\gamma}\left(p,q\right)$ this becomes the standard Kullback-Leibler-divergence for normalized densities. For $\gamma = 1$ the **Cauchy-Schwarz-divergence**

$$d_{\mathrm{CS}}\left(p,q\right) = \frac{1}{2}\log\left(q^{2}\left(x\right)\mathrm{d}x \cdot \int p^{2}\left(x\right)\mathrm{d}x\right) - \log\left(\int p\left(x\right) \cdot q\left(x\right)\mathrm{d}x\right) \tag{3.2.5}$$

is obtained, which was suggested for information theoretic learning by (Principe et al. 2000) investigating the Cauchy-Schwarz-inequality for norms.

The question when to use which divergence for what data is an issue of ongoing research as the incorporation of divergences in information processing algorithms is a quite new idea. For the parameterized divergences it is also possible to adapt the parameter (e.g. γ or α) during learning to yield better and more stable results, see (Villmann and Haase 2011). In principle divergences are suitable for all kinds of probability densities and some are suited for unnormalized positive measures. As it is possible to interpret histograms or distributions of image features as probability densities, they can be compared using divergences. Other examples are spectra that can be interpreted as positive measures. Due to measuring inaccuracy it is sometimes necessary to renormalize the spectra so that they sum up to one.

(Mwebaze et al. 2011) give an example where classification rates after learning from histograms were better when using γ-divergence than for the Euclidean comparison of the histograms. The Rényi-divergence is in some cases more robust compared to the Kullback-Leibler-divergence due to the fact that in the Rényi-divergence, see equation (3.2.1), the integral is calculated first and then the logarithm is determined afterwards. In the Kullback-Leibler-divergence the logarithm is calculated directly over possibly very small values, which can cause numerical instabilities due to error propagation.

In image processing frequently properties of the image e.g. the intensity values are modeled by fitting a Gaussian distribution to them. The mean μ and the standard deviation σ of the fitted Gaussian are then used as image features. Divergences offer a good way to compare these Gaussian distributions g_1, g_2 with $g_i = g\left(\mu_i, \sigma_i\right)$ for $i = 1, 2$ and thus for comparing these image features. For example the Kullback-

Leibler-divergence as given in equation (3.2.1) can be calculated for two Gaussian functions in terms of their mean μ and standard deviation σ:

$$d\left(g_1, g_2\right) = \frac{1}{2}\left[\frac{\left(\mu_1 - \mu_2\right)^2}{\sigma_2^2} + \frac{\sigma_1^2}{\sigma_2^2} - \log\frac{\sigma_1^2}{\sigma_2^2} - 1\right]. \qquad (3.2.6)$$

3.2.2 Other dissimilarity measure functions

Another important dissimilarity measure function especially for functional data represented by vectors v_k and w_n is the *Pearson correlation coefficient*, given by:

$$d_r\left(v_k, w_n\right) = \frac{1}{M-1}\sum_{m=1}^{M}\frac{\left([v_k]_m - \mu_{v_k}\right)\left([w_n]_m - \mu_{w_n}\right)}{\sigma_{v_k}\sigma_{w_n}} \qquad (3.2.7)$$

where μ_z and σ_z are the mean and the standard deviation over the vector components $[z]_m$, with $z = w_n, v_k$ respectively, i.e.

$$\mu_z = \frac{1}{M}\sum_{m=1}^{M}[z]_m \text{ and } \sigma_z = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left([z]_m - \mu_z\right)^2}.$$

(Pękalska and Duin 2006) give an overview over the large variety of different dissimilarity measures and their properties. (Paclík and Duin 2003) review dissimilarity measures suitable for spectral data e.g. the *Spectral Angle Mapper*.

3.3 Pairwise dissimilarities

In some applications only pairwise dissimilarities between the input data points are given. This often occurs for categorical data. There are some suggestions for the determination of pairwise dissimilarities in data comprising categorical and numerical features. As we will introduce advanced integration approaches for mixed data later we focus here on the calculation of the dissimilarity for the categorical feature parts. We discuss two possibilities for pairwise dissimilarity calculation for categorical features from literature that we used in our application example.

(Li and Biswas 2002) define a similarity measure for nominal data. For this measure the similarity of the feature value matches between a pair of objects is weighted by the frequency of occurrence of the feature value in the data set. The idea is that two object representations are more similar to each other if they agree in a less common feature value than if they agree in a frequent one. This idea is based on the information theoretic assumption that importance of an information is anti-correlated to its frequency.

In the following a formalization of this concept is given. Without loss of generality we assume two pairs of objects (v_i, v_l) and (v_r, v_u) with $v_i, v_l, v_r, v_u \in V$. Furthermore we assume that for the m^{th} feature dimension the values in the feature vector are equal within each pair, i.e. $[v_i]_m = [v_l]_m$ and $[v_r]_m = [v_u]_m$ but $[v_i]_m \neq [v_r]_m$. We further assume that the value $[v_i]_m$ appears equally often or more frequent in the set of input vectors V than the value $[v_r]_m$. (Li and Biswas 2002) express that fact as

$$\big((p_i)_m = (p_l)_m \big) \geq \big((p_r)_m = (p_u)_m \big)$$

where $(p_i)_m, (p_l)_m, (p_r)_m, (p_u)_m$ define the probabilities (relative amounts) of occurrence of the respective feature values in V. Their idea is summarized in the following relation (Li and Biswas 2002):

$$\left.\begin{array}{r} \big([v_i]_m = [v_l]_m\big) \wedge \big([v_r]_m = [v_u]_m\big) \\ \big((p_i)_m = (p_l)_m \big) \geq \big((p_r)_m = (p_u)_m \big) \end{array}\right\} \quad \Rightarrow (S_{i,l})_m \leq (S_{r,u})_m \tag{3.3.1}$$

where $(S_{i,l})_m$ and $(S_{r,u})_m$ are the similarity values for the respective feature values and objects.

According to this relation Li and Biswas define for a pair of objects (v_i, v_l) with equal value in feature dimension m the *More Similar Feature Value Set*, abbreviated by MSFVS$\big([v_i]_m, [v_l]_m\big)$. "This is the set of all pairs of values for feature m that are equally or more similar to the pair $([v_i]_m, [v_l]_m)$" (Li and Biswas 2002, p. 678). It is given by:

$$\text{MSFVS}\big([v_i]_m, [v_l]_m\big) = \big\{ \big([v_r]_m, [v_u]_m\big) : (S_{i,l})_m \leq (S_{r,u})_m \big\}\,.$$

We denote the number of occurrence for a special value $[v_r]_m$ in the set V as $(\eta_r)_m$ with $(\eta_r)_m \in \{1, \ldots, N\}$. Than the "probability of picking a pair $([v_r]_m, [v_u]_m)$ from MSFVS $([v_i]_m, [v_l]_m)$ at random is

$$(p_r)^2_m \big([v_r]_m, [v_u]_m\big) = \frac{(\eta_r)_m \cdot \big((\eta_r)_m - 1\big)}{N \cdot (N-1)},$$

where $(\eta_r)_m = (\eta_u)_m$ as $[v_r]_m = [v_u]_m$ and N is, just as before, the total number of objects in the population" (Li and Biswas 2002, p. 678).

By summing up the picking probabilities for all pairs in MSFVS$\big([v_i]_m, [v_l]_m\big)$ the dissimilarity score $(D_{i,l})_m$ of the pair $\big([v_i]_m, [v_l]_m\big)$ is given (Li and Biswas 2002):

$$(D_{i,l})_m = \sum_{\big([v_r]_m, [v_u]_m\big) \in \text{MSFVS}\big([v_i]_m, [v_l]_m\big)} (p_r)^2_m \big([v_r]_m, [v_u]_m\big). \tag{3.3.2}$$

with $(S_{i,l})_m = 1 - (D_{i,l})_m$.

To combine the single dissimilarity values, Li and Biswas use Lancaster's mean value χ^2 transformation (Lancaster 1949):

$$(\chi)^2_{i,l} = 2 \sum_{m\text{:nominal feature}} \left(1 - \frac{\left(D_{i,l}\right)_m \ln\left(D_{i,l}\right)_m - \left(D_{i,l}\right)'_m \ln\left(D_{i,l}\right)'_m}{\left(D_{i,l}\right)_m - \left(D_{i,l}\right)'_m}\right)$$

where $\left(D_{i,l}\right)_m$ is the dissimilarity score for the nominal attribute value pair given by $\left([v_i]_m, [v_l]_m\right)$ which is the actually observed event and $\left(D_{i,l}\right)'_m$ which is the next smaller dissimilarity score in the nominal set. The factor $(\chi)^2_{i,l}$ is χ^2-distributed with T degrees of freedom where T is the number of nominal features. The significance value of this χ^2-distribution can be looked up in standard tables or approximated from the expression:

$$D_{i,l} = e^{\frac{-\chi^2_{i,l}}{2}} \sum_{t=0}^{T-1} \frac{\left(\frac{1}{2}\chi^2_{i,l}\right)^t}{t!} \tag{3.3.3}$$

The overall similarity score representing the set of T independent similarity measures is $S_{i,l} = 1 - D_{i,l}$. We used the pairwise dissimilarities $D_{i,l}$ in our tests.

(Li and Biswas 2002) used their measure for clustering different publicly available data sets from the UCI repository, cf. (Frank and Asuncion 2010), and interpreted their clustering results by using known labels for post evaluation and looking for semantic interpretations of the clustering. For the mushroom data (Lincoff 1981) using 22 nominal values the clustering achieved a perfect separation into edible and poisonous species. The heart disease data set (Detrano 1989) was clustered into two clusters that according to the medical interpretation suitably separated patients with low risk for cardiac diseases from patients with high risk. Taking into account the diagnostic labels of diseased or not, they yielded a clustering accuracy of 75.2%.

Based on Li and Biswas' results, (Luo et al. 2006) introduced the idea to define the similarity between objects by the number of clusters shared by two objects in the partitions of a clustering ensemble. For the categorical features they consider each attribute with its attribute values as a clustering on the data set. A data set with T nominal attributes has T clusterings. The co-occurrences of pairs of patterns in the same cluster votes for their association. The T clusterings are mapped into a $N \times N$ co-association matrix

$$S_{i,l} = \frac{\mathfrak{n}_{i,l}}{T} = \frac{\sum_{t=1}^{T} \mathfrak{C}^t\left(v_i, v_l\right)}{T} \tag{3.3.4}$$

where $\mathfrak{n}_{i,l}$ is the number of times the pattern pair (v_i, v_l) is assigned to the same cluster among the T clusterings, and $\mathfrak{C}^t\left(v_i, v_l\right) = 1$ if the pattern pair (v_i, v_l) is in the same cluster of the t^{th} clustering, else $\mathfrak{C}^t\left(v_i, v_l\right) = 0$.

They used their measure for mixed data successfully also for clustering the heart disease data set (Detrano 1989) from the UCI Repository (Frank and Asuncion 2010).

Using the given labels for post evaluation they yielded a clustering accuracy of 81.3%, which is significantly better than the corresponding clustering by Li and Biswas.

This dissimilarity measure from Luo et al. is one variant of the *Jaccard index* or *Jaccard coefficient* used to measure the dissimilarity of two sets A and B introduced by Jaccard in 1901 for the comparison of flower growth distributions (Jaccard 1901). This general index is given by

$$J\left(A, B\right) = \frac{|A \cap B|}{|A \cup B|}. \tag{3.3.5}$$

Where $|A|$ refers to the cardinality of the set A. The Jaccard index is for example used as measure in information retrieval applications like spelling correction in dictionaries and robust retrieval (Manning et al. 2008).

There are many more approaches towards pairwise dissimilarities. It is a very application and domain specific question how to represent those dissimilarities. In section 7.1.4 we introduce the considerations and an approach towards domain knowledge based pairwise dissimilarities for our application example Exprimage.

3.4 Embedding arbitrary dissimilarities into the Euclidean space

Many assertions about adaptive algorithms to converge to an optimum solution only hold for Euclidean dissimilarities. For other metrics or even dissimilarity measures there is no assertion that they converge to a useful solution as e.g. negative dissimilarities can corrupt the approach towards the optimum. It is a matter of ongoing research how cost functions, decision boundaries and prototype positions behave during and after learning when non-Euclidean measures are applied. Nevertheless, in most practical applications the used dissimilarities behave almost like the Euclidean distance so that it is likely that the approaches converge to an optimum solution.

To avoid this problem of non-Euclidean dissimilarities or to control the error resulting from it, it is desirable to embed arbitrary dissimilarities, e.g. given in the dissimilarity matrix, into the Euclidean space. The following sections are based on the tutorial of (Pękalska and Duin 2009) that is related to their book (Pękalska and Duin 2006). They show two main approaches of embedding dissimilarities: the so-called *dissimilarity space* and the *embedding of the dissimilarity matrix*. Furthermore we give the basics for the application of kernelization to yield an implicit embedding.

3.4.1 Dissimilarity space embedding

Given a training set V of data points $v_1, \ldots, v_K$ and a dissimilarity function d or a dissimilarity matrix D we define a subset of V as representation set. The dissimilarities to the representation set are interpreted as features for the *dissimilarity space* representations of the data points v_k. Their characteristics of dissimilarities is not used when a general classifier is applied to these dissimilarity space representations. Thus the dissimilarity matrix is considered as a set of row vectors, one for each object. They represent the objects in a vector space constructed by the dissimilarities to the other objects. The resulting vector space is endowed with the traditional inner product and the Euclidean metric. This distance is then computed on vectors defined by original dissimilarities. This can lead to changes in the nearest neighbor objects. The good side of this disadvantage is that the dissimilarity space can be used for any dissimilarity representation, including ones that are negative or asymmetric.

3.4.2 Embedding the dissimilarity matrix into the Euclidean space

Embedding the dissimilarity matrix into the Euclidean space can only be realized error free if the original set of dissimilarities are Euclidean themselves. In any other case an approximation procedure has to be used or the objects should be embedded into a non-Euclidean space. It appears that an exact embedding into a non-Euclidean space is possible for every symmetric dissimilarity matrix with zeros on the diagonal. The resulting space is the so-called pseudo-Euclidean space. Many of the dissimilarity measures used in the pattern recognition practice appear to be non-Euclidean. In this section we will concentrate on the test whether dissimilarities are Euclidean and possibilities to correct dissimilarities towards Euclidean behavior as many adaption processes require the corresponding conditions.

To formalize the underlying concepts we will give further definitions that are based on the description in Pękalska and Duin's "The dissimilarity representation for pattern recognition" (Pękalska and Duin 2006). The *Euclidean vector space* is a real vector space that is equipped with a positive definite, symmetric bilinear form. A *real vector space* over the field of real numbers $\mathbb{R}$ is a set $\mathbb{V}$ together with two binary operators (vector addition and scalar multiplication) that satisfy eight axioms: Associativity of addition, commutativity of addition, identity element of addition, inverse element of addition, distributivity of scalar multiplication with respect to vector addition, distributivity of scalar multiplication with respect to field addition, compatibility of scalar multiplication with field multiplication, and identity element of scalar multiplication. The elements of the set $\mathbb{V}$ in a vector space are also called *points*.

If we look for an Euclidean embedding of the dissimilarities given in dissimilarity matrix D, the task is to find a suitable set of vectors $x_k \in \mathbb{R}^{M'}, k = 1, \ldots, K$ that is considered as a base for an M'-dimensional embedding space $\mathbb{F}$ such that the Euclidean distances between the vectors x_k (represented in a matrix $D^{\mathbb{F}}$) are equal to the given dissimilarities in D, i.e.

$$D_{l,k} = d\left(v_l, v_k\right) \overset{!}{=} \|x_l - x_k\|^2 = D^{\mathbb{F}}_{l,k}. \qquad (3.4.1)$$

We place all x_k in rows of an $K \times M'$ matrix X. Then according to (Pękalska and Duin 2006) the *Gram matrix for the vector representation* X is defined as the matrix of inner products and given by $G = XX^\top$. They also proved that this Gram matrix can be expressed in terms of the dissimilarities $D^{\mathbb{F}}$ between the base vectors:

$$G = -\frac{1}{2} C_K \left(D^{\mathbb{F}}\right)^{\star 2} C_K \qquad (3.4.2)$$

where C_K is the *centering matrix* and $\left(D^{\mathbb{F}}\right)^{\star 2} = D^{\mathbb{F}} \circ D^{\mathbb{F}}$ is the Hadamard (entry-wise) product of the dissimilarity matrix with itself. The centering matrix is a symmetric and idempotent matrix, that, when multiplied with a vector, results in the subtraction of the mean of the vector components from all components of a vector. The K-dimensional centering matrix given by

$$C_K = I_K - \tfrac{1}{K} \mathbb{1}\mathbb{1}^\top$$

"projects the data such that the final configuration has a zero mean vector" (Pękalska and Duin 2006, p. 118). $\mathbb{1}$ is a column vector of all ones with length K and I_K is the $K \times K$ dimensional identity matrix. We can interpret the matrix $D^{\mathbb{F}}$ as the defining matrix of the bilinear form representing the dissimilarities in the embedding space $\mathbb{F}$ (cf. equation (3.1.4) in section 3.1.1).

Pękalska and Duin introduce a variety of different tests whether the given dissimilarities D show Euclidean behavior or not. We will focus on one test that directly implies possibilities of correcting the dissimilarities if they are non-Euclidean. First we calculate G according to equation (3.4.2) assuming the embedding dissimilarities in $D^{\mathbb{F}}$ to be equal to the given dissimilarities in D. Then we test G for its symmetry and positive (semi-)definiteness which ensure the existence of an Euclidean embedding of the dissimilarities. I f the embedding exists the dissimilarities are Euclidean[3].

To test G for its positive (semi-)definiteness we use the equivalent condition that G is positive definite if and only if all eigenvalues of G are positive. We calculate the

[3] This embedding exists for every configuration of dissimilarities D that according to equation (3.4.2) yield a symmetric and positive semidefinite Gram matrix G. The embedding is unique for positive definite G.

eigendecomposition of the Gram matrix $G = Q\Lambda Q^\top$, where Λ is a diagonal matrix of eigenvalues λ_k with $k = 1, \ldots, K$ and Q is the matrix of eigenvectors.

In the special case where we have positive eigenvalues followed by zero eigenvalues, the Gram matrix G is positive semidefinite. If we have $K' \leq K$ positive and no negative eigenvalues, the representation X can be calculated from $G = XX^\top$ and equation (3.4.2) "as

$$X = Q_{K'}\Lambda_{K'}^{\frac{1}{2}}$$

where $Q_{K'} \in \mathbb{R}^{K \times K'}$ is the matrix of K' leading eigenvectors, i.e. corresponding to the K' largest eigenvalues, and $\Lambda_{K'}^{\frac{1}{2}} \in \mathbb{R}^{K' \times K'}$ contains the square roots of the corresponding eigenvalues." (Pękalska and Duin 2006, p. 119)

In the case of general dissimilarities in D there are positive, negative and zero eigenvalues. Without loss of generality we assume them to be ordered such that we first have decreasing K' positive eigenvalues of G, then according to the absolute value increasing K'' negative ones, followed by zeros. Pękalska and Duin introduce four different approaches to change the dissimilarities D in order to make G positive definite:

1. The dissimilarities are not corrected directly but for calculating the Euclidean configuration X only the positive eigenvalues are taken into account, neglecting the others. There are K' non-zero eigenvalues, yielding the configuration $X = Q_{K'}\Lambda_{K'}^{\frac{1}{2}}$.

2. Determining a positive constant $\tau \geq -\lambda_{\text{min}}$ where λ_{min} is the smallest (negative) eigenvalue of G such that $D_{2\tau} = \left[D^{\star 2} + 2\tau\left(\mathbb{1}\mathbb{1}^\top - I_K\right)\right]^{\star\frac{1}{2}}$ is Euclidean. The original dissimilarities are distorted significantly if τ is a large value. In physics this approach is called spectral shift.

3. Determine a positive constant $\kappa \geq \lambda_{\text{max}}$ such that $D_\kappa = D + \kappa\left(\mathbb{1}\mathbb{1}^\top - I_K\right)$ is Euclidean. The corresponding Gram matrix has different eigenvalues and eigenvectors than the original one.

4. Determine a parameter p for the function g given in (Courrieu 2002) such that $D_p = \left(g\left(d_{i,j}; p\right)\right)$ is Euclidean. Thereby g is a nonlinear, parameter dependent transformation.

All those approaches are "useful when the negative eigenvalues are relatively small in magnitude, which suggests that the original distance measure is nearly Euclidean. In such cases, the negative eigenvalues can be interpreted as noise contributions. If the negative eigenvalues are relatively large (in magnitude), then by neglecting them, important information might be disregarded"(Pękalska and Duin 2006, pp. 121-122).

This is in parallel with the interpretation of the Eigenvalues as measure of variance in the single dimensions. As variance with a high magnitude is assumed to yield much information, neglecting Eigenvalues that are large in magnitude is assumed to introduce high information loss.

The described corrections are suitably applied to a definite, symmetric dissimilarity matrix. For an arbitrary zero-diagonal matrix D of dissimilarities the following preprocessing steps have to be conducted:

- make D definite by either
  - change all zero dissimilarities between two different objects into a small fixed value, depending on overall distances or
  - consider objects with zero dissimilarity as belonging to the same equivalence class
- make D symmetric by either
  - averaging all $d_{i,j}$ and $d_{j,i}$ or
  - taking their maximum

It is a matter of ongoing research for which dissimilarity matrices the correction of the non-Euclidean behavior is suitable. To give an orientation in this question Pękalska and Duin introduced the *Non-Euclidean Coefficient (NEC)* to measure the "amount of non-Euclidean influence"(Pękalska and Duin 2009, p. 8) in the embedding space. It is defined as:

$$NEC = \frac{\sum_{i=K'+1}^{K'+K''} |\lambda_i|}{\sum_{l=1}^{K'+K''} |\lambda_l|} \in [0; 1] \tag{3.4.3}$$

We will evaluate this coefficient for our application in chapter 7.

3.4.3 Kernelization

As opposed to the explicit embedding and calculation of the representation X it is possible to apply the *Kernel trick*, cf. (Schölkopf et al. 1999). Using a given input vector set V, a possibly nonlinear mapping function $\Phi(\cdot)$ is assumed that maps an input vector v_k from the input data space $\mathbb{R}^M$ to a feature vector $\Phi(v_k) = x_k$ in the embedding or feature space $\mathbb{F}$. We assume that the function $\mathfrak{k}_\Phi(\cdot)$ is a Mercer kernel function (Schölkopf et al. 1999) associated to the mapping Φ that can be used to calculate the inner product of two points $x_l = \Phi(v_l)$ and $x_k = \Phi(v_k)$ in the Hilbert feature space $\mathbb{F}$ by

$$\mathfrak{k}_\Phi(v_l, v_k) = \left\langle \Phi(v_l), \Phi(v_k) \right\rangle = \langle x_l, x_k \rangle \tag{3.4.4}$$

without need of the knowledge about the specific form of the nonlinear mapping $\Phi\left(\cdot\right)$. In that case, any computations in the feature space $\mathbb{F}$ can be efficiently converted into operations in the data space $\mathbb{R}^M$ through this kernel function $\mathfrak{k}_\Phi$.

Assuming the existence of such a mapping function Φ with the corresponding kernel function $\mathfrak{k}_\Phi$ we can express the prototype vectors as linear combinations of the images of the input vectors in the feature space $\mathbb{F}$ according to

$$w_n^{\mathbb{F}} = \sum_{k=1}^{K}[\beta_n]_k \Phi\left(v_k\right) = \sum_{k=1}^{K}[\beta_n]_k x_k \tag{3.4.5}$$

where $\sum_{k=1}^{K}[\beta_n]_k = 1$ for all n and $[\beta_n]_k$ is the k^{th} component of the prototype representing coefficient vector $\vec{\beta_n}$ for prototype $w_n^{\mathbb{F}}$. The dissimilarity in the feature space $\mathbb{F}$ between a projected sample $\Phi\left(v_k\right)$ and the feature space prototype vector $w_n^{\mathbb{F}}$ represented by the coefficients $\vec{\beta}_n$ can be formulated as:

$$d_{k,n}^{\mathbb{F}} = \mathfrak{k}_\Phi\left(v_k, v_k\right) - 2\cdot\sum_{l=1}^{K}[\beta_n]_l \mathfrak{k}_\Phi\left(v_k, v_l\right) + \sum_{i,u=1}^{K}[\beta_n]_i[\beta_n]_u \mathfrak{k}_\Phi\left(v_i, v_u\right). \tag{3.4.6}$$

The kernel function can for example be chosen to be the Gaussian kernel function given by

$$\mathfrak{k}_\Phi\left(v_i, v_u\right) = \exp\left(-\frac{\|v_i - v_u\|^2}{2\sigma^2}\right) \tag{3.4.7}$$

as used by (Qinand and Suganthan 2004) in their kernelized version of the Generalized Learning Vector Quantization approach.

As the kernel matrix $\mathfrak{K} = \left(\mathfrak{k}_\Phi\left(v_i, v_u\right)\right)$ is a matrix of inner products, it can also be interpreted as Gram matrix to the base $X = (x_1, \ldots, x_K)$ and thus the relation given in equation (3.4.2) can be used to reformulate equation (3.4.6) in terms of dissimilarities:

$$d_{k,n}^{\mathbb{F}} = \left\|w_n^{\mathbb{F}} - \Phi\left(v_k\right)\right\|^2 = \left(D^{\mathbb{F}}\cdot\vec{\beta}_n\right)_k - \frac{1}{2}\cdot\vec{\beta}_n^{\top}\cdot D^{\mathbb{F}}\cdot\vec{\beta}_n \tag{3.4.8}$$

where $D^{\mathbb{F}} = \left(d_{k,l}^{\mathbb{F}}\right) = \left\|\Phi\left(v_k\right) - \Phi\left(v_l\right)\right\|^2$ with $v_k, v_l \in V$ are the Euclidean distances in the feature space $\mathbb{F}$.

If dissimilarities $D = (d_{k,l}) = d\left(v_k, v_l\right)$ are given for the input vectors $v_k, v_l \in V$ in the input space, we further assume the mapping Φ to be isometric, i.e. that the Euclidean distances in the feature space $D^{\mathbb{F}}$ are equal to the dissimilarities D given for the input space. By adapting the coefficient vectors in accordance with the given dissimilarities D, an implicit representation in the high-dimensional feature space $\mathbb{F}$ is learned. We discuss the corresponding unsupervised method Relational Neural Gas in section 4.2.1 and the supervised approach Kernel Learning Vector Quantization in section 4.2.2.

Chapter 4

Vector Quantization

In the following sections we will give definitions and details of a choice of Vector Quantization (VQ) based algorithms, as well as their evaluation. Throughout the chapter we will call the unsupervised methods Vector Quantization (VQ) methods and the supervised approaches are referred to as Learning Vector Quantization (LVQ)[1] methods. The sections show algorithms that are approaches for learning using a dissimilarity function, relational learning and median learning. In every section we will first consider unsupervised algorithms, turning then to supervised algorithms. Table 4.1 gives the overview over the introduced algorithms and the order in which they will be given. Additionally, it provides information about the supervision modality and mapping kind of the algorithms.

4.1 VQ based learning using a dissimilarity function

In this first section of Vector Quantization based algorithms we will consider all the algorithms that need some kind of dissimilarity that can be determined at time according to a given function or procedure. So for every possible point in the input space we can calculate the dissimilarity to any other possible point in the input space and thus the prototypes can be placed on arbitrary positions in the input space. We will start with the unsupervised variants in ascending structural complexity and dissimilarity function including a metric adaptation variant and a fuzzy variant, proceeding with supervised algorithms in the same order, again including a metric adaptation variant.

4.1.1 Unsupervised variants of VQ using a dissimilarity function

We look at unsupervised variants of VQ algorithms starting with the simplest approach: Standard Vector Quantization. Then a model inspired by physics, Neural Gas in its batch version, is introduced, followed by its corresponding dissimilarity

[1] In publications by Bezdek (e.g. (Bezdek 1981)) the term Learning Vector Quantization is also used for unsupervised methods. This is in contrast to the conventional terminology used here.

Learning Type	Name	metric adaptation	unsupervised	supervised	crisp	fuzzy	online	batch
Learning using a dissimilarity measure	Standard VQ		❙		❙		❙	
	Batch Neural Gas		❙		❙			❙
	Matrix Neural Gas	❙	❙		❙			❙
	Fuzzy C-Means		❙			❙		❙
	Kohonen's LVQ			❙	❙		❙	
	Generalized LVQ			❙	❙		❙	
	Gen. Relevance LVQ	❙		❙	❙		❙	
	Gen. Matrix LVQ	❙		❙	❙		❙	
Relational learning	Relational Batch NG		❙		❙			❙
	Kernel LVQ			❙	❙		❙	
Median learning	Median C-Means		❙		❙			❙
	Affinity Propagation		❙		❙			❙
	M. Fuzzy C-Means		❙			❙		❙
	Fuzzy Affinity Prop.		❙			❙		❙

Table 4.1: *Overview of Vector Quantization based algorithms for clustering and classification considered in this thesis and their properties*

adaptation variant Matrix Batch Neural Gas. The last algorithm discussed in this section is the Fuzzy C-Means as a variant developed for fuzzy mappings.

Standard Vector Quantization – SVQ

Standard Vector Quantization is a basic approach to encode or compress data using simple competitive learning. The data are given as input vectors $v_k \in V$ in an input space[2]. A codebook which is defined by a set W of N prototype vectors w_n in the same space is assumed. The number of prototypes N is chosen by the user and the initial positions of the prototype vectors in the input space are often chosen randomly[3]. In the first step of VQ – the learning or inference stage – the prototype

[2] usually $V \subseteq \mathbb{R}^M$

[3] We refer to section 4.4.2 for other prototype position initialization possibilities.

vectors are adapted to minimize some cost or energy function. The cost function that is referred to by Standard Vector Quantization in common sense is given by:

$$E_{\text{SVQ}} = \int p(v) \cdot d\left(v, w_s\right)^2 \, \mathrm{d}v$$

where $p(v)$ is the probability density of the data in the input space and $d\left(v, w_s\right)^2$ is the squared dissimilarity of the input v to the closest prototype w_s – the *winner* with index s according to equation (2.2.1). Optimizing this energy function with respect to the prototype positions w_n with $n = 1, \ldots, N$ leads to prototypes which represent the feature space proportional to the data density $P\left(v\right)$. For a finite training set of K input vectors $V_{\text{Tr}} \subseteq V$ the cost function specified above reduces to

$$E_{\text{SVQ}} = \sum_{k=1}^{K} d\left(v_k, w_s\right)^2$$

We assume $V_{\text{Tr}} = V$ throughout the next sections and for reasons of simplicity will only denote it by V. In the VQ implementation of online learning, randomly chosen input vectors from the set V are "presented" to the prototypes one by one. For every training input vector $v_k \in V$ the nearest prototype w_s is adapted towards the training input vector according to

$$\Delta w_s = \epsilon_w \cdot \left(v_k - w_s\right) \tag{4.1.1}$$

where ϵ_w is the learning rate controlling the adaption strength. The approach is summarized in algorithm 4.1.1.

Algorithm 4.1.1 *Standard Vector Quantization – SVQ*

```
initialize prototype positions $w_n \in W$ for all $n = 1, \ldots, N$
repeat
    randomly chose an input vector $v_k$ from $V$
    determine the winning prototype $w_s$ according to equation (2.2.1)
    adapt the winning prototype's position according to equation (4.1.1)
until convergence
```

In the decision stage newly incoming input vectors are mapped to the cluster indexes according to the mapping function Ψ, that was introduced in equation (2.2.1) in section 2.2. If the used dissimilarity measure is the Euclidean distance this has the effect that the receptive fields of the prototypes divide the input vector space into a Voronoi tesselation, for definition cf. e.g. (Hastie, T. et al. 2003).

Batch Neural Gas – BNG

Neural Gas (NG) is an unsupervised algorithm for finding compressed data representations of feature vectors. The basic idea of NG was introduced by (Martinetz and Schulten 1991). The name "Neural Gas" was inspired by the dynamics of the prototype vectors during the adaptation process, which distribute themselves like an ideal gas within the data space.

In contrast to SVQ where only one prototype vector is adapted, in the Neural Gas adaption scheme all prototype vectors are adapted according to their dissimilarity rank with respect to a specific input vector. This Neural Gas approach is "a very robust clustering algorithm given Euclidean data which does not suffer from the problem of local minima like simple vector quantization" (Cottrell et al. 2006, p. 2).

Batch Neural Gas was introduced by (Cottrell et al. 2006) as extension of NG towards a batch adaption scheme. It is based on the cost function:

$$E_{\text{BNG}} = \sum_{n=1}^{N} \sum_{k=1}^{K} h_\sigma\big(r_W\left(v_k, w_n\right)\big) \cdot d\left(v_k, w_n\right)^2 \tag{4.1.2}$$

where d is a dissimilarity function, N is the number of prototype vectors $w_n \in W$ and K is the number of input points $v_k \in V$. Furthermore

$$r_W\left(v_k, w_n\right) = r_{k,n} = \Big|\big\{w_l : d\left(v_k, w_l\right) < d\left(v_k, w_n\right)\big\}\Big| \tag{4.1.3}$$

is the dissimilarity rank of prototype w_n with respect to input vector v_k, given by the number of prototypes that are closer to the input vector v_k then w_n itself. The rank $r_{k,n}$ for some special pair (v_k, w_n) according to the current set of prototype vectors W can also be expressed in terms of Heaviside step functions

$$r_{k,n} = \sum_{l=1}^{N} \Theta\left(d\left(v_k, w_n\right)^2 - d\left(v_k, w_l\right)^2\right)$$

with $w_l \in W$ and $\Theta(x)$ the Heaviside step function, defined as 0 for $x \leq 0$ and 1 for $x > 0$, cf. (Martinetz et al. 1993) for details. In Equation (4.1.2) the function h_σ defines a neighborhood on the ranks with range parameter σ^2. Frequently a normal Gaussian with σ^2 variance is used as neighborhood function.

In BNG the cost function is optimized via a Newton scheme. For this purpose the cost function E_{BNG} is interpreted as a function depending only on w_n and $r_{k,n}$ (Cottrell et al. 2006). It is optimized in turn with respect to the now hidden variables $r_{k,n}$ and the prototypes w_n, with the constraint that the values $r_{k,n}$ $(n = 1, \ldots, N)$ constitute a permutation of $0, ..., N-1$ for each point v_k. That means that ties have to be broken. For the ranks we yield the update rule given in equation (4.1.3). The prototype positions are updated as

$$w_n = \frac{\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \cdot v_k}{\sum_{l=1}^{K} h_\sigma\left(r_{l,n}\right)}. \tag{4.1.4}$$

A summary of the BNG steps is given in algorithm 4.1.2.

Algorithm 4.1.2 *Batch Neural Gas – BNG*

```
initialize prototype positions $w_n \in W$ for all $n = 1, \dots, N$
repeat
    determine the rank of the given prototype positions $w_n \in W$ according to
    equation (4.1.3)
    based on the hidden variables $r_{k,n}$ set new prototpye positions according to
    equation (4.1.4)
until convergence
```

(Cottrell et al. 2006) proved that Batch NG can be interpreted as Newton optimization method, which takes second order information into account. Usually only a few adaptation steps are necessary for convergence. Online NG in contrast is given by a simple stochastic gradient descent.

Matrix Neural Gas – MNG

In BNG frequently the basic interpretation of dissimilarity, the Euclidean metric, is used. Matrix Neural Gas, as introduced by (Arnonkijpanich and Hammer 2010), extends the mapping abilities of BNG by locally adapting a real quadratic form as given in section 3.1.1 by equation (3.1.5):

$$d_{\Lambda_n}\left(v, w_n\right) = \left(v - w_n\right)^\top \Lambda_n \left(v - w_n\right) \tag{4.1.5}$$

where the Λ_n are associated to the prototypes w_n.

In a global variant all Λ_n are constrained to be equal. The constraints of symmetry, positive definiteness and a unity determinant for every parameter matrix "are necessary to guarantee that the resulting formula defines a metric which does not degenerate to a trivial form" (Arnonkijpanich and Hammer 2010, p. 87). $\Lambda_n = 0$ constitutes an obvious trivial optimum of the cost function.

Replacing the dissimilarity in the BNG cost function, we obtain the cost function of MNG

$$E_{\text{MNG}} = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} h_\sigma\left(r_W\left(v_k, w_n\right)\right) \cdot d_{\Lambda_n}\left(v_k, w_n\right) \tag{4.1.6}$$

where $r_W(v_k, w_n)$ is the dissimilarity rank of prototype w_n to a specific input vector v_k as introduced in equation (4.1.3) for BNG, but with the accordingly modified dissimilarity:

$$r_W(v_k, w_n) = r_{k,n} = \left|\{w_l \middle| d_{\Lambda_l}(v_k, w_l) < d_{\Lambda_n}(v_k, w_n)\}\right| \tag{4.1.7}$$

Performing again a Newton optimization on this cost function (accordingly interpreting the other values as hidden variables) gives the following update rule for the prototype positions

$$w_n = \frac{\sum_{k=1}^{K} h_\sigma(r_{k,n})\, v_k}{\sum_{l=1}^{K} h_\sigma(r_{l,n})} \tag{4.1.8}$$

and for the parameter matrix

$$\Lambda_n = S_n^{-1}(\det S_n)^{\frac{1}{M}} \text{ where } S_n = \sum_{k=1}^{K} h_\sigma(r_{k,n})(v_k - w_n)(v_k - w_n)^\top . \tag{4.1.9}$$

For detailed derivation of the update rules we refer to (Arnonkijpanich et al. 2011). All update steps are summarized in algorithm 4.1.3. For locally adapted MNG "ellipsoidal cluster shapes arise which are aligned according to local principal components of the data." (Arnonkijpanich and Hammer 2010, p. 87)

Algorithm 4.1.3 *Matrix Neural Gas – MNG*

```
initialize prototype positions $w_n \in W$ for all $n = 1, \dots, N$
repeat
    determine the rank of the prototypes $w_n \in W$ according to equation (4.1.7)
    based on the hidden variables $r_{k,n}$ update the prototype positions $w_n$ according to equation (4.1.8)
    based on the hidden variables $r_{k,n}$ and $w_n$ update the parameter matrix $\Lambda$ according to equation (4.1.9)
until convergence
```

Fuzzy C-Means – FCM

Fuzzy C-Means – an unsupervised algorithm – was first introduced by (Dunn 1973) and later reformulated by (Bezdek 1981). As introduced in section 2.2 and equation (2.2.2) the basic idea of a fuzzy mapping is that an input vector is no longer mapped to one single prototype index but rather to a vector of "membership degrees" where $\psi_{w_n}(v_k)$ is the assignment degree of input v_k to the prototype with index n.

Thereby the $\psi_{k,n} = \psi_{w_n}(v_k)$ lie in the interval $[0,1]$. The larger the $\psi_{k,n}$ the higher is the assignment degree.

Bezdek introduced the objective function of FCM to be:

$$E_{\text{FCM}} = \sum_{n=1}^{N}\sum_{k=1}^{K} (\psi_{k,n})^{\mathfrak{f}}\, d_{k,n}^2 \tag{4.1.10}$$

where $d_{k,n}^2 = (v_k - w_n)^\top (v_k - w_n)$ stands for the squared Euclidean distance between the training input vector v_k and the prototype w_n. The "fuzzifier" $\mathfrak{f} > 1$ controls how fuzzy the prototype memberships are. In the limit of $\mathfrak{f} \to \infty$ for the assignment degrees it holds that $\psi_{k,n} = \frac{1}{N}$, i.e. the membership degree is equal for all prototypes. If $\mathfrak{f}$ is chosen near to one, mapping is done rather crisp with the membership degrees being either near one or near zero, a choice of $\mathfrak{f} = 2$ often has proven suitable in praxis (Geweniger et al. 2010).

The prototype positions and the membership degrees are determined by minimizing the objective function (4.1.10). This optimization is done under two constraints:

1. For every input vector the sum of the membership degrees is equal to one: $\sum_{n=1}^{N} \psi_{k,n} = 1, \forall k$.
2. The fuzzy receptive fields $\Upsilon_n^{\mathcal{F}}$ (cf. equation (2.2.5)) of the prototypes w_n are non-empty: $\sum_{k=1}^{K} \psi_{k,n} > 0, \forall n$.

For solving the constraint minimization problem the Lagrange function

$$\mathfrak{L}(\Psi^{\mathcal{F}}, W, \lambda) = \sum_{n=1}^{N}\sum_{k=1}^{K} (\psi_{k,n})^{\mathfrak{f}}\, d_{k,n}^2 - \sum_{k=1}^{K}\left(\lambda_k\left(\sum_{n=1}^{N}\psi_{k,n} - 1\right)\right) \tag{4.1.11}$$

with $W = \{w_1, \ldots, w_N\}$ and Lagrange multipliers $\lambda = (\lambda_1, \ldots, \lambda_K)$ is considered. This yields the following update rules for the prototype positions

$$w_n = \frac{\sum_{k=1}^{K} (\psi_{k,n})^{\mathfrak{f}}\, v_k}{\sum_{j=1}^{K} \psi_{j,n}^{\mathfrak{f}}} \tag{4.1.12}$$

as well as for the assignment degrees

$$\psi_{k,n} = \frac{1}{\sum_{j=1}^{N}\left(\frac{d_{k,n}}{d_{k,j}}\right)^{\frac{2}{\mathfrak{f}-1}}}. \tag{4.1.13}$$

All together we obtain the FCM algorithm as given in algorithm 4.1.4.

There is a huge variety of FCM variants, e.g. the assignment degrees can also be interpreted in a possibilistic manner, cf. (Pal et al. 2005).

Algorithm 4.1.4 *Fuzzy C-Means – FCM*

```
initialize assignment degrees $\psi_{k,n}$ for all $k, n$ with $\sum_{n=1}^{N} \psi_{k,n} = 1$ and $\sum_{k=1}^{K} \psi_{k,n} >$
0
repeat
  determine the prototype positions according to equation (4.1.12)
  determine the assignment degrees according to equation (4.1.13)
until convergence
```

4.1.2 Supervised variants of VQ using a dissimilarity function

In this section we consider the supervised variants. Starting with the simplest approach of Learning Vector Quantization we go further on with the Generalized Learning Vector Quantization and its dissimilarity adapting variants Generalized Relevance Learning Vector Quantization and Generalized Matrix Learning Vector Quantization.

Kohonen's Learning Vector Quantization – LVQ1

Kohonen's Learning Vector Quantization as introduced by (Kohonen 1986) is a family of heuristic LVQ schemes used for crisp classifications. We recall that for crisp classification a set of class labels $Z = \{z_1, \ldots, z_C\}$ is predefined and that the set of input vectors V is labeled, i.e. for every $v_k \in V$ there exists a $z_{v_k} \in Z$. The training set is iterated and the input vectors are presented to the prototypes. For every input vector the nearest prototype w_s according to some dissimilarity measure d – that is commonly interpreted as Euclidean distance – is determined, cf. equation (2.2.1).

The most prominent variant of Kohonen's Learning Vector Quantization is LVQ1. It updates the prototype positions as follows: If the winning prototype w_s has the same class label as the input vector v_k under consideration, this winning prototype is adapted towards the training vector. If in contrast the closest prototype has a different class label it is repelled from the training vector.

This can be formalized as

$$\Delta w_s = \begin{cases} \epsilon_w \cdot (v_k - w_s), & \text{if } z_{w_s} = z_{v_k} \\ -\epsilon_w \cdot (v_k - w_s), & \text{if } z_{w_s} \neq z_{v_k} \end{cases} \tag{4.1.14}$$

The adaption strength is controlled by the learning rate ϵ_w. Often the learning rate is defined as decaying over time (number of epochs t), i.e. $\epsilon_w = f(t)$ with $\epsilon_w(t_1) > \epsilon_w(t_2)$ if $t_1 < t_2$. In 1978 Kushner and Clark proved that convergence of stochastic approximation methods can be guaranteed if the following set of conditions holds for the decay of the learning rate:

$$\lim_{t\to\infty} \epsilon_w\left(t\right) = 0 \tag{4.1.15}$$

$$\sum_t \epsilon_w = \infty \tag{4.1.16}$$

$$\sum_t \epsilon_w^2 < \infty \tag{4.1.17}$$

For a detailed discussion on the choice of the learning rate we refer to section 4.4.4. The update rules of LVQ1 are summarized in algorithm 4.1.5.

Algorithm 4.1.5 *Learning Vector Quantization – LVQ1*

```
initialize the prototype positions $w_n \in W$ for all $n = 1, \ldots, N$
repeat
  randomly choose an input vector $v_k$ from $V$ with its label $z_{v_k}$
  determine the winning prototype $w_s$ according to equation (2.2.1) (ties are broken
  arbitrarily), where $d$ in general is interpreted as Euclidean distance
  determine new prototype position for $w_s$ according to equation (4.1.14)
until convergence
```

Generalized Learning Vector Quantization – GLVQ

As mentioned before the LVQ1 update rules are heuristically motivated and not explicitly based on the optimization of a cost function. In contrast to this (Sato and Yamada 1996) developed Generalized Learning Vector Quantization to optimize a cost function that approximates the classification error and allows to use gradient descent. It considers not only one nearest prototype but selects two prototypes for adaptation: the nearest prototype with the same class label as the considered input vector and the nearest prototype with a different label.

Let $W_{v_k}^+$ be the set of prototype vectors, that have the same class label as the input vector v_k, i.e.

$$W_{v_k}^+ = \{w_n : z_{v_k} = z_{w_n}\}$$

and accordingly we define $W_{v_k}^-$ to be

$$W_{v_k}^- = \{w_n : z_{v_k} \neq z_{w_n}\}\,.$$

Then we can formalize the considered prototypes for the update as

$$w_+^{v_k} = \arg \min_{w_{n^+} \in W_{v_k}^+} \big(d\left(v_k, w_{n^+}\right)\big) \tag{4.1.18}$$

and

$$w_{-}^{v_k} = \arg \min_{w_{n^-} \in W_{v_k}^{-}} \left(d \left(v_k, w_{n^-} \right) \right). \tag{4.1.19}$$

As in most cases the relation to the corresponding input vector v_k is clear we drop this explicit notation for the sake of readability and refer to the considered prototype vectors as w_+ and w_-.

To optimize the classifier's hypothesis margin[4] the *relative difference distance*

$$\mu_k = \frac{d^+ \left(v_k \right) - d^- \left(v_k \right)}{d^+ \left(v_k \right) + d^- \left(v_k \right)} \tag{4.1.20}$$

is incorporated into learning. For sake of readability we abbreviated $d \left(v_k, w_+ \right)$ by $d^+ \left(v_k \right)$ and $d^- \left(v_k, w_- \right)$ by $d^- \left(v_k \right)$ respectively.

The value μ_k gives a measure of the prototype-based classification confidence (Schneider 2010). In the case where the numerator is smaller than 0, the classification of the data point is correct. A correct classification decision is more certain, the smaller the numerator is because the difference of the dissimilarity between the closest correct and wrong prototype is large. The numerator term is scaled by the denominator to satisfy $-1 < \mu_k < 1$.

Sato and Yamada additionally defined a sigmoid loss function taking this relative difference distance as input

$$L(\mu_k) = \left(1 + \exp(-\mu_k) \right)^{-1}. \tag{4.1.21}$$

It determines the "active region of the algorithm". With this sigmoid function training samples lying close to the decision boundary influence learning more. They are assumed to carry most information (Schneider 2010). The cost function is the sum of the loss functions of the relative difference distances for all training data points:

$$E_{\text{GLVQ}} = \sum_{k=1}^{K} L \left(\mu_k \right) = \sum_{k=1}^{K} L \left(\frac{d^+ \left(v_k \right) - d^- \left(v_k \right)}{d^+ \left(v_k \right) + d^- \left(v_k \right)} \right) \tag{4.1.22}$$

where K is the number of the training data points used.

The prototype positions w_+ and w_- are iteratively optimized with respect to the cost function after every "presentation" of a training input vector, i.e. GLVQ is an

[4] Assuming some arbitrary input vector and interpreting the given dissimilarity in terms of distance the hypothesis margin is the largest distance the prototype can travel without altering the label of the presented input vector (Crammer et al. 2002).

online learning approach. The update rules are given according to the stochastic gradients on the cost function:

$$\Delta w_+ = \epsilon_w \cdot L'(\mu_k) \cdot \frac{4 \cdot d^-(v_k)}{\big(d^+(v_k) + d^-(v_k)\big)^2} \cdot (v_k - w_+) \tag{4.1.23}$$

and

$$\Delta w_- = -\epsilon_w \cdot L'(\mu_k) \cdot \frac{4 \cdot d^+(v_k)}{\big(d^+(v_k) + d^-(v_k)\big)^2} \cdot (v_k - w_-) \tag{4.1.24}$$

where the *learning rate* ϵ_w controls the adaptation strength. Algorithm 4.1.6 summarizes the learning steps.

Algorithm 4.1.6 *Generalized Learning Vector Quantization – GLVQ*

```
initialize the prototype positions $w_n \in W$ for all $n = 1, \dots, N$
repeat
    randomly choose an input vector $v_k$ from $V$ with its label $z_{v_k}$
    determine $w_+$ and $w_-$ according to $v_k$ and $z_{v_k}$ and dissimilarity $d$
    determine new prototype position for $w_+$ according to equation (4.1.23)
    determine new prototype position for $w_-$ according to equation (4.1.24)
until convergence
```

Minimizing with respect to the GLVQ criterion using the squared Euclidean metric, the optimal Bayesian boundaries are approximated by the piecewise linear boundaries of the receptive fields of all prototypes (Schneider 2010).

Generalized Relevance Learning Vector Quantization – GRLVQ

The GLVQ algorithm as introduced before does not perform well in cases where the squared Euclidean metric is not appropriate for the data or where the vector dimensions are unequally scaled or unequally important, e.g. unequally subject to noise. To overcome this problem (Hammer and Villmann 2002) introduced Generalized Relevance Learning Vector Quantization, where the dissimilarity term given in GLVQ is replaced by the parameterized variant of the Euclidean distance introduced in equation (3.1.7) in section 3.1.1 as

$$d_\alpha(v, w) = \sum_{m=1}^{M} \alpha_m \big([v]_m - [w]_m\big)^2$$

with $\vec{\alpha} = (\alpha_1, \dots, \alpha_M)$, $\alpha_m \geq 0$ and $\sum_{m=1}^{M} \alpha_m = 1$. They focus on global dissimilarity adaptation and thus one single parameter vector $\vec{\alpha}$ is learned for all prototypes.

The cost function of GLVQ, cf. equation (4.1.22), changes by this dissimilarity measure and is given by

$$E_{\text{GRLVQ}} = \sum_{k=1}^{K} L\left(\mu_\alpha^k\right) = \sum_{k=1}^{K} L\left(\frac{d_\alpha^+\left(v_k\right) - d_\alpha^-\left(v_k\right)}{d_\alpha^+\left(v_k\right) + d_\alpha^-\left(v_k\right)}\right) \tag{4.1.25}$$

with $d_\alpha^+\left(v_k\right)$ and $d_\alpha^-\left(v_k\right)$ defined as $d^+\left(v_k\right)$ and $d^-\left(v_k\right)$ before with the additional dissimilarity parameter vector $\vec{\alpha}$.

Hammer and Villmann emphasize the interpretation of the dissimilarity parameters in terms of relevance for the classification process. Appropriate values for the dissimilarity parameters α_m are determined by stochastic gradient learning:

$$\Delta\alpha_m = -\epsilon_\alpha \cdot L'\left(\mu_\alpha^k\right) \cdot \left(\left(\mu_\alpha^{k,+}\right)'\left(\left[v_k\right]_m - \left[w_+\right]_m\right)^2 - \left(\mu_\alpha^{k,-}\right)'\left(\left[v_k\right]_m - \left[w_-\right]_m\right)^2\right). \tag{4.1.26}$$

where $\left(\mu_\alpha^{k,+}\right)'$ is now given by

$$\left(\mu_\alpha^{k,+}\right)' = \frac{4 \cdot d_\alpha^-\left(v_k\right)}{\left(d_\alpha^+\left(v_k\right) + d_\alpha^-\left(v_k\right)\right)^2}$$

and $\left(\mu_\alpha^{k,-}\right)'$ is determined accordingly. ϵ_α is the learning rate controlling the adaptation strength of the dissimilarity. As the prototypes require a stationary dissimilarity measure, it has been suggested to perform the $\vec{\alpha}$ adaption using a smaller learning rate $\epsilon_\alpha \ll \epsilon_w$, see (Kato 1950) for the theoretic foundations and section 4.4.5. Additionally, the metric parameters in $\vec{\alpha}$ have to be renormalized after every iteration, so that they again sum up to one.

The prototype updates as given for GLVQ in equation (4.1.23) and (4.1.24) change accordingly:

$$\Delta w_+ = +\epsilon_w \cdot L'\left(\mu_\alpha^k\right) \cdot \left(\mu_\alpha^{k,+}\right)' \cdot \vec{\alpha} \circ \left(v_k - w_+\right) \tag{4.1.27}$$

and

$$\Delta w_- = -\epsilon_w \cdot L'\left(\mu_\alpha^k\right) \cdot \left(\mu_\alpha^{k,-}\right)' \cdot \vec{\alpha} \circ \left(v_k - w_-\right) \tag{4.1.28}$$

where ϵ_w is the learning rate for the prototype vectors and the derivatives $\left(\mu_\alpha^{k,+}\right)'$ and $\left(\mu_\alpha^{k,-}\right)'$ are given as before. Elementwise multiplication is denoted by $\circ$, i.e. the Hadamard product. For details of the derivation we refer to (Hammer and Villmann 2002). All rules for GRLVQ are summarized in algorithm 4.1.7.

To interpret the vector $\vec{\alpha}$ in terms of relevance for the classification process, the feature dimensions have to be normalized to a common variance. Furthermore, it is suggested to run the algorithm several times and analyze the variation of the resulting vectors $\vec{\alpha}$.

Algorithm 4.1.7 *Generalized Relevance Learning Vector Quantization – GRLVQ*

```
initialize the prototype positions $w_n \in W$ for all $n = 1, \dots, N$
repeat
    randomly choose an input vector $v_k$ from $V$ with its label $z_{v_k}$
    determine $w_+$ and $w_-$ according to $v_k$ and $z_{v_k}$ and $d_\alpha (v_k, w_n)$ from equation (3.1.7)
    determine new prototype position for $w_+$ with equation (4.1.27)
    determine new prototype position for $w_-$ with equation (4.1.28)
    determine new dissimilarity defining vector $\vec{\alpha}$ with equation (4.1.26)
    renormalize metric parameters
until convergence
```

Generalized Matrix Learning Vector Quantization – GMLVQ

(Biehl et al. 2006) introduced the idea that later was elaborated as Generalized Matrix Learning Vector Quantization (e.g. in Schneider's PhD thesis on "Advanced methods for prototype-based classification" (Schneider 2010)) to use the quadratic form, as given in equation (3.1.5), in the GLVQ learning scheme. As discussed in section 3.1.1, symmetry and positive semi-definiteness for this dissimilarity measure are fulfilled when substituting the parameter matrix Λ according to equation (3.1.6) by $\Lambda = \Omega^\top \Omega$. To guarantee positive definiteness additionally $\det \Lambda \neq 0$ has to be enforced. Schneider states in her thesis that "in practice, positive semi-definiteness of the matrix is sufficient, since data often only populates a sub-manifold of the full data space and definiteness has to hold only with regard to the relevant subspace of data" (Schneider 2010, p. 17). We consider here only the global variant of dissimilarity adaption, where one single parameter matrix Λ is adapted.

Introducing the matrix form into the cost function of GLVQ, cf. equation (4.1.22), the cost function of GMLVQ is obtained as:

$$E_{\text{GMLVQ}} = \sum_{k=1}^{K} L\left(\mu_\Lambda^k\right), \text{ with } \mu_\Lambda^k = \frac{d_\Lambda^+ (v_k) - d_\Lambda^- (v_k)}{d_\Lambda^+ (v_k) + d_\Lambda^- (v_k)} \tag{4.1.29}$$

where $d_\Lambda^+ (v_k)$ and $d_\Lambda^- (v_k)$ are defined as before for GLVQ but using the quadratic form with parameter Λ as dissimilarity measure.

The update rules summarized in algorithm 4.1.8 for the prototype vectors and the metric parameters w_+, w_- and $\Omega_{l,m}$ are given by

$$\Delta w_+ = + \epsilon_w \cdot L'\left(\mu_\Lambda^+ (v_k)\right) \cdot \left(\mu_\Lambda^{k,+}\right)' \cdot \Lambda \cdot (v_k - w_+) \tag{4.1.30}$$

and

$$\Delta w_{-} = -\,\epsilon_w \cdot L'\left(\mu_\Lambda^{-}\left(v_k\right)\right) \cdot \left(\mu_\Lambda^{k,-}\right)' \cdot \Lambda \cdot \left(v_k - w_{-}\right) \tag{4.1.31}$$

as well as

$$\begin{aligned}\Delta\Omega_{l,m} = &-\,\epsilon_\Omega \cdot L'\left(\mu_\Lambda^{+}\left(v_k\right)\right) \cdot \\ &\left(\left(\mu_\Lambda^{k,+}\right)' \cdot \left(\left(\left[v_k\right]_m - \left[w_{+}\right]_m\right)\left[\Omega\left(v_k - w_{+}\right)\right]_l\right) - \right. \\ &\left.\left(\mu_\Lambda^{k,-}\right)' \cdot \left(\left(\left[v_k\right]_m - \left[w_{-}\right]_m\right)\left[\Omega\left(v_k - w_{-}\right)\right]_l\right)\right)\end{aligned} \tag{4.1.32}$$

where ϵ_w, ϵ_Ω and $\left(\mu_\Lambda^{k,-}\right)'$ and $\left(\mu_\Lambda^{k,-}\right)'$ are defined as for GRLVQ but with the quadratic form with parameter Λ as dissimilarity measure. We refer to (Schneider 2010) for a detailed derivation. In practical applications the learning rates are often set to satisfy $\epsilon_\Omega \ll \epsilon_w$.

Algorithm 4.1.8 *Generalized Matrix Learning Vector Quantization – GMLVQ*

```
initialize the prototype positions $w_n$
repeat
  randomly choose an input vector $v_k$ from $V$ with its label $z_{v_k}$
  determine $w_+$ and $w_-$ according to $v_k$ and $z_{v_k}$ and $d_\Lambda(v_k, w_n)$
  determine new prototype position for $w_+$ with equation (4.1.30)
  determine new prototype position for $w_-$ with equation (4.1.31)
  determine new metric defining matrix $\Omega$ with equation (4.1.32)
  renormalize metric parameters
until convergence
```

Normalization is achieved by dividing the elements of the parameter matrix Λ by $\sqrt{\sum_{l,m}\left(\Omega_{l,m}\right)^2}$ (Schneider 2010). A local variant of the GMLVQ can be derived analogously and yields a higher degree of flexibility (Schneider et al. 2009).

4.2 VQ based relational learning

The two approaches discussed in the following section are applicable if only pairwise dissimilarities are given for the input vectors. Assuming that the input vectors that induce these dissimilarities are embeddable into the Euclidean space (for details cf. section 3.4) we can find representations of prototypes in this embedding space.

No explicit knowledge about the input vectors is needed. In relational learning they are often referred to as data points. In this case we refer to the matrix of pairwise data dissimilarities as relational data. The embedding trick is used in the unsupervised approach Relational Neural Gas. In the kernelized version of Learning Vector Quantization (KLVQ) the assumption is used that the embedding space is a Hilbert space.

4.2.1 Unsupervised variant of VQ based relational learning

As unsupervised variant in the VQ based relational learning we consider Relational Neural Gas that was developed as an extension of BNG.

Relational Neural Gas – RNG

The approaches BNG and MNG where designed for the use of explicit Euclidean based dissimilarity functions. (Hammer and Hasenfuss 2007) introduced Relational Neural Gas for learning using relational data. For this algorithm we assume the existence of a mapping Φ from the data space into a feature space $\mathbb{F}$ such that equation (3.4.1) holds, i.e. the given pairwise relational dissimilarities $D = \big(d\left(v_k, v_l\right)\big)$ are equal to the Euclidean distances of the mapped data points $D^{\mathbb{F}} = \|\Phi\left(v_k\right) - \Phi\left(v_l\right)\|^2$.

For RNG the prototypes are expressed in terms of the given data points $v_k \in V$ as given in equation (3.4.5) and explained in detail in section 3.4.3:

$$w_n^{\mathbb{F}} = \sum_{k=1}^{K} [\beta_n]_k \Phi\left(v_k\right) \text{ with } \sum_{k=1}^{K} [\beta_n]_k = 1.$$

Together with the assumption in equation (3.4.1) that $D = D^{\mathbb{F}}$ and the distance definition given in equation (3.4.8) the cost function of Batch Neural Gas, cf. equation (4.1.2), can be reformulated to:

$$E_{\text{RNG}}\big(r_{k,n}, [\beta_n]_k\big) = \sum_{n=1}^{N}\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \left(\sum_{l=1}^{K} d_{k,l}[\beta_n]_l - \frac{1}{2} \cdot \sum_{l,l'=1}^{K} d_{l,l'}[\beta_n]_l[\beta_n]_{l'}\right) \tag{4.2.1}$$

with coefficients $[\beta_n]_k \in \mathbb{R}$. Here $r_{k,n}$ are the ranks from the prototypes $w_n^{\mathbb{F}}$ to the projected data point $\Phi\left(v_k\right)$. This cost function is optimized iteratively by a newton scheme of alternating steps: all ranks $r_{k,n}$ are calculated for fixed $[\beta_n]_k$, then all $[\beta_n]_k$

are adapted for fixed $r_{k,n}$. For optimizing the coefficient vectors $\vec{\beta}_n$ we determine the derivative of the cost function:

$$\frac{\partial E_{\mathrm{RNG}}\big(r_{k,n},[\beta_n]_k\big)}{\partial[\beta_i]_u} = \sum_{k=1}^{K} h_\sigma\left(r_{k,i}\right) d_{k,u} - \sum_{k=1}^{K} h_\sigma\left(r_{k,i}\right) \sum_{l=1}^{K} d_{l,u}[\beta_r]_l \tag{4.2.2}$$

$$= \sum_{k=1}^{K} d_{k,u}\left(h_\sigma\left(r_{k,i}\right) - \sum_{l=1}^{K} h_\sigma\left(r_{l,i}\right)[\beta_i]_k\right) \tag{4.2.3}$$

If D is nonsingular this derivative is zero for all i and u if and only if

$$[\beta_i]_u = \frac{h_\sigma\left(r_{u,i}\right)}{\sum_l h_\sigma\left(r_{l,i}\right)}. \tag{4.2.4}$$

We obtain algorithm 4.2.1 for Relational Neural Gas.

Algorithm 4.2.1 *Relational Neural Gas – RNG*

```
initialize $[\beta_n]_k$ with $\sum_{k=1}^{K}[\beta_n]_k = 1$
repeat
    determine the ranks $r_{k,n}$ according to equation (4.1.3) with the distance definition given in equation (3.4.8)
    determine the prototype position representations $[\beta_n]_k$ according to equation (4.2.4)
until convergence
```

Often relational data are not completely embeddable into an Euclidean space. Frequently in real life applications Relational Neural Gas is applied anyway. By calculating the pseudo Euclidean embedding (cf. section 3.4.2) it is possible to calculate the error term given in equation (3.4.3) representing a measure for the error made by the assumption of the existence of an Euclidean embedding and the application of Relational Neural Gas.

4.2.2 Supervised variant of VQ based relation learning

The supervised LVQ relational learning discussed here is Kernel Learning Vector Quantization which is very powerful in modeling as dissimilarities can be based on any Mercer kernel.

Kernel Learning Vector Quantization – KLVQ

The supervised variants of GLVQ (GRLVQ, GMLVQ) discussed so far used linear data transformations. If the class boundaries are complex and non-linear there is need of a large number of prototypes per class to suitably approximate these class boundaries. Local distances also correspond to non-linear decision boundaries. Localized GMLVQ, for instance, implements piecewise quadratic boundaries. Another possibility to cope with the non-linearity of the problem is to map the data non-linearly into a Hilbert space $\mathbb{F}$ by some mapping Φ such that the class boundaries become linear. This linearity is usually obtained if the mapping space $\mathbb{F}$ is very high-dimensional or has infinite dimensions. Using the theory of Mercer kernels (Schölkopf et al. 1999) the distance calculation in the mapping space $\mathbb{F}$ can be done by application of the Mercer kernel function $\mathfrak{k}_\Phi$ associated to the mapping Φ, cf. equation (3.4.6). (Qinand and Suganthan 2004) introduced an extension of GLVQ using this kernel trick: Kernel Learning Vector Quantization.

In this approach the prototype vectors $w_n^{\mathbb{F}}$ are expressed in terms of linear combinations of the projected data samples as given in equation (3.4.5) by $w_n^{\mathbb{F}} = \sum_{k=1}^{K}[\beta_n]_k\Phi\left(v_k\right)$. The given dissimilarities $D = (d_{k,l})$ for the data points v_k, v_l in the input set V are used according to equation (3.4.2) to calculate the corresponding Gram matrix G. This is a matrix of inner products and can be interpreted as kernel matrix $\mathfrak{K}$. The distance between a projected data point $\Phi\left(v_k\right)$ and a prototype vector $w_n^{\mathbb{F}}$ represented by the coefficient vector $\vec{\beta}_n$ is, as already introduced in section 3.4.3, given by equation (3.4.6) as

$$d_{k,n}^{\mathbb{F}} = \mathfrak{k}_\Phi\left(v_k, v_k\right) - 2\cdot\sum_{l=1}^{K}[\beta_n]_l\mathfrak{k}_\Phi\left(v_k, v_l\right) + \sum_{i,u=1}^{K}[\beta_n]_i[\beta_n]_u\mathfrak{k}_\Phi\left(v_i, v_u\right).$$

Using these assumptions, we can determine the winning prototypes $w_+^{\mathbb{F}}$ and $w_-^{\mathbb{F}}$ according to the definition given in equations (4.1.18) and (4.1.19) with the dissimilarity given in equation (3.4.6). The updating rules defined in equations (4.2.5) and (4.2.6) given a data point v_k can be generalized from the original data space $\mathbb{R}^M$ into the feature space $\mathbb{F}$:

$$\Delta w_+^{\mathbb{F}} = \epsilon\cdot\frac{\partial L\Big(\mu\left(\Phi\left(v_k\right), W^{\mathbb{F}}\right)\Big)}{\partial\mu\big(\Phi\left(v_k\right), W^{\mathbb{F}}\big)}\cdot\frac{4\cdot d_-^{\mathbb{F}}\left(\Phi\left(v_k\right)\right)}{\Big(d_+^{\mathbb{F}}\left(\Phi\left(v_k\right)\right) + d_-^{\mathbb{F}}\left(\Phi\left(v_k\right)\right)\Big)^2}\cdot\left(\Phi\left(v_k\right) - w_+^{\mathbb{F}}\right)$$

and

$$\Delta w_-^{\mathbb{F}} = \epsilon\cdot\frac{\partial L\Big(\mu\left(\Phi\left(v_k\right), W^{\mathbb{F}}\right)\Big)}{\partial\mu\big(\Phi\left(v_k\right), W^{\mathbb{F}}\big)}\cdot\frac{4\cdot d_+^{\mathbb{F}}\left(\Phi\left(v_k\right)\right)}{\Big(d_+^{\mathbb{F}}\left(\Phi\left(v_k\right)\right) + d_-^{\mathbb{F}}\left(\Phi\left(v_k\right)\right)\Big)^2}\cdot\left(\Phi\left(v_k\right) - w_-^{\mathbb{F}}\right)$$

with $d_{+}^{\mathbb{F}}\left(\Phi\left(v_k\right)\right)$ and $d_{-}^{\mathbb{F}}\left(\Phi\left(v_k\right)\right)$ defined as before in GLVQ but transferred to the feature space $\mathbb{F}$. Together with the abbreviation $c = \epsilon_w \cdot \frac{\partial L\left(\mu\left(\Phi(v_k),W^{\mathbb{F}}\right)\right)}{\partial \mu\left(\Phi(v_k),W^{\mathbb{F}}\right)}$ and equation (3.4.5) the update rules for adaptation step t are rewritten as

$$[\beta_+]_r\,(t+1) = \begin{cases} \left[1 - c \cdot \frac{4 \cdot d_{-}^{\mathbb{F}}\left(\Phi(v_k)\right)}{\left(d_{+}^{\mathbb{F}}\left(\Phi(v_k)\right) + d_{-}^{\mathbb{F}}\left(\Phi(v_k)\right)\right)^2}\right] \cdot [\beta_+]_r\,(t) & \text{if } v_r \neq v_k \\ \left[1 - c \cdot \frac{4 \cdot d_{-}^{\mathbb{F}}\left(\Phi(v_k)\right)}{\left(d_{+}^{\mathbb{F}}\left(\Phi(v_k)\right) + d_{-}^{\mathbb{F}}\left(\Phi(v_k)\right)\right)^2}\right] \cdot [\beta_+]_r\,(t) & \\ +c \cdot \frac{4 \cdot d_{-}^{\mathbb{F}}\left(\Phi(v_k)\right)}{\left(d_{+}^{\mathbb{F}}\left(\Phi(v_k)\right) + d_{-}^{\mathbb{F}}\left(\Phi(v_k)\right)\right)^2} & \text{if } v_r = v_k \end{cases} \tag{4.2.5}$$

and

$$[\beta_-]_r\,(t+1) = \begin{cases} \left[1 - c \cdot \frac{4 \cdot d_{+}^{\mathbb{F}}\left(\Phi(v_k)\right)}{\left(d_{+}^{\mathbb{F}}\left(\Phi(v_k)\right) + d_{-}^{\mathbb{F}}\left(\Phi(v_k)\right)\right)^2}\right] \cdot [\beta_-]_r\,(t) & \text{if } v_r \neq v_k \\ \left[1 - c \cdot \frac{4 \cdot d_{+}^{\mathbb{F}}\left(\Phi(v_k)\right)}{\left(d_{+}^{\mathbb{F}}\left(\Phi(v_k)\right) + d_{-}^{\mathbb{F}}\left(\Phi(v_k)\right)\right)^2}\right] \cdot [\beta_-]_r\,(t) & \\ +c \cdot \frac{4 \cdot d_{+}^{\mathbb{F}}\left(\Phi(v_k)\right)}{\left(d_{+}^{\mathbb{F}}\left(\Phi(v_k)\right) + d_{-}^{\mathbb{F}}\left(\Phi(v_k)\right)\right)^2} & \text{if } v_r = v_k. \end{cases} \tag{4.2.6}$$

The summary of all adaptation steps is shown in algorithm 4.2.2.

Algorithm 4.2.2 *Kernel Learning Vector Quantization – KLVQ*

```
initialize all $[\beta_n]_k$ with $\sum_{k=1}^{K}[\beta_n]_k = 1$ for $n = 1,\dots,N$ and $k = 1,\dots,K$
repeat
  randomly choose an input vector $v_k$ from $V$ with its label $z_{v_k}$
  determine winner prototypes $\vec{\beta}_+$ and $\vec{\beta}_-$ according to $v_k$, $z_{v_k}$ and $d_{k,n}^{\mathbb{F}}$
  determine new prototype representation for coefficients $[\beta_+]_r$ according to equation (4.2.5)
  determine new prototype representation for coefficients $[\beta_-]_r$ according to equation (4.2.6)
until convergence
```

In 2010 Schleif et al. introduced a derivative based, generalized version of KLVQ that operates on a "kernelized, differentiable metric called D-KGLVQ which allows the non-linear representation of the data"(Schleif et al. 2010, p. 22). It avoids the implicit mapping such that the prototypes still are defined in the original data space.

4.3 VQ based median learning

In this last section of the review of Vector Quantization based methods, four unsupervised learning algorithms are considered that take a matrix of pairwise dissimilarities as input, which does not need to be embeddable into the Euclidean space. We first present two crisp variants followed by their fuzzy extensions.

4.3.1 Crisp variants of VQ based median learning

In this section two algorithms for median clustering are given that use crisp mappings. The Median C-Means as well as the Affinity Propagation method are batch learning schemes.

Median C-Means – M-CM

Median c-means is a variant of classic k-means introduced by (Cottrell et al. 2006) in the generalization and extension of different VQ-based approaches towards median learning. The cost function for M-CM is given by

$$E_{\text{M-CM}} = \sum_{n=1}^{N} \sum_{k=1}^{K} \Xi_{\Psi(v_k)}(n) \cdot d(v_k, w_n) \tag{4.3.1}$$

with $\Xi_{\Psi(v_k)}(n)$ being the characteristic function of the winner index $\Psi(v_k) = s$ as given in equation (2.2.1), which refers to the index of the prototype w_s with minimum dissimilarity $d(v_k, w_s)$ to v_k. The number N of the prototypes has to be chosen beforehand. Only the dissimilarities between the data points $D = (d_{k,l}) = \big(d(v_k, v_l)\big)$ are given. Dissimilarities between data points and arbitrary prototypes cannot be calculated. The idea of M-CM is to restrict the prototypes to be chosen from the data points such that the dissimilarity between data points and prototypes can be obtained from the data dissimilarity matrix D.

As introduced in batch neural gas, cf. section 4.1.1, the cost function $E_{\text{M-CM}}$ is optimized by iteration through two alternating adaptation steps – one according to the prototype memberships and the other according to the prototype positions:

1. The first step in median c-means is the same assignment update as in classical k-means, see also equation (2.2.1)

   $$s = \Psi\left(v_k\right) = \arg\min_{n \in \mathbb{I}} \big(d\left(v_k, w_n\right) \big)$$

   where $\mathbb{I} = \{1, \ldots, N\}$ and the prototypes w_n are a fixed choice of data points, i.e. $w_n = v_l$. From these assignments the characteristic functions $\Xi_{\Psi(v_k)} = \Xi_s$ are defined.

2. In the second step the prototypes are determined according to

   $$w_n = v_l \quad \text{where} \quad l = \arg\min_{l'} \sum_{k=1}^{K} \Xi_n\left(l'\right) \cdot d\left(v_k, v_{l'}\right) \tag{4.3.2}$$

   assuming fixed $\Xi_n\left(l'\right)$ from step one. In this step it is necessary to avoid $w_i = w_u$ for $i \neq u$, the second best data point is chosen as prototype in this case.

These formulas lead to algorithm 4.3.1.

Algorithm 4.3.1 *Median C-Means – M-CM*

```
initialize prototype positions $w_n$ with data points $v_k$
repeat
    determine new cluster assignments (winners) $\Psi(v_k)$ for each input vector $v_k$
    according to the dissimilarity matrix $D$ and equation (2.2.1)
    determine new prototype positions according to equation (4.3.2)
until convergence
```

Affinity Propagation – AP

Another clustering approach where cluster centers are restricted to be data points and hands only data dissimilarities are required is Affinity Propagation as introduced by (Frey and Dueck 2007). It is based on message passing and closely related to spectral clustering, cf. (von Luxburg 2007) for an overview on spectral clustering. It is no median clustering in the above sense. The algorithm starts by assuming all input vectors as potential prototypes (exemplars) whose number is reduced in the course of calculation.

The idea of AP is to build a graph (or network) from the input vectors interpreted as nodes and exchanging real-valued messages along the edges until a stable set of prototypes and corresponding clusters emerges. Following (Frey and Dueck 2007) the dissimilarities between the K input vectors v_k – each one being a potential exemplar –

are interpreted as log-likelihoods of the probability that the vectors assume each other as prototypes, resulting in an exemplar-dependent probability model. Maximizing the cost function:

$$E_{\text{AP}}\left(\Psi\right) = \sum_{k=1}^{K} d\left(v_k, v_{\Psi(v_k)}\right) + \sum_{l=1}^{K} \delta_l\left(\Psi\right) \tag{4.3.3}$$

where $\Psi : \{1, \ldots, K\} \to \{1, \ldots, K\}$ is the mapping function defining the prototypes for each data point. $\delta_l\left(\Psi\right)$ is a penalty function

$$\delta_l\left(\Psi\right) = \begin{cases} -\infty & \text{if } \Psi\left(v_l\right) \neq l \text{ and there exists } k \text{ with }, \Psi\left(v_k\right) = l \\ 0 & \text{otherwise,} \end{cases}$$

penalizing invalid configurations where some data point k chooses l as an exemplar without l being labeled as an exemplar by $\Psi\left(v_l\right) = l$ (Frey and Dueck 2007).

It is also possible to formulate the cost function in terms of log-probabilities:

$$E_{\text{AP}}\left(\Psi\right) = \log\left(\Pi_{k=1}^{K} P\big(v_k, \Psi\left(v_k\right)\big) \cdot P(\Psi)\right) \tag{4.3.4}$$

where $P\big(v_k, \Psi\left(v_k\right)\big)$ is the probability that $\Psi\left(v_k\right)$ is the prototype for v_k and $P(\Psi)$ is the probability that this assignment is valid. It is stated by Frey and Dueck that normalization of this value has no effect on the solution.

The messages exchanged in the AP graph are of two (interdependent) kinds:

- The *responsibilities*

  $$r\left(l, k\right) = d\left(v_l, v_k\right) - \max_{l' \neq k} \left\{a\left(l, l'\right) + d\left(v_l, v_{l'}\right)\right\} \tag{4.3.5}$$

  measure how well the data point v_k can represent data point v_l, also accounting for other potential prototypes for v_l.

- The *availabilities*

  $$a(l, k) = \begin{cases} \min\left\{0, r\left(k, k\right) + \sum_{l \neq l', k} \max\left\{0, r\left(l', k\right)\right\}\right\} & \text{if } l \neq k \\ \max_{l' \neq k}\left\{\max\left\{0, r\left(l', k\right)\right\}\right\} & \text{if } l = k \end{cases} \tag{4.3.6}$$

  measure how well v_l is represented by v_k, also taking into account the measure of other data points to chose v_k as a prototype.

Using the responsibilities and the availabilities, the mapping function describing the prototypes is determined as:

$$\Psi\left(v_l\right) = \arg\max_{k} \left\{a\left(l, k\right) + r\left(l, k\right)\right\} \tag{4.3.7}$$

The values $a(l,k)$ and $r(l,k)$ can be interpreted as log-probability ratios. Interpreting the AP graph as factor graph (cf. e.g. (Pearl 1988) and (von Luxburg 2007)) and applying the max-sum-algorithm leads to an iterative alternating calculation of the $a(l,k)$ and $r(l,k)$ that results in the clustering of the nodes. Altogether algorithm 4.3.2 is obtained.

In AP the number of resulting prototypes is implicitly influenced by the self-dissimilarities $d(v_k, v_k)$ – also denoted as *preferences*. Data points with larger preferences are more likely to be chosen as exemplars. If all data points have equal preferences, the granularity of clustering is finer, the larger the self-dissimilarities are. They are commonly chosen equal to the median of input similarities or the minimum thereof.

Algorithm 4.3.2 *Affinity Propagation – AP*

```
initialize self-dissimilarities and $a(l,k) = 0, \forall\, l,k$
repeat
  determine the responsibilities according to equation (4.3.5)
  determine the availabilities according to equation (4.3.6)
  determine the prototype defining function according to equation (4.3.7)
until convergence
```

All these explanations are mainly taken from (Frey and Dueck 2007). For further reading we refer to this article.

4.3.2 Fuzzy variants of VQ based median learning

We introduce the extensions of the previously discussed median learning algorithms to fuzzy mappings. We published these batch schemes for Median Fuzzy C-Means in (Geweniger et al. 2010) and for Fuzzy Affinity Propagation in (Geweniger et al. 2009).

Median Fuzzy C-Means – M-FCM

We developed the median fuzzy c-means merging median c-means (M-CM) and fuzzy c-means (FCM) (Geweniger et al. 2010) . It uses fuzzy assignments of the objects to the cluster prototypes as in FCM but with the restriction of the prototypes being objects themselves as in M-CM. The resulting cost function is given by

$$E_{\text{M-FCM}} = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} \left(\psi_{w_n}(v_k)\right)^{\mathfrak{f}} \cdot d(v_k, w_n)^2 \tag{4.3.8}$$

with the fuzzifier $\mathfrak{f} > 1$ as before in FCM (cf. section 4.1.1). It is the same as for FCM apart from the fact that now the $d(v_k, w_n)$ may be arbitrary data dissimilarities as for M-CM.

Optimizing this cost function is again done by performing two alternating steps iteratively, first updating the assignments with fixed prototype positions and then taking the fixed assignments updating the prototype positions. The update rules are defined as follows

1. The assignment update has the same structure as the assignment update in FCM but with some differences in derivation of the cost function that can be found in detail in (Geweniger et al. 2010). It is defined by

$$\psi_{k,n} = \psi_{w_n}(v_k) = \frac{\sqrt[\mathfrak{f}-1]{d(v_k, w_n)^{-2}}}{\sum_{n'=1}^{N} \sqrt[\mathfrak{f}-1]{d(v_k, w_{n'})^{-2}}} \tag{4.3.9}$$

   with fixed prototype positions $w_n = v_l \in V$ at a choice of data point positions and dissimilarities $d(v_k, v_l)$ given in a dissimilarity matrix $D = \big(d(v_k, v_l)\big)$.

2. As in M-CM the prototype positions are restricted to be data point positions. The prototype position update of M-FCM is obtained in accordance to that of M-CM. It is given by

$$w_n = v_l \text{ with } l = \arg\min_{l'} \left[\sum_{k=1}^{K} \left(\psi_{w_n}(v_k)\right)^{\mathfrak{f}} d(v_k, v_{l'})^2 \right] \tag{4.3.10}$$

   for fixed assignments $\psi_{w_n}(v_k)$.

These update steps together with the initialization are summarized in algorithm 4.3.3.

Algorithm 4.3.3 *Median Fuzzy C-Means – M-FCM*

```
initialize membership degrees $\psi_{k,n}$ for all $n, k$ with $\sum_{n=1}^{N} \psi_{k,n} = 1$ and $\sum_{k=1}^{K} \psi_{k,n} > 0$
repeat
  determine the prototype positions $w_n$ according to equation (4.3.10)
  determine the assignment degrees $\psi_{k,n}$ according to equation (4.3.9)
until convergence
```

We showed the convergence of the algorithm in (Geweniger et al. 2010) and omit this proof here.

Fuzzy Affinity Propagation – FAP

In 2009 we introduced Fuzzy Affinity Propagation (Geweniger et al. 2009) as a heuristically motivated, direct fuzzy extension of the descriptions for original Affinity Propagation as given in section 4.3.1. The set of exemplars $W \subset V$ is defined as in equation 4.3.7. We define a cluster member probability $P\left(l,k\right)$ for each pair $v_l, v_k \in W$ to be $P\left(l,k\right) = 0$ if and only if $l \neq k$ and $P\left(l,l\right) = 1$. To gain a valid probability description of cluster assignments from the responsibilities as given in equation 4.3.5, we introduce the normalized responsibilities for non-exemplars

$$\widehat{r}\left(l,k\right) = C\frac{r\left(l,k\right) - \max_{l|v_l \notin W}\left\{r\left(l,k\right)\right\}}{\max_{l|v_l \notin W}\left\{r\left(l,k\right)\right\} - \min_{l|v_l \notin W}\left\{r\left(l,k\right)\right\}}. \qquad (4.3.11)$$

Choosing the normalization constant C appropriately according to the variance of $r\left(l,k\right)$ the probabilities for the mapping of data point v_l to the cluster prototype represented by v_k is defined by

$$P\left(l,k\right) = e^{\widehat{r}(l,k)}, \text{ with } P\left(l,k\right) \in [0,1].$$

These possibilistic cluster assignments can be interpreted as fuzzy degrees, subsequent normalization of the $P\left(l,k\right)$ yields a probabilistic variant. Summarizing we yield the update rules as given by algorithms 4.3.4.

Algorithm 4.3.4 *Fuzzy Affinity Propagation – FAP*

```
initialize self-dissimilarities and $a\left(l,k\right) = 0, \forall l,k$
repeat
  determine the responsibilities $r\left(l,k\right)$ according to equation (4.3.5) and the normalized responsibilities for the non-exemplars $\widehat{r}\left(l,k\right)$ according to equation (4.3.11)
  determine the availabilities $a\left(l,k\right)$ according to equation (4.3.6)
until convergence
```

4.4 Initialization, parameter setting and convergence of VQ algorithms in general

There is a number of settings necessary for applying vector quantization based methods concerning the initialization, the choice of parameters and the definition of convergence. In this section we introduce general considerations about these settings. The discussion of the specific settings in our application example is given in section 7.1.2.

4.4.1 Number of prototypes

With the exception of the Affinity Propagation based methods, the introduced VQ based algorithms require a predefined number of prototypes. In general, the problem how many prototypes are optimal is ill-posed. Additional constraints or prior knowledge are required. The initialization of the prototype positions is interdependent to the question of a suitable prototype number. We return to corresponding considerations in the discussion of initialization in section 4.4.2.

If there is prior knowledge, e.g. biological evidence, about the data structure in the given task this can be incorporated into the decision for the number of prototypes. If classes are known or expected to be multi modal, i.e. comprising subtypes, it is preferable to choose more than one prototype for these classes. The same holds for classes that are expected to comprise more variability within their data cloud(s), e.g. uneven formed data clouds in high dimensions.

For small data sets there is a trade off between the correct modeling of the multimodality/variability and the complexity of the model that can be approximated by the given samples. Careful evaluation of the results can narrow down the interval of a suitable prototype number. For evaluation several validation measures and visualizations were developed, see section 4.5 for the numerical evaluation and section 4.6 for visual inspection.

This evaluation is also needed in tasks without prior knowledge or where the prior knowledge is not reliable. A heuristic for the determination of a suitable number of prototypes is to vary their number and determine corresponding external or internal validation measures. The number of prototypes is chosen in accordance to the optimum measure values.

An example for a growing model is an extension to the online Neural Gas variant called *Growing Neural Gas (GNG)* by (Fritzke 1995). Its idea is to successively add prototypes to an initially small model. To determine the location of this addition a local statistical measure is gathered during the adaption steps. This can lead to too complex models because of the local (greedy) view of the decision. Therefore a reduction mechanism has to be installed in parallel. In GNG this is realized by removing prototypes, if they were not adapted over a specific period of time during training. Inspired by the GNG, (Qin and Suganthan 2004) introduced *Growing Generalized Learning Vector Quantization (G-GLVQ)* as a variant for the supervised algorithm GLVQ. These extensions are also applicable to the algorithms in our framework. With respect to the given complexity of the approaches this integration is postponed to further improvements of the system.

For the supervised algorithms it is possible to use an unsupervised (clustering) algorithm to determine a suitable number of clusters and use its results as initialization for the supervised learning. This is only successful if the cluster structure is congruent to the class structure.

4.4.2 Initialization of prototype positions

In the initialization of the prototypes the use of domain specific knowledge is possible by pertinent suggestions of the human experts. This can be done for unsupervised as well as for supervised methods. The experts can for example choose prototypical objects from the training data that are used as initial prototype positions for the approaches. As this is an annotation based approach it can produce the same problems as described for labeling in section 2.5.1.

If no reliable prior knowledge is available, for supervised as well as for unsupervised learning, frequently the prototypes are initialized around the center of mass of the training data. This has the advantage that there is no randomness of the result of the algorithm with respect to initializations. This initialization method can be misleading if the data for example are lying on a sphere.

Another common approach for initialization is to randomly initialize the positions of the prototypes in the data space. This approach is applicable for supervised and unsupervised data. If the method is sensitive to initializations, repeated trials have to be performed. The random choice of data points used as initial prototypes is a common alternative, e.g. for k-Means (Hastie, T. et al. 2003).

If label information is available it can be preferable to choose the prototype positions and labels according to the distribution of labels in the training set with at least one prototype per class. This can be interpreted as stratification.

For supervised learning it is possible to use the results of unsupervised method for initialization, i.e. unsupervised vector quantization with post labeling, cf. section 2.6. Difficulties arise with rare events and unbalanced data sets.

4.4.3 Initialization of parameters in dissimilarity adaptation

In approaches using dissimilarity adaptation the dissimilarity parameters have to be initialized for the first step of adaptation.

If there is prior domain and task specific knowledge it often is integrated to yield faster convergence and more stable results. For example, if some feature dimensions are known to be more important for the discrimination between different classes or clusters, the dissimilarity in these dimension should be weighted higher than the others in the initialization. The same holds for combinations of feature dimensions in the matrix-based integration. After the representation of the prior knowledge the factors are normalized according to the requirements of the algorithms.

A starting point for the initialization of the relevance factors without reliable prior knowledge is to give them all similar weights. Correspondingly, relevance matrices in GMLVQ are frequently initialized as the identity.

4.4.4 Learning rate for the prototype positions

In online learning processes the prototype positions are iteratively updated using a stochastic gradient on the cost function of the algorithm. The strength of this update is controlled by a factor – the learning rate. The pertinent choice of this learning rate is crucial for the success of learning. The learning rate lies in an interval of $(0, 1)$. If it is near zero, adaptation is done relatively careful and convergence in learning can take a long time. If the learning rate is close to one, adaptation is done in big steps which can prolong the adaptation time when jumping around the optimum.

From theory a decay of the learning rate during the overall learning process is needed for convergence. The respective theoretic base was provided by the theorem of Kushner-Clark (Kushner and Clark 1978) proving guaranteed convergence for the conditions given in section 4.1.2 in equations (4.1.15)–(4.1.17). These conditions enforce learning that needs infinite time, which is not applicable in praxis.

Frequently a small but fixed learning rate is chosen. In this case it has to be ensured that learning lasts sufficiently long. The learning rate is problem dependent and has to be chosen carefully to be small enough, i.e. $\epsilon_w \ll 1$. Experience influences the adequate choice.

(Papari et al. 2011) introduced a general method for way point averaging and step size control in gradient descent based learning approaches. After a predefined number of adaptation steps the cost function for normal adaptation is compared to the cost function of a sliding average over the most recent positions. If the latter has a smaller value a *jump* is performed. The averaged position is the new starting point for further adaptations and the step size is decreased by a predefined factor.

4.4.5 Learning rate for the dissimilarity adaptation

In online dissimilarity adaptation approaches like the GRLVQ we have a hierarchy of learning. On the first level the prototype positions and on the second level the dissimilarity is adapted. In theory, for suitable convergence of the whole model it is necessary that the dissimilarity is fixed during the adaptation process of the prototype positions. As stated in section 4.1.2 the dissimilarity has to change in an adiabatic manner. To approximately yield this stationarity of the dissimilarity, the dissimilarity has to be adapted significantly slower than the prototype positions. For the dissimilarity adaptation the frequent starting point is to use a learning rate of at least one magnitude smaller then the learning rate for the prototype positions.

In batch learning the dissimilarity is fixed during one epoch and the changes are applied afterwards. There is no strict condition for adiabatic dissimilarity adaptation in this case.

4.4.6 Convergence

The convergence of a learning algorithm is often defined by some stopping criterion. A frequent starting point is to use either the sum of the absolute amount of all prototype movements in one iteration or the single maximum movement found in the iteration. Both values are suited as a measure for the progress of learning. If this measure drops below a predefined threshold the model is expected to be fully adapted. This factor interacts with a controlled decrease of the learning rate. It leads to smaller movements in later adaptation steps. Scaling the relative movement by the learning rate prevents early termination of the learning process.

A computationally simple approach is to use a large number of sweeps through the whole training data set. One sweep is also called *epoch*. A too small number of sweeps will result in an underadapted model. In general a careful training is mandatory.

A more advanced criterion can be applied if the algorithm optimizes an energy or cost function. If the cost function does not change significantly any more, the algorithm may have reached a, possibly local, optimum. A simple test for convergence is

$$\frac{\big(E\left(t\right)-E\left(t+1\right)\big)}{E\left(t+1\right)}<\nu \tag{4.4.1}$$

where $E\left(t\right)$ is the corresponding value of the cost function in sweep t and ν is some predefined threshold. The choice of ν is highly application specific. The controlled decrease of the learning rate has to be considered.

It is possible to calculate more sophisticated stopping criteria for example by additionally estimating the slope of the cost function development. This way the flattening of the cost function development can be identified easily. This calculation can be time-consuming. Problems can occur where cost functions display *flat regions* which do not correspond to locally optimal solutions and may result in so-called quasi-stationary plateau states.

In supervised learning methods with a high model complexity the convergence problem is related to the precision-generalization-dilemma, i.e. more precision of $E\left(t\right)$ can lead to a loss of generalization ability. This problem is called *overtraining*. It is suitable to stop the training process before convergence if the generalization ability

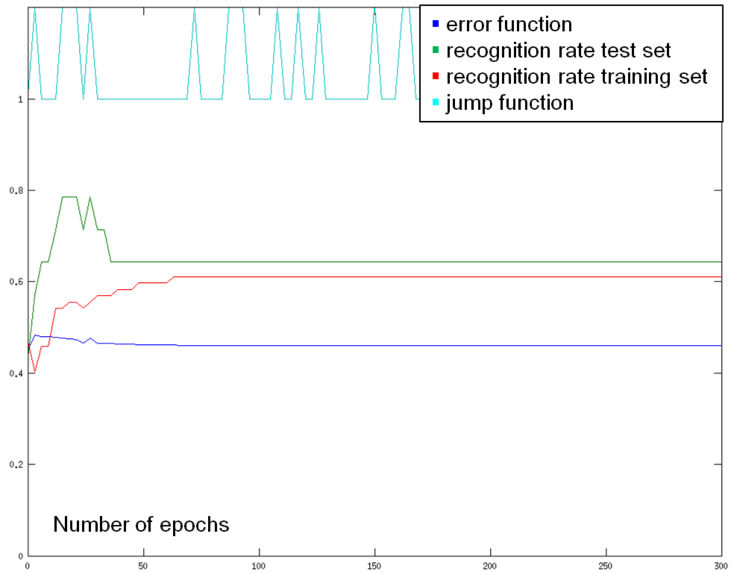


Figure 4.1: *Example plot of a process controlling the generalization ability during adaptation. In this example overtraining occurred. This is identified by an increasing training recognition rate (red) by simultaneously decreasing error function (blue) and dropping test recognition rate (green). In turquoise we show the jump function that is explained in detail in the text.*

significantly decreases. The generalization ability can be estimated using a test data set not used for training.

Figure 4.1 shows an example of overtraining in a process controlling the generalization ability during the adaptation in every third epoch. The error function is plotted over the number of epochs in the blue line. The recognition rates for the training and the test data set are given by the red line and the green line respectively. We coded the behavior in Papari's gradient descent as introduced in section 4.4.4 by the turquoise line. A value of 1.2 of the function corresponds to a jump event. In the case of a normal adaptation step the function value is 1. The error function decreases and the training recognition rate increases around epoch 40 but the test recognition rate drops. This indicates overtraining.

4.5 Numerical evaluation of vector quantization results

In the following section about numerical evaluation of vector quantization results we consider four constellations:

- crisp clustering
- crisp classification
- fuzzy clustering
- fuzzy classification

Often the evaluation measures for methods with fuzzy mappings are derived from evaluation measures for crisp methods. If the prototypes in VQ are interpreted as cluster centers regardless of the procedure from which they are obtained, all methods of cluster evaluation can be applied.

4.5.1 Evaluation of VQ based learning for crisp clustering

Our discussion of evaluation starts with the evaluation of crisp clusterings. Although clusters can be represented by more than one prototype, for simplicity we assume in the following that there is a unique correspondence between prototypes and clusters. In general the clustering problem is ill-posed. That means that in clustering the Hadamard's properties of well-posed problems[5] are not fulfilled.

Due to the task being ill-post the evaluation of clustering is manifold and many different evaluation measures and methods exist, highlighting different aspects of the clustering solution properties. To get an overall judgment of the quality, it is useful to find a joint evaluation of several evaluation possibilities. Especially for measures and indices a simple combination of the measures would frequently result in information loss. Forming tuples from these measure values that are then evaluated with respect to Pareto optimality is a comprehensive way of studying an evaluation task, suggested e.g. by (Handl et al. 2005). As it is computationally expensive it is seldom used.

For the numerical evaluation measures and approaches we focus mainly on three references: (Handl et al. 2005) on evaluation of crisp clustering of biological data, (Halkidi et al. 2001) on clustering validation techniques in general and the different sections on evaluation from (Manning et al. 2008). The variety of evaluation possibilities can be categorized in several ways (for examples we refer to (Duda et al. 2001), (Manning et al. 2008) and (Halkidi et al. 2001)). We use the categorization given by (Handl et al. 2005):

Internal validation measures These methods "attempt to measure how well a given partitioning corresponds to the natural cluster structure of the data" (Handl

[5] existence of a solution, uniqueness of the solution, continuous dependency of the solution on given data in some reasonable topology (Hadamard 1902)

et al. 2005, p. 3203). Usually the quality is expressed in terms of properties like compactness, separability or density information.

External validation measures These measures are applicable to evaluate the cluster model if some kind of ground truth is given for the training and test data. They can also be used for comparing different clustering algorithms, e.g. "on benchmark data, for which the class labels are known to correspond to the true cluster structure." (Handl et al. 2005, p. 3203)

External measures frequently ignore any data dissimilarity information. They can also be used in the evaluation of supervised methods. We will introduce them in section 4.5.2 dealing with the evaluation of crisp classification.

Internal measures

Internal measures evaluate cluster quality properties like separation and compactness. These properties are determined by the relation between the data points and the prototypes in the data space that are given by the dissimilarity measure. Originally most internal measures were defined for the Euclidean distance. Many of them can be applied if other dissimilarities are used instead, paying attention to the possibly changed properties of the measures. It is conceptually pertinent to use the same dissimilarity measure in learning and evaluation.

In this section we restrict the detailed introduction to measures that we will use in our application example (see chapters 6 and 7). (Handl et al. 2005) distinguish three main internal clustering properties:

- compactness
- connectedness
- separation

Compactness and separation exhibit opposing trends. To get a more comprehensive quality assessment for the clustering often *combination measures*, combining the evaluation of these quality trends and allowing their trade off are used. Very popular combination measures are the Dunn index and Dunn-like indices as well as the Davies-Bouldin index. These measures integrate several cluster properties and calculate one global value for the evaluation of a cluster solution. This is especially suited for automatic evaluation e.g. in automated parameter tests. It does not allow a local evaluation of the cluster quality for single data points.

In our application example, see section 6.5.2, we used the silhouette width. To introduce this measure, we recall the following definitions. We assume a learned crisp clustering for a data point set V, represented by the assignment function

$$\Psi : V \to \mathbb{I} : v_k \mapsto s = \arg\min_{n \in \mathbb{I}} \left(d\left(v_k, w_n\right) \right)$$

according to equation (2.2.1) with the index set $\mathbb{I} = \{1, \ldots, N\}$ of the prototypes and their corresponding receptive fields

$$\Upsilon_n = \left\{ v_k \in V : \Psi\left(v_k\right) = n \right\}$$

as defined in equation (2.2.4) in section 2.2.

The *silhouette width* $S\left(v_k\right)$ for every data point v_k is a measure of how well the data point matches the clustering. It is formalized as

$$\mathrm{S}\left(v_k\right) = \frac{b\left(v_k\right) - a\left(v_k\right)}{\max\left\{a\left(v_k\right), b\left(v_k\right)\right\}} \tag{4.5.1}$$

where $a\left(v_k\right)$ is the average dissimilarity of v_k to all other objects in the same cluster Υ_s with $s = \Psi\left(v_k\right)$

$$a\left(v_k\right) = \frac{1}{\left|\Upsilon_s\right| - 1} \sum_{v_l \in \Upsilon_s, v_l \neq v_k} d\left(v_l, v_k\right)$$

and $b\left(v_k\right)$ is the average dissimilarity of v_k to the *neighboring cluster* given by

$$b\left(v_k\right) = \min_{\Upsilon_u \neq \Upsilon_s} d\left(v_k, \Upsilon_u\right)$$

where $d\left(v_k, \Upsilon_u\right)$ is defined as

$$d\left(v_k, \Upsilon_u\right) = \frac{1}{\left|\Upsilon_u\right|} \sum_{v_l \in \Upsilon_u} d\left(v_l, v_k\right).$$

The values for the silhouette width range from minus one to one, i,e, $-1 \leq S\left(v_k\right) \leq 1$. The $S\left(v_k\right)$ are calculated for all given data points $v_k \in V$. The presence of many negative silhouette values for one data set V and a given clustering indicates weak separation of the clusters. The averaged silhouette index (over all $S\left(v_k\right)$ for $v_k \in V$) can be used for an overall cluster quality assessment. A higher value refers to better cluster separation and compactness.

Additionally, *silhouette plots* allow a cognitive ergonomic presentation of all silhouette values at a glance. To generate these plots all silhouette values are sorted in decreasing order for each cluster separately. Taking this order as curve and concatenating all of the several clusters we obtain an overall curve which allows an interpretation according to the separation in the clustering. According to (Rousseeuw 1987) this interpretation is oriented mainly at the shape of the curve (heavy slopes, ranges of negative values). There are no absolute criteria for the interpretation of the silhouette plots in terms of quality. A suitable approach to yield an evaluation framework in a given application is to compare the silhouette plots of several different clustering results for the application. Figure 4.2 shows the comparison of two silhouette plots using different cluster numbers for the same application.

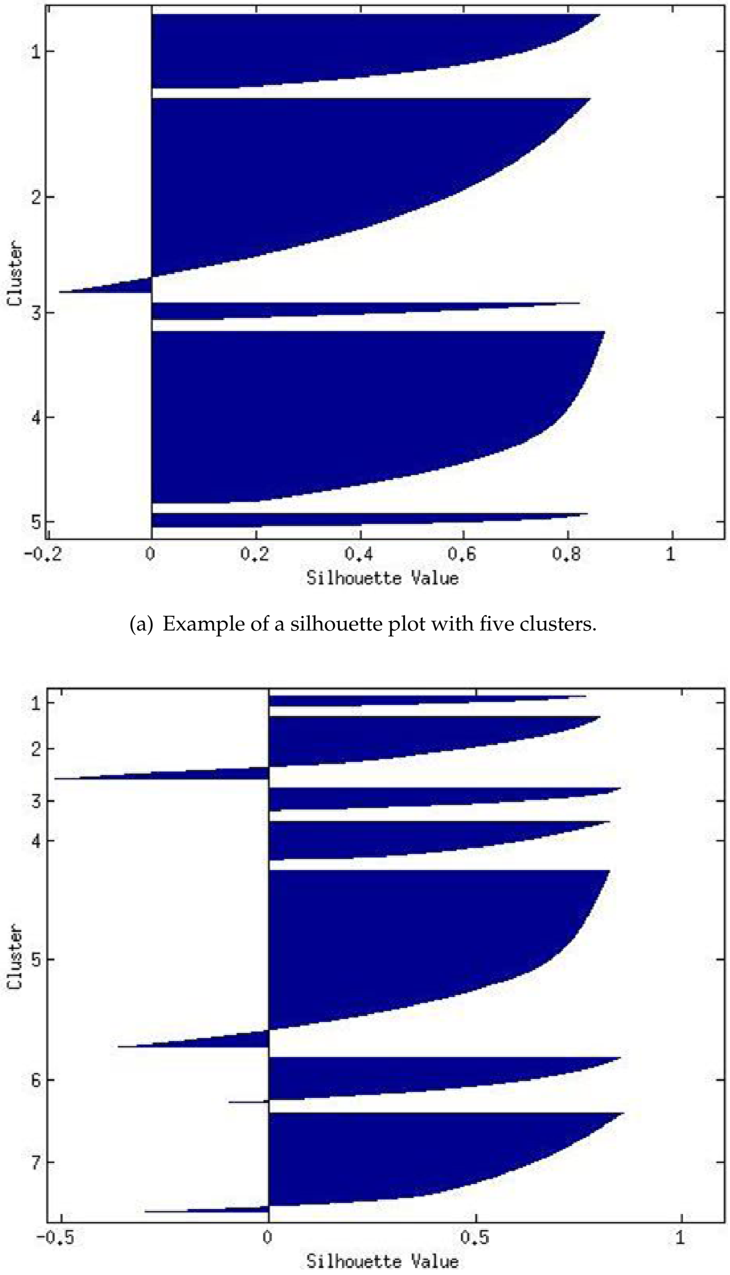
Like the Dunn and Dunn-like indices and the Davies-Bouldin index, the silhouette width is adequate when evaluating clusterings using large data sets. All these measures represent the dissimilarity of clusters by data point dissimilarities between these clusters. These dissimilarities are incalculable in the case of empty clusters. Empty clusters appear for example when the test data set is small. A work around that enables the use of these measures also for small data sets is to ignore empty clusters in the evaluation.

To assess the *predictive power or the stability* of a clustering algorithm, it is rerun several times with data re-sampled or perturbed from the original dataset. From the consistency of the results for the different runs, e.g. in terms of cluster centers, a non-deterministic statistical estimate of their significance can be determined. There exist several different approaches for re-sampling or perturbing (cf. (Handl et al. 2005) for details).

Hit statistics for crisp clustering

The *hit statistics* are instruments for evaluating the learning process. They are frequently applied in crisp clustering approaches where during one iteration only selected prototypes are adapted. This is often the case in online learning approaches. The hit statistic counts for every prototype how often it was selected for adaptation. This allows the identification of prototypes, that were seldom or never adapted during training. Possible reasons for this underadaption are the presence of a small subgroup in the data or that too many prototypes are used.

The first reason can only be approved by domain experts looking at the receptive fields of the prototypes and validating the conceptual appropriateness of the modeled groups. To identify the second constellation the dissimilarities between the prototypes can be additionally evaluated. Reducing the number of prototypes or choosing a different initialization are possible approaches to overcome the problem of underadapted prototypes.



(a) Example of a silhouette plot with five clusters.

(b) Example of a silhouette plot with seven clusters.

Figure 4.2: *Example of two silhouette plots in one application for evaluating and comparing the cluster quality for different cluster numbers. In figure 4.2(b) there are comparatively many negative values and smaller positive values. The clustering quality is higher in figure 4.2(a).*

4.5.2 Evaluation of LVQ based learning for crisp classification

In the following section we will focus on the evaluation of crisp classifications. The prototypes with their corresponding receptive fields can be handled as cluster centers and respective evaluation methods can also be applied to these learning results.

External measures are useful tools in the evaluation of prototype based classification. A suitable extension of the hit statistics gives further evaluation possibilities for supervised VQ based learning.

We discuss further measures specialized for evaluating classification performance independent from the underlying prototype structure that focus on correct class label mappings:

- Cohen's kappa for comparison of two classifiers and
- Fleiss' kappa for more than two classifiers.

Furthermore the error rate measure is used in evaluation strategies like

- the error rate evaluation and $\mathbb{k}$-fold cross-validation as well as in
- the leave-one-out validation.

External measures

We base our explanation of external measures on the definitions given in (Handl et al. 2005) and (Manning et al. 2008). Only measures that we use in our application example are introduced in detail.

Purity and *completeness* are two of the simplest external measures for evaluating the quality of groupings using label information. Purity "denotes the fraction of the cluster taken up by its predominant class label, whereas completeness denotes the fraction of items in this predominant class that is grouped in the cluster at hand." (Handl et al. 2005, p. 3203) Both measures have trivial optima: singleton groups for purity and one group for completeness. They are frequently used together.

To introduce two important measures of classification quality we give more definitions. Consider a trained crisp mapping $\Psi : V \to \mathbb{I}$ from the set of labeled data points $v_k \in V$ to the index set $\mathbb{I}$ for the set of trained prototypes W. The classification of a data point v_k is given by $\zeta_{v_k}(Z) = z_{w_s}$, see equation (2.3.1). By definition there are the following sets for a considered class label z_c:

$$\begin{aligned} \mathrm{TP}_c &= \{v_k : \zeta_{v_k} = z_c \text{ and } z_{v_k} = z_c\} && \text{the set of } \textit{true positives} \\ \mathrm{TN}_c &= \{v_k : \zeta_{v_k} \neq z_c \text{ and } z_{v_k} \neq z_c\} && \text{the set of } \textit{true negatives} \\ \mathrm{FP}_c &= \{v_k : \zeta_{v_k} = z_c \text{ and } z_{v_k} \neq z_c\} && \text{the set of } \textit{false positives} \\ \mathrm{FN}_c &= \{v_k : \zeta_{v_k} \neq z_c \text{ and } z_{v_k} = z_c\} && \text{the set of } \textit{false negatives} \end{aligned} \tag{4.5.2}$$

The cardinalities of these sets $|\cdot|$ are also called *contingency values*.

There are two important basic measures that are defined by these contingency values. They are used to measure the classification quality separately for the single classes $z_c \in Z$:

The precision of class z_c The precision is the fraction of all data points classified as belonging to class z_c that actually are labeled as belonging to this class. This is formalized as:

$$\mathrm{P}_c = \frac{|\mathrm{TP}_c|}{|\mathrm{TP}_c| + |\mathrm{FP}_c|} \tag{4.5.3}$$

The recall of class z_c The recall is the fraction of all data points actually labeled as belonging to class z_n that were classified as belonging to this class. The formalization yields:

$$\mathrm{R}_c = \frac{|\mathrm{TP}_c|}{|\mathrm{TP}_c| + |\mathrm{FN}_c|} \tag{4.5.4}$$

These two measures express opposing trends. While the recall is a non-decreasing function of the data points classified as belonging to class z_c[6], the precision usually decreases as the number of data points classified as belonging to class z_c increases[7].

There exists a variety of other external quality measures. We refer to (Halkidi et al. 2001) who gave a broad overview on these techniques.

Hit statistics for crisp classification

In every approach that discriminates different kinds of adaptations we use an extension of the hit statistic that was introduced in section 4.5.1. For every prototype it counts the number of selections for the different adaptations separately. In GLVQ based algorithms two kinds of adaptations take place: an attracting and a repelling adaption. If a prototype was relatively often selected for the repelling adaptation it was probably pushed out of the data space. Its position is no reliable model of the data structure. Hit statistics are of limited expressiveness and have to be combined e.g. to suitable visualization approaches of the data space[8] for a differentiated quality assessment of prototype based classification.

Cohen's kappa κ_C

Cohen's κ_C (see (Cohen 1960) and (Cohen 1972)) gives the inter-classifier agreement of two classifiers by relating the relative agreement p_0 among the classifiers $\mathfrak{C}_1$ and

[6] as it ignores the false negatives

[7] as it incorporated the number of false negatives

[8] We refer to section 4.6 for corresponding visualization approaches in our application example.

$\mathfrak{C}_2$ to the agreement occurring by chance p_c. This accidental agreement is expressed by the expected value of the joined event of $\mathfrak{C}_1$ and $\mathfrak{C}_2$ classifying certain data points to the same class.

Often the compared classifiers were trained on data sets with a large intersection or even the same data set. In the strict sense, the probability of classifier $\mathfrak{C}_1$ classifying a data point to a certain class is not independent from the probability of classifier $\mathfrak{C}_2$ classifying the same data point to the same class. To model and incorporate the dependence between these two events into a joined probability is virtually impossible. Simplifying, it is assumed that both classifiers are independent. Under this assumption the calculation of the joined probability reduces to the product of the probabilities for the single classification events.

We express the classification result of classifier $\mathfrak{C}$ for a data point v_k in a vector $\tilde{\zeta}^{\mathfrak{C}}(v_k) = \left(\tilde{\zeta}_1^{\mathfrak{C}}(v_k), \ldots, \tilde{\zeta}_C^{\mathfrak{C}}(v_k)\right)$. Its values sum to one, $\sum_{c=1}^{C} \tilde{\zeta}_c(v_k)^{\mathfrak{C}} = 1$, and are either zero or one, $\tilde{\zeta}_c^{\mathfrak{C}}(v_k) \in \{0, 1\}$. A value of the vector is one, $\tilde{\zeta}_c^{\mathfrak{C}}(v_k) = 1$, if and only if the classifier $\mathfrak{C}$ assigns to the data point v_k the class c, i.e. $\zeta_{v_k} = z_c$.

Using this expression and the independence assumption, p_c can be formalized as follows (Zühlke et al. 2009)

$$p_c = \sum_{c=1}^{C} \sum_{\tilde{\zeta}_c^{\mathfrak{C}_1}=0}^{1} \sum_{\tilde{\zeta}_c^{\mathfrak{C}_2}=0}^{1} \pi_c^{\mathfrak{C}_1} \cdot \pi_c^{\mathfrak{C}_2} \left(\tilde{\zeta}_c^{\mathfrak{C}_1} \cdot \tilde{\zeta}_c^{\mathfrak{C}_2}\right).$$

Here $\pi_c^{\mathfrak{C}_1}$ and $\pi_c^{\mathfrak{C}_2}$ are the margin probabilities $\pi_c^{\mathfrak{C}_q} = \frac{1}{K}\sum_{k=1}^{K} \tilde{\zeta}_c^{\mathfrak{C}_q}(v_k)$, $q = 1, 2$.

The relative agreement p_0 of the two classifiers $\mathfrak{C}_1$ and $\mathfrak{C}_2$ is given according to the contingency table of the classifications of both classifiers

$$p_0 = \frac{1}{K} \sum_{k=1}^{K} \sum_{c=1}^{C} \tilde{\zeta}_c^{\mathfrak{C}_1}(v_k) \cdot \tilde{\zeta}_c^{\mathfrak{C}_2}(v_k)$$

Using these values of p_c and p_0 the Cohen's kappa is given by

$$\kappa_C = \frac{p_o - p_c}{1 - p_c}. \tag{4.5.5}$$

The κ_C value range is $[-1, 1]$. In table 4.2 we show a commonly accepted categorization of the values where e.g. values less than zero mean a poor agreement whereas values between 0.8 and 1 describe perfect agreements (Sachs 2006).

Fleiss' kappa κ_F

(Fleiss et al. 2003) extended Cohen's kappa directly for more than two classifiers. The respective expected value of the agreement by chance p_c^F is calculated under the

κ	value meaning
$\kappa < 0$	poor agreement
$0 \leq \kappa \leq 0.2$	slight agreement
$0.2 < \kappa \leq 0.4$	fair agreement
$0.4 < \kappa \leq 0.6$	moderate agreement
$0.6 < \kappa \leq 0.8$	substantial agreement
$0.8 < \kappa \leq 1$	perfect agreement

Table 4.2: *Interpretation of kappa values according to (Sachs 2006).*

assumption of the independence between the classification events. The formalization of the probability for the joined event is the product of the single classification event's probabilities. This yields

$$p_c^F = \sum_{c=1}^{C} \sum_{\tilde{\zeta}_c^{\mathfrak{C}_1}=0}^{1} \cdots \sum_{\tilde{\zeta}_c^{\mathfrak{C}_Q}=0}^{1} \prod_{q=1}^{Q} \pi_c^{\mathfrak{C}_q} \tilde{\zeta}_c^{\mathfrak{C}_q} \tag{4.5.6}$$

where Q is the number of classifiers to be compared. The margin probabilities $\pi_c^{\mathfrak{C}_q}$ and the classification result values $\tilde{\zeta}_c^{\mathfrak{C}_q}$ are analogously defined as before for Cohen's kappa.

The relative agreement of the set of classifiers p_0^F is obtained from the contingency table and given as:

$$p_0^F = \frac{1}{K} \sum_{k=1}^{K} \sum_{c=1}^{C} \prod_{q=1}^{Q} \tilde{\zeta}_c^{\mathfrak{C}_q} (v_k)$$

The Fleiss' κ_F is calculated from the expected agreement by chance and the relative agreement with the structurally equivalent equation as for Cohen's kappa

$$\kappa_F = \frac{p_o^F - p_c^F}{1 - p_c^F}.$$

For the possible values of Fleiss' kappa $\kappa_F \in [-1, 1]$ holds and the value categorization as found in table 4.2 can be applied.

Error rate (ER) evaluation and $\Bbbk$-fold cross-validation ($\Bbbk$CV)

The following descriptions on error rate evaluation of learning for labeled data are based on Witten and Frank's "Data Mining: Practical Machine Learning Tools and Techniques" (Witten and Frank 2005). For classification models it is pertinent to measure the credibility of the learned model in terms of the error rate.

Despite simply counting errors, the error rate ER can be determined from the contingency values as

$$\mathrm{ER} = \frac{|\mathrm{FP}| + |\mathrm{FN}|}{|\mathrm{TP}| + |\mathrm{FP}| + |\mathrm{FN}| + |\mathrm{TN}|}.$$

Equivalently, the recognition rate or accuracy of a classifier, with $\mathrm{RR} = 1 - \mathrm{ER}$ can be used

$$\mathrm{RR} = \frac{|\mathrm{TP}| + |\mathrm{TN}|}{|\mathrm{TP}| + |\mathrm{FP}| + |\mathrm{FN}| + |\mathrm{TN}|}. \tag{4.5.7}$$

The overall contingency values TP, FP, FN and TN are given as sum of the contingency values for the single classes, introduced in section 4.5.2 in the equations (4.5.2), for example

$$\mathrm{TP} = \sum_{c=1}^{C} \mathrm{TP}_c.$$

Testing the classifier on the training data in general leads to overly optimistic recognition rates as the classifier was specialized for this data and may not generalize to new data. Therefore the test recognition rate is a more adequate estimate. The test recognition rate is unrepresentative if only a unrepresentative amount of data points is available in the test data set.

One method to cope with a limited amount of data is $\Bbbk$-fold cross-validation. A rule of thumb for small data sets is to split the data set randomly into at least three folds (thus $\Bbbk = 3$) and use two of them for training and the third for testing. This can lead to unrepresentative data sets as by the random splitting it is possible that seldom classes are not represented in one of the sets. To overcome this problem, the data sets are randomly sampled in a way assuring that every class is represented in about the right proportion in training and test set. This technique is called stratification and leads to the stratified cross-validation.

This stratified cross-validation still has a bias caused by the choice of the data points for the folds. A way to reduce this phenomenon is to run the process several times for different random fold splittings. Averaging the recognition rates on the different sets leads to an estimate of an overall recognition rate. To estimate the stability of the recognition rate usually its standard deviation is evaluated additionally.

Leave-one-out validation (LOOV) and interpretation

A special case of the $\Bbbk$-fold cross-validation is the leave-one-out validation where $\Bbbk$ is chosen as the number of data points in V. This method yields the following advantages:

- The greatest possible amount of data is used for training.

- Sampling is deterministic.
- Outliers are easily detected.

The method has also some draw backs:

- The computational costs are high.
- The test sampling (only one data point) is unstratified.

The initialization of a LOOV run for a single data point is nondeterministic. To avoid accidental unrepresentative results in the LOOV, we suggest to run it several times for every data point with different initializations, cf. algorithm 4.5.1. We assemble the results for such repeated runs in terms of single data point's recognition rather than in data set error rate estimation. These data point recognition rate results give an estimate of how well a single data point is represented by the remaining set of data points.

Algorithm 4.5.1 *Leave-one-out validation for learning algorithms*

```
Require: data set $V$, LearningAlgorithm
  for all data points $v_k \in V$ do
    Initialize countRight=0, countFalse=0;
    for $i = 1$ to e.g. $100$ do
      Train classifier $\mathcal{C}$ with all data points except $v_k$ and the given LearningAlgorithm
      Classify $v_k$ using classifier $\mathcal{C}$
      if $v_k$ classified right then
        countRight++
      else
        countFalse++
      end if
    end for
    print countRight, countFalse for $v_k$
  end for
```

4.5.3 Evaluation of VQ based learning for fuzzy clustering

The evaluation of fuzzy mappings of unlabeled data is often not as straight forward as for crisp mappings. For sake of completeness, we will mention some fuzzy clustering evaluation possibilities. We will not go into detail as we do not consider fuzzy clustering in our application example[9].

(Kim et al. 2003) give a review of several frequently used indexes for fuzzy cluster evaluation in the context of Fuzzy C-Means (FCM) algorithms. (Geweniger et al. 2011) use a selection of these measures for non-Euclidean dissimilarities in FCM. They considered

- the partition entropy by (Bezdek 1974)
- the partition coefficient by (Bezdek 1974)
- a validity index by (Xie and Beni 1991)
- a validity index by (Fukuyama and Sugeno 1989)

The partition entropy is a measure that solely reflects the compactness of the clustering and should be evaluated together with its opposite measure, the partition coefficient. The validity indices of Xie-Benii and Fukuyama-Seguno are both combined measures reflecting separation in addition to compactness of the clustering.

4.5.4 Evaluation of LVQ based learning for fuzzy classification

In this section we focus on evaluation possibilities for fuzzy classification. The methods that we introduce are direct extensions of the corresponding evaluation approaches for crisp classification:

- Fuzzy Cohen's kappa
- Fuzzy Fleiss' kappa
- Fuzzy recognition rate

As defined in equation (2.3.2) in section 2.3.1, the output of a fuzzy classifier is a vector of continuous assignment values $\vec{\zeta}_{v_k}(Z) = \left(\zeta_{v_k}(1), \dots, \zeta_{v_k}(C)\right)$. The assignment values lie between zero and one, $\zeta_{v_k}(c) \in [0,1]$, and they sum up to one for probabilistic classification, i.e. $\sum_{c=1}^{C} \zeta_{v_k}(c) = 1$.

[9] For the discussion of the algorithms that are suitable for our application example we refer to section 7.1.1.

Fuzzy Cohen's kappa $\kappa_C^{\mathcal{F}}$

To extend Cohen's kappa to fuzzy mappings, (Dou et al. 2007) introduce the fuzzy agreement between the fuzzy class assignments $\vec{\zeta}_{v_k}^{\mathfrak{C}_1}$ and $\vec{\zeta}_{v_k}^{\mathfrak{C}_2}$ of two classifiers $\mathfrak{C}_1$ and $\mathfrak{C}_2$ for the data point v_k. It is given by the fuzzy agreement function

$$f^{\mathcal{F}}(v_k) = \sum_{c=1}^{C} \left(\zeta_{v_k}^{\mathfrak{C}_1}(c) \wedge \zeta_{v_k}^{\mathfrak{C}_2}(c) \right)$$

with the following properties:

- The function values lie in the interval zero to one, $f^{\mathcal{F}}(v_k) \in [0, 1]$.
- The function value for a data point v_k is one, $f^{\mathcal{F}}(v_k) = 1$, if and only if the classifiers assigns for all classes $z_c \in Z$ the same assignment values to the data point v_k , i.e. $\zeta_{v_k}^{\mathfrak{C}_1}(c) = \zeta_{v_k}^{\mathfrak{C}_2}(c)$.

In accordance to this agreement function the proportion of observed agreement between two fuzzy classifiers on a data set V is

$$p_0^{\mathcal{F}} = \frac{1}{K} \sum_{k=1}^{K} \sum_{c=1}^{C} \left(\zeta_{v_k}^{\mathfrak{C}_1}(c) \wedge \zeta_{v_k}^{\mathfrak{C}_2}(c) \right).$$

To calculate the expected agreement by chance we assume the independence between the classification events and get the joined probability as product of the single classification event probabilities. This is formalized as

$$p_c^{\mathcal{F}} = \sum_{c=1}^{C} \int_{\zeta_c^{\mathfrak{C}_1}=0}^{1} \int_{\zeta_c^{\mathfrak{C}_2}=0}^{1} \pi\left(\zeta_c^{\mathfrak{C}_1}\right) \cdot \pi\left(\zeta_c^{\mathfrak{C}_2}\right) \left(\zeta_c^{\mathfrak{C}_1} \wedge \zeta_c^{\mathfrak{C}_2}\right) \mathrm{d}\zeta_c^{\mathfrak{C}_1} \mathrm{d}\zeta_c^{\mathfrak{C}_2}$$

where $\pi\left(\zeta_c^{\mathfrak{C}_1}\right)$ and $\pi\left(\zeta_c^{\mathfrak{C}_2}\right)$ are the margin probabilities of the classification events $\zeta_c^{\mathfrak{C}_1}$ and $\zeta_c^{\mathfrak{C}_2}$ and the abbreviation $\zeta_{v_k}^{\mathfrak{C}_q}(c) = \zeta_c^{\mathfrak{C}_q}$ is used (Geweniger et al. 2010). The equation for the resulting fuzzy Cohen's kappa structurally stays the same:

$$\kappa_C^{\mathcal{F}} = \frac{p_0^{\mathcal{F}} - p_c^{\mathcal{F}}}{1 - p_c^{\mathcal{F}}} \tag{4.5.8}$$

The important question for the evaluation of the given equations is the definition of the fuzzy AND-Operation $\wedge$. There is no unique determination of the AND-operation for fuzzy values. The theoretic basis is the definition by *t-norms* (Geweniger et al. 2010), that was detailed by (Hammer and Villmann 2007). According to their definition a function $\mathfrak{T} : [0, 1]^2 \to [0, 1]$ is called a t-norm if the following conditions are fulfilled

1. $\mathfrak{T}(a, 1) = a$ (neutral element)
2. $a \leq b \Rightarrow \mathfrak{T}(a, c) \leq \mathfrak{T}(b, c)$ (monotonicity)
3. $\mathfrak{T}(a, b) = \mathfrak{T}(b, a)$ (commutativity)
4. $\mathfrak{T}\big(a, \top(b, c)\big) = \mathfrak{T}\big(\top(a, b), c\big)$ (associativity).

These conditions do not determine a unique norm. Possible implementations are (Zühlke et al. 2009)

the min norm $\mathfrak{T}_{\text{min}}(a, b) = \min\{a, b\}$

the product norm $\mathfrak{T}_{\text{prod}}(a, b) = a \cdot b$

the Lukasiewicz norm $\mathfrak{T}_{\text{Luka}}(a, b) = \max\{0, a + b - 1\}$

In (Zühlke et al. 2009) we tested the behavior of the κ values using different norms on different data sets and also compared these values to the crisp κ values. From the tests we conclude "that the minimum norm $\mathfrak{T}_{\text{min}}$ seems to be most appropriate" (Zühlke et al. 2009, p. 274) for the fuzzy κ calculation.

Fuzzy Fleiss' kappa $\kappa_F^{\mathcal{F}}$

Structurally analog to the fuzzy Cohen's kappa derivation we introduced a fuzzy variant of Fleiss' kappa in (Zühlke et al. 2009). To use this structural analogy we rewrite the expected value of agreement by chance for the crisp Fleiss' kappa, see equation (4.5.6), as

$$p_c^F = \sum_{c=1}^{C} \sum_{\tilde{\zeta}_c^{\mathfrak{C}_1}=0}^{1} \cdots \sum_{\tilde{\zeta}_c^{\mathfrak{C}_Q}=0}^{1} \left(\prod_{q=1}^{Q} \pi^{\mathfrak{C}_q}(c) \right) \cdot \left(\prod_{r=1}^{Q} \tilde{\zeta}_c^{\mathfrak{C}_r} \right)$$

with separated products for the margin probabilities of the classification events $\pi^{\mathfrak{C}_q}(c)$ and the classification result vectors $\tilde{\zeta}_c^{\mathfrak{C}_r}$.

Analog to the Cohen's kappa fuzzy extension, the product of the classification result vectors is changed into the fuzzy AND-operator. The sums are changed into integrals over the continuous values. Retaining the assumption of independent classification events, the expected agreement by chance is formalized by

$$p_c^{\mathcal{F},F} = \sum_{c=1}^{C} \int_{\zeta_c^{\mathfrak{C}_1}=0}^{1} \cdots \int_{\zeta_c^{\mathfrak{C}_Q}=0}^{1} \left(\prod_{q=1}^{Q} \pi\left(\zeta_c^{\mathfrak{C}_q}\right) \right) \cdot \left(\bigwedge_{r=1}^{Q} \zeta_c^{\mathfrak{C}_r} \right) \mathrm{d}\zeta_c^{\mathfrak{C}_1} \ldots \mathrm{d}\zeta_c^{\mathfrak{C}_Q}$$

and analogously for the relative agreement

$$p_0^{\mathcal{F},F} = \frac{1}{K}\sum_{k=1}^{K}\sum_{c=1}^{C}\bigwedge_{q=1}^{Q}\zeta_{v_k}^{\mathfrak{C}_q}(c)\,.$$

The integration of the expected and the relative agreement is done in the structurally known way

$$\kappa_F^{\mathcal{F}} = \frac{p_0^{\mathcal{F},F} - p_c^{\mathcal{F},F}}{1 - p_c^{\mathcal{F},F}}$$

yielding the fuzzy Fleiss' kappa. As in the fuzzy Cohen's kappa, the fuzzy interpretation of the AND-operator is ambiguous. A usual choice using t-norms is the minimum norm (Zühlke et al. 2009).

Fuzzy recognition rate (FRR) evaluation

Using a suitable definition of a fuzzy recognition rate we can apply the recognition rate evaluation and $\Bbbk$-fold cross-validation described in section 4.5.2 as well as the leave-one-out validation from section 4.5.2 analogously for fuzzy evaluation. For such a pertinent fuzzy recognition rate definition the contingency values for the single classes c have to be adapted. We define them as

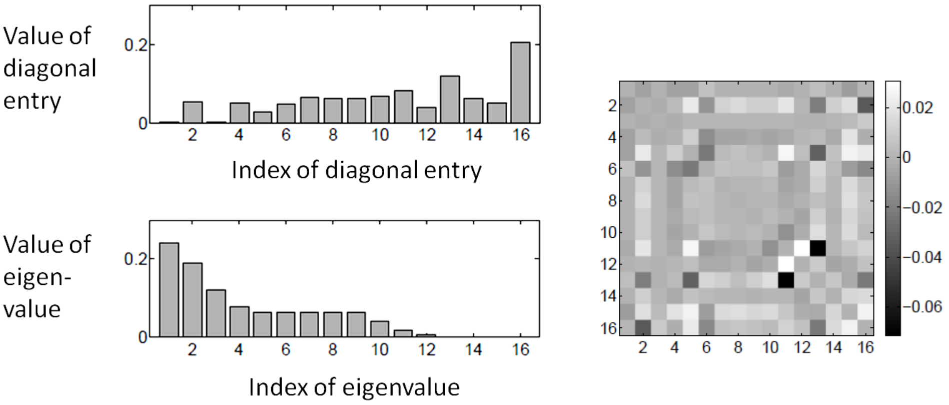
$$\begin{aligned}
\mathrm{TPA}_c &= \sum_{k=1}^{K}\left\{\zeta_{v_k}(c) : z_{v_k} = z_c\right\} && \text{the amount of \textit{true positiveness}}\\
\mathrm{TNA}_c &= \sum_{k=1}^{K}\left\{\sum_{z_r\neq z_c\in Z}\zeta_{v_k}(r) : z_{v_k} \neq z_c\right\} && \text{the amount of \textit{true negativeness}}\\
\mathrm{FPA}_c &= \sum_{k=1}^{K}\left\{\zeta_{v_k}(c) : z_{v_k} \neq z_c\right\} && \text{the amount of \textit{false positiveness}}\\
\mathrm{FNA}_c &= \sum_{k=1}^{K}\left\{\sum_{z_r\neq z_c\in Z}\zeta_{v_k}(r) : z_{v_k} = z_c\right\} && \text{the amount of \textit{false negativeness}.}
\end{aligned}$$

The overall contingency values are obtained as sums of the class-wise contingency values over all classes, for example

$$\mathrm{TPA} = \sum_{c=1}^{C}\mathrm{TPA}_c.$$

From these values we define the fuzzy recognition rate as

$$\mathrm{FRR} = \frac{\mathrm{TPA}+\mathrm{TNA}}{\mathrm{TPA}+\mathrm{TNA}+\mathrm{FPA}+\mathrm{FNA}}. \tag{4.5.9}$$



(a) Tables of diagonal entries and eigenvalues for an example relevance matrix of GMLVQ

(b) Visualization of off-diagonal entries for an example relevance matrix of GMLVQ (diagonal entries set to zero for this visualization)

Figure 4.3: *Evaluation visualizations for an example relevance matrix of GMLVQ (Schneider et al. 2009)*

4.5.5 Evaluation of metric adaptation results

An important insight possibility is the evaluation of the metric adaptation results as they approximate a relevance voting for the single feature components as well as for combinations thereof. This frequently allows a direct biological interpretation of relevances. As we have seen, there are different kinds of relevance information that can be learned. In the GRLVQ algorithm, cf. section 4.1.2, this is e.g. a vector of weight factors for single feature dimensions.

In the GMLVQ algorithm, cf. section 4.1.2, the relevances are given by a matrix describing relevant feature dimension correlations. Because frequently the diagonal elements dominate over the off-diagonal, visualization of the full matrix is done by setting the diagonal elements to zero (figure 4.3(b)) and depicting them separated, as shown in figure 4.3(a) at the top. The diagonal gives a direct mapping to the relevance of the single feature dimension.

The eigenvalues of the relevance matrix (figure 4.3(a) at the bottom) give a hint on the number of relevant directions in the feature space. In the case where the first two or three eigenvalues of the relevance matrix are significantly larger than the others, it is convenient to project the data points according to this reduced form of the adapted dissimilarity. This projection can directly be used for the cluster visualization in DPP approaches mentioned in section 4.6.2.

Normalization of the data and the used dissimilarity measures has a high influence on relevance learning and its interpretation. Without a suitable normalization, for instance, a small relevance value does not necessarily indicate that the corresponding feature is irrelevant for the classification.

4.6 Visual evaluation of LVQ based classification results

In addition to the numerical evaluation given in the last section, an evaluation of the learning results by the domain experts is mandatory. In this section we introduce pertinent methods for a back projection of the learning results to the original problem domain. This is necessary for the proof of the ecological validity of the results as well as for insight possibilities. We limit the introduction to the visual evaluation of classification results as we do not consider these kinds of visual evaluation for clusterings in our application.

Humans are only able to perceive two or three dimensional scenes. The data have to be fitted into two or three dimensions. Visualizing clustering results of high-dimensional data is always error prone and generally endangered to drop relevant information. The visualization space usually is Euclidean. If learning is based on other dissimilarities, the Euclidean visualization can be misleading for interpretation. Instead of visualization of data points or prototypes the presentation should rather visualize the relation between them.

In the following we explain two methods that are specifically suitable for our application example, see e.g. section 7.3.3, where the dissimilarity between the data points was adapted during learning. The receptive field density diagrams (RFDDs) allow for VQ algorithms to judge the mapping of data points to the prototypes. The data point projections (DPPs) illustrate the dissimilarity relation between the data points and the prototypes respectively.

In the last part of this section we discuss possibilities of inducing domain specific knowledge from the introduced visualizations.

4.6.1 Receptive field density diagrams (RFDD)

After learning, for every prototype the data points in its receptive field are plotted according to their dissimilarity to the prototype. Figure 4.4 gives a crisp supervised example. This shows the density of the data points around a prototype. In a cognitive support system, it is pertinent to use an interactive plot linking the data points to their underlying data.

Comparing the density plots for different prototypes, it is possible to identify prototypes with different variations in the dissimilarity. This can argue for under-represented classes or possibly subclasses if e.g. there is a group of data points with higher dissimilarity to their respective prototype. Running the learning process again with additional prototypes can yield better results.

Using receptive field density diagrams, visual outlier detection is easy. If the data points are close to the prototypes, i.e. the data clouds are compact, the result is credible and can induce domain knowledge.

Labeled RFDDs (lRFDDs) in crisp classification

Using the receptive field density diagrams for the evaluation of crisp clustering a label visualization is needed. In figure 4.4 we show an example from our application: The first two prototypes should represent patients that were healthy five years after the tumor surgery and the last two should represent patients that died in this time period. The data points are color coded with green standing for healthy patients and red standing for dead patients. This way misclassification is highlighted. For the second prototype an outlier is identified.

Fuzzy labeled RFDDS (flRFDDs) in fuzzy classification

The receptive field density diagrams (RFDD) can be adapted to the fuzzy evaluation such that for every prototype all data points with their corresponding assignments are displayed in a two dimensional plot for each prototype. The dissimilarity between the data point and the respective prototype is given on the abscissa whereas the class assignment is given on the ordinate. It can be convenient to uniquely identify the data points in all plots for easy comparison e.g. by color labels of data point indices. Also interactive identification possibilities can ease evaluation.

4.6.2 Data point projections (DPP)

A complex visualization is the mapping of the high-dimensional, and possibly non-metric, data space onto a low dimensional metric visualization space $\mathfrak{V}$. This exploits the full information about the dissimilarities between the data points and prototypes that were inferred during dissimilarity adaptation.

In case of local dissimilarity adaptation the learning result is a set of dissimilarity measures. A single learned dissimilarity measure is commonly associated to one or several prototypes. These dissimilarities are used in the visualization of the corresponding prototypes. For the data points we have to determine which of the learned dissimilarities is applicable for the specific data point. In crisp clustering for every data point we calculate the mapping to the winner prototype. The dissimilarity associated to this prototype is used to calculate the dissimilarity for the corresponding

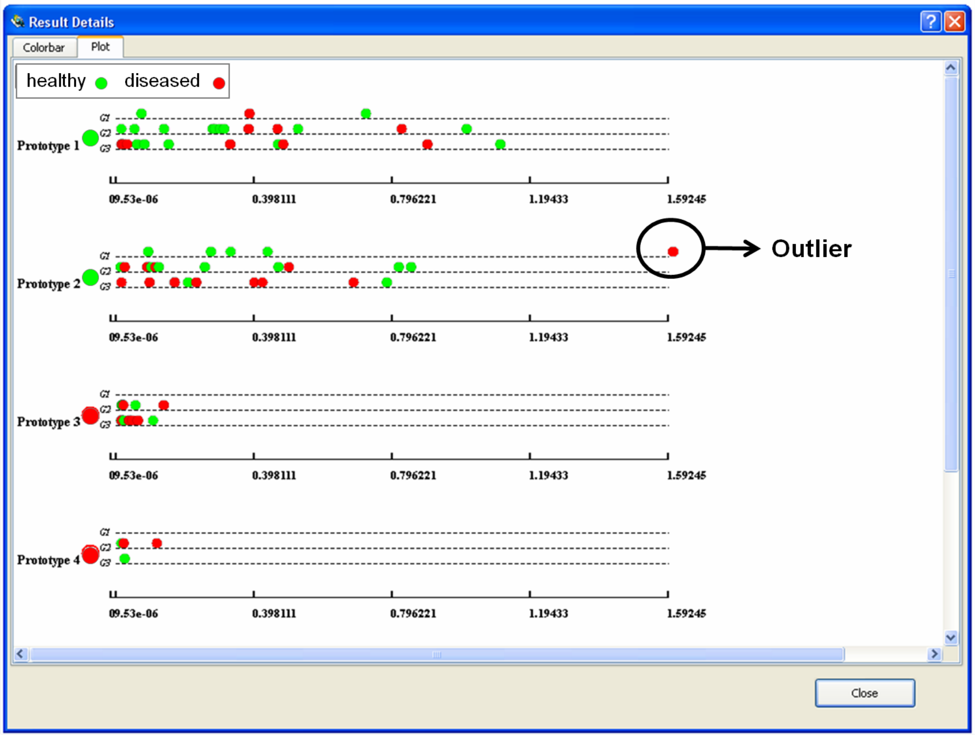


Figure 4.4: *Visualization evaluation example of a classification with four prototypes in two classes using a labeled receptive field density diagram (lRFDD)*

data point to all other data points and all prototypes for visualization with respect to this prototype. The dissimilarities between data can become non-symmetric. This situation becomes even more complex in the case of fuzzy prototype mappings, because the data points are mapped to all prototypes but with different degrees, according to their own dissimilarity.

A particular problem arises if the dissimilarities in the data space are not symmetric, whereas in the visualization space $\mathfrak{V}$ often the Euclidean distance is used that is symmetric. For embedding the data points using these asymmetric dissimilarities it is either necessary to symmetrize the dissimilarities or to choose a visualization method that can cope with these asymmetric dissimilarities between the data points. Both solutions yield an error in the embedding, that has to be kept in mind when conclusions are drawn from the visualization.

The appeal of incorporating all dissimilarities into the visualization lies in the evaluation possibility of relations between the data clouds. If the mapping of the dissimilarities and data points is credible, in addition to the compactness of single cloud representations[10], this visualization allows the judgment of the separation of different cloud representations.

The group of pairwise dissimilarity based visualization methods is capable for the dimension reduction from the high-dimensional data space to a two or three dimensional visualization space. (Hastie, T. et al. 2003) describe multi-dimensional scaling (MDS) as a common example for pairwise dissimilarity based dimension reduction. It minimizes a stress function. This stress function can be chosen to emphasize the reduction of different visualization errors, e.g. large fractional errors. Further examples are given e.g. in (Duda et al. 2001). All stress function definitions are based on the data dissimilarities. If no further constraints are given, this can cause errors in the neighborhood relations.

(Hastie, T. et al. 2003) mention two methods that focus on the preservation of the neighborhood: isomap and locally-linear embedding. Their disadvantage is that they are often restricted to special kinds of dissimilarities (e.g. metrics). Local multidimensional scaling (Local MDS) as introduced by (Chen and Buja 2009) tries to combine the advantages of both approaches. In the stress function the small dissimilarities given in the dissimilarity data are used for building local embeddings. The stress function is stabilized by the introduction of repulsion between points with large distances.

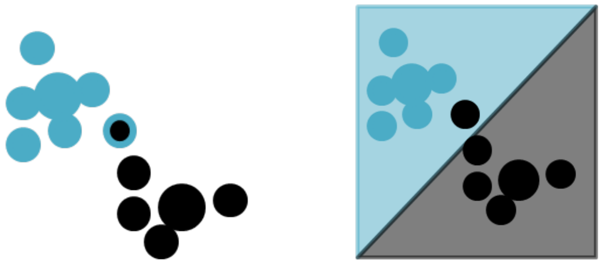
All data point projection methods discussed so far intrinsically minimize some cost criterion. The optimization of these criteria can end up in local minima. The difference between the dissimilarities in the projected space and the dissimilarities in the original space can be evaluated using a normal or a non-metric Shepard plot. Figure 7.9 fives an example for a Shepard plot example in our application. It is recommended to do this comparison for dissimilarities that came up using different stress functions, e.g. Sammon mapping, stress, metric-stress; cf. (Buja et al. 2008) for details.

Labeled DPPs (lDPPs) in crisp classification

To use data point projections in the evaluation of crisp labeled learning results, we map the class labels to artificial colors. It is possible to visualize the actual class label of the data points by coloring the drawn points correspondingly. The prototypes with their class labels are highlighted by larger points with the corresponding color.

We can also visualize the predicted and the actual class label of the data points at the same time. In figure 4.5 two possibilities of class label visualizations are shown

[10] which is suitably evaluable in RFDDs



(a) Visualizing predicted class via outer circle's and actual class as inner circle's color

(b) Visualizing predicted class via background color and actual class as circle's color

Figure 4.5: *Ideas for visualization of predicted and actual class label at the same time.*

for an artificial two class problem. In this figure the prototypes are represented by circles with greater diameter than the data points.

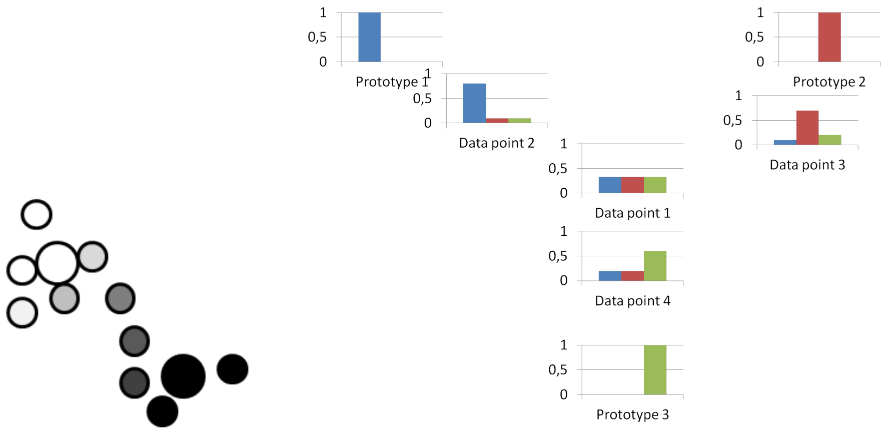
Fuzzy labeled DPPs (flDPPs) in fuzzy classification

The RFDDs for visualization of crisp labeled data are correspondingly applicable for fuzzy labeled data if the fuzzy class assignments are used in the visualization of the data points. Each class z_c, is assigned a color $\mathfrak{c}_c = (r_c, g_c, b_c)$. The respective color for the predicted fuzzy label of a data point v_k is a mixture, according to

$$\mathfrak{c}_W\left(v_k\right) = \sum_{c=1}^{C} \zeta_{v_k}\left(c\right) \cdot \mathfrak{c}_c = \left(r_k, g_k, b_k\right). \tag{4.6.1}$$

where $\vec{\zeta}_{v_k} = \left(\zeta_{v_k}\left(1\right), \ldots, \zeta_{v_k}\left(C\right)\right)$ is the corresponding class membership vector. This color can be chosen as body color of the data point display. This visualization is only practicable for a small number of clusters. For many classes it can lead to indistinguishable mixtures of color. To visualize the true label e.g. the border of the data point can be colored correspondingly. For true fuzzy labels the same calculation base as for the predicted fuzzy label can be used. If true crisp labels are available for the data points, the respective class color is displayed. The prototypes can be visualized as larger points. This concept is exemplified in figure 4.6(a).

If many clusters or many data points have to be visualized, one possibility is the visualization of the membership degrees of every displayed data point in a bar chart. Figure 4.6(b) exemplarily shows such a visualization. The prototypes are highlighted by the title of the charts.



(a) Visualizing fuzzy cluster memberships for crisp data points by gradually mixing cluster colors

(b) Visualizing fuzzy cluster memberships by bar diagrams

Figure 4.6: *Ideas for visualization of fuzzy cluster memberships in fuzzy data point projections.*

4.6.3 Knowledge gain by visual evaluation

In this section we introduce possibilities to use the visual evaluation approaches for inducing domain specific knowledge. We restrict this introduction to approaches that we used in our application example.

General domain knowledge gain by outlier analysis

Outlier detection yields a high potential for deeper insights into the data and domain specific knowledge. There exist many specialized methods that cope with the task of detecting outliers. The receptive field density diagrams are well suited for this approach. In the evaluation of the identified outliers there are different conclusion possibilities for model or data adaption.

One possibility is that the data point that was identified as an outlier from the experts point of view is no outlier and has to be integrated into the model. In this case the model can be inadequate so that it has to be adapted for example in terms of the number of prototypes. If that does not improve the situation the choice of a different learning algorithm is possible. If both improvement possibilities fail, new clusters or classes are created by hand, integrating human expert knowledge which can be difficult to capture by any clustering or classification strategy.

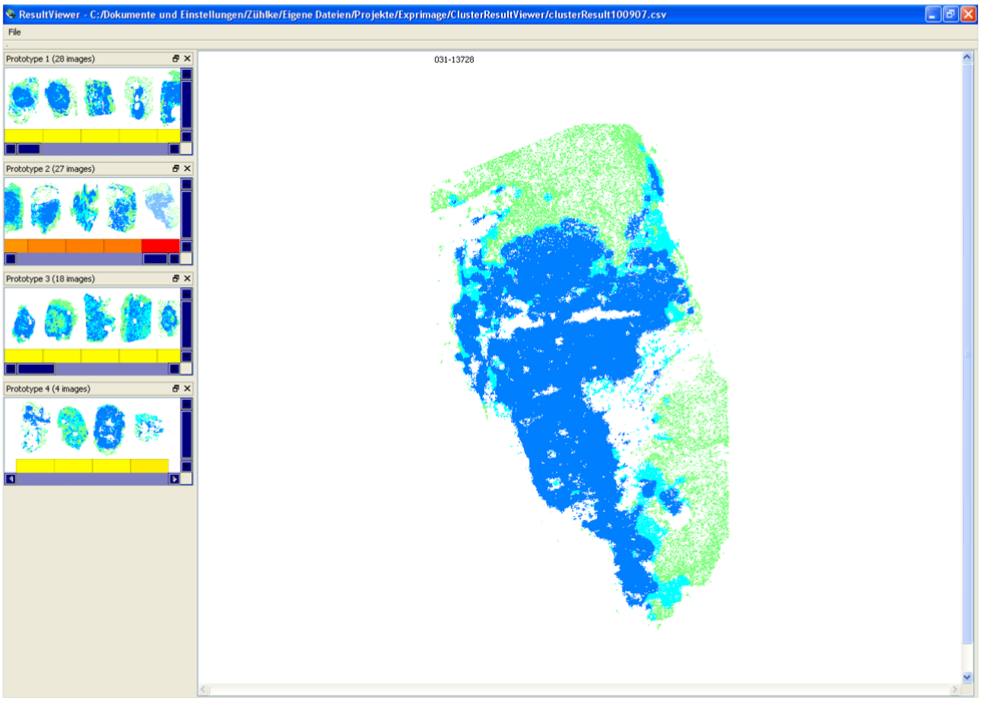

ResultViewer - C:/Dokumente und Einstellungen/Zühlke/Eigene Dateien/Projekte/Exprimage/ClusterResultViewer/clusterResult100907.csv
File
Prototype 1 (28 images)
Prototype 2 (27 images)
Prototype 3 (18 images)
Prototype 4 (4 images)
031-13728

Figure 4.7: *SOM evaluation image of the outlier identified in figure 4.4. It is a patient case that underwent a bad slice preparation.*

If in contrast the expert decides in the evaluation that the data point is a real outlier, it is recommended to remove it from the data space. Thus additionally this instrument is a tool for quality control of the underlying data generation or measurement methods. Figure 4.7 shows the outlier identified in figure 4.4. It is a patient case that underwent a bad slice preparation.

4.6.4 Specific domain knowledge gain by receptive field density diagrams or data point projections

If the data clouds or classes in this visualization are compact, the biomedical experts can analyze the data points mapped to one prototype to possibly find an underlying biomedical concept for grouping these data points. The learned dissimilarities with the relevance votings for the single feature groups should be taken into account, as they lead to this compact data representation and their relevance weights can give hints for the biomedical interpretation.

Chapter 5

Mathematical framework for learning mixed data

This chapter introduces a framework developed for learning of mixed and structured data. In the first section we will give some further nomenclature and settings. We furthermore introduce the two principle ways of integrating different structural components with different dissimilarity measures that will be used later in the algorithms we developed.

In this chapter we start with supervised online learning of mixed data in the two different kinds of integration for different dissimilarity measures. The second part of the chapter is concerned with unsupervised learning in a batch manner.

5.1 Nomenclature

From now on we assume that the input objects are no longer encoded by simple vectors of numbers but rather by data points comprising different feature groups of possibly different data types. Still we use v_k to refer to an input data point but we use the notation $[v_k]_{[j]}$ with $j = 1, \dots, J$ to refer to the single feature groups. The term d_j denotes the dissimilarity measure that is used to compare the j^{th} feature group in two data points.

5.1.1 Integrating different dissimilarities into a combined dissimilarity

For the integration of different dissimilarities d_j, $j = 1, \dots, J$ into one combined dissimilarity measure $D_\star$ the ideas used in the Generalized Relevance Learning Vector Quantization (GRLVQ) and the Generalized Matrix Learning Vector Quantization (GMLVQ) on parameterized dissimilarities were an inspiration for our work. We realized two different integration variants:

1. *Vector-based integration*: Inspired by the GRLVQ, the overall dissimilarity $D_\star = D_\alpha$ is a weighted sum of the dissimilarities d_j in the single feature groups j, given by

$$D_\alpha\left(v_k, w_n\right) := \sum_{j=1}^{J} \left(\alpha_j^n\right)^2 d_j \left(\left[v_k\right]_{[j]}, \left[w_n\right]_{[j]}\right) \tag{5.1.1}$$

with the constraint that $\sum_{j=1}^{J} \left(\alpha_j^n\right)^2 = 1$ for all $n = \{1, \ldots, N\}$.

2. *Matrix-based integration*: Inspired by the GMLVQ, the overall dissimilarity $D_\star = D_\Lambda$ is the weighted sum of the dissimilarities d_j in the single feature groups j and weighted first order combinations of them, given by

$$D_\Lambda\left(v_k, w_n\right) := \begin{pmatrix} d_1\left(\left[v_k\right]_{[1]}, \left[w_n\right]_{[1]}\right) \\ \vdots \\ d_J\left(\left[v_k\right]_{[J]}, \left[w_n\right]_{[J]}\right) \end{pmatrix}^\top \Lambda_n \begin{pmatrix} d_1\left(\left[v_k\right]_{[1]}, \left[w_n\right]_{[1]}\right) \\ \vdots \\ d_J\left(\left[v_k\right]_{[J]}, \left[w_n\right]_{[J]}\right) \end{pmatrix} \tag{5.1.2}$$

with the constraint that Λ_n is symmetric and positive-definite for all $n = \{1, \ldots, N\}$.

So both overall dissimilarity measures are parameterized dissimilarities where the parameters can be adapted during training using a stochastic gradient or other suitable optimization methods.

Because we assume that the single d_j only operate on the corresponding feature groups $\left[v_k\right]_{[j]}$ and $\left[w_n\right]_{[j]}$, we may neglect this explicit nomenclature and we abbreviate $d_j\left(\left[v_k\right]_{[j]}, \left[w_n\right]_{[j]}\right)$ by $d_j\left(v_k, w_n\right)$.

5.2 Learning mixed data using only dissimilarity functions

In this section we introduce methods applicable to such problems where the dissimilarity d_j of each feature group $\left[v_k\right]_{[j]}$ of an object v_k to another one can be determined by a dissimilarity function instantaneously.

5.2.1 Unsupervised variants of mixed data learning using only dissimilarity functions

In the following we demonstrate the integration of mixed data for batch learning algorithms using BNG, cf. section 4.1.1. In the first part we show the vector-based integration and in the second part the matrix-based integration of mixed data.

Batch Neural Gas for vector-based integration of mixed data with dissimilarity functions – vb-BNG

For this first variant we integrated the vector-based overall distance in equation (5.1.1) into the cost function for Batch Neural Gas as given in equation (4.1.2)

$$E_{\text{vb-BNG}} = \sum_{n=1}^{N} \sum_{k=1}^{K} h_\sigma\big(r_W\left(v_k, w_n\right)\big) \cdot D_\alpha\left(v_k, w_n\right) \tag{5.2.1}$$

where $r_W\left(v_k, w_n\right)$ is the dissimilarity rank as given in equation (4.1.3) but with the corresponding dissimilarity

$$r_W\left(v_k, w_n\right) = r_{k,n} = \Big|\big\{w_l : D_\alpha\left(v_k, w_l\right) < D_\alpha\left(v_k, w_n\right)\big\}\Big|.$$

Note that now the v_k are objects and that the prototype vectors w_n have the same object structure. Together with the constraint that $\sum_{j=1}^{J} \alpha_j^n = 1$ for every prototype w_n and the dissimilarity D_α given in equation (5.1.1), we get the following Lagrange function

$$\begin{aligned}\mathfrak{L}\left(W, \alpha\right) = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \cdot \sum_{j=1}^{J} \left(\alpha_j^n\right)^2 \cdot d_j\left(v_k, w_n\right) \\ - \sum_{n=1}^{N} \lambda_n \left(\sum_{j=1}^{J} \alpha_j^n - 1\right)\end{aligned} \tag{5.2.2}$$

To update the prototype vectors, we treat each feature group separately. As in BNG we consider the root of the derivative of $\mathfrak{L}$ – this time with respect to a single feature group $j\star$ of the prototype vector w_n

$$\frac{\partial \mathfrak{L}\left(W, \alpha\right)}{\partial \left[w_n\right]_{[j\star]}} = \frac{1}{2} \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \cdot \left(\alpha_{j\star}^n\right)^2 \cdot \frac{\partial d_{j\star}\left(v_k, w_n\right)}{\partial \left[w_n\right]_{[j\star]}} \overset{!}{=} 0.$$

For solving this equation and determining the new $\left[w_n\right]_{[j\star]}$ it is necessary to know the structure of the $d_{j\star}$ under consideration.

In the example of the squared Euclidean distance function used as dissimilarity function $d_{j\star}$ the derivative is determined as

$$\frac{\partial \mathfrak{L}\left(W, \alpha\right)}{\partial \left[w_n\right]_{[j\star]}} = \frac{1}{2} \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \left(\alpha_{j\star}^n\right)^2 \left(\left[w_n\right]_{[j\star]} - \left[v_k\right]_{[j\star]}\right).$$

Setting this term to zero we get

$$\left[w_n\right]_{[j\star]} = \frac{\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \left[v_k\right]_{[j\star]}}{\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)} \tag{5.2.3}$$

for the prototype vector in the corresponding feature group. A more detailed description of the derivation and realizations for other dissimilarity measures can be found in the appendix A.

To adapt the dissimilarity parameter vector $\vec{\alpha}^n$, the weights for the single feature groups are considered separately. To yield the update rules for the dissimilarity parameter of the $j\star^{\text{th}}$ feature group, in the derivation of the Lagrange function for $\alpha_{j\star}^n$ we get

$$\frac{\partial \mathfrak{L}\left(W,\alpha\right)}{\partial \alpha_{j\star}^n} = \frac{1}{2}\cdot\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) 2\alpha_{j\star}^n d_{j\star}\left(v_k, w_n\right) - \lambda_n .$$

This must be 0 and we obtain

$$\begin{aligned}\lambda_n &= \frac{1}{2}\cdot\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) 2\alpha_{j\star}^n d_{j\star}\left(v_k, w_n\right)\\ &= \alpha_{j\star}^n\cdot\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) d_{j\star}\left(v_k, w_n\right)\end{aligned}$$

and

$$\alpha_{j\star}^n = \lambda_n\left(\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) d_{j\star}\left(v_k, w_n\right)\right)^{-1} . \tag{5.2.4}$$

If we consider the constraint $\sum_{j=1}^{J}\left(\alpha_j^n\right)^2 = 1$ we get

$$\sum_{j=1}^{J}\left(\alpha_j^n\right)^2 = 1 = \sum_{j=1}^{J}\left(\lambda_n\left(\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) d_j\left(v_k, w_n\right)\right)^{-1}\right)$$

which yields the following substitution for the Lagrange multipliers

$$\lambda_n = \left(\sum_{j=1}^{J}\left(\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) d_j\left(v_k, w_n\right)\right)^{-1}\right)^{-1} . \tag{5.2.5}$$

Combining this substitution with equation 5.2.4 for $\alpha_{j\star}^n$ we get the following update rule:

$$\alpha_{j\star}^n = \left(\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) d_{j\star,k,n}\cdot\sum_{j=1}^{J}\left(\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) d_{j,k,n}\right)^{-1}\right)^{-1} \tag{5.2.6}$$

with $d_{j,k,n} = d_j\left([v_k]_{[j]}, [w_n]_{[j]}\right)$ and $d_{j\star,k,n}$ analogously. This update rule is used in the determination of local dissimilarities. It gives single parameter vectors $\vec{\alpha}^n$ for every prototype w_n. In the case of global updates we additionally get the summation over all prototypes but have only one Lagrange multiplier. This yields

$$\alpha_{j\star} = \left(\sum_{n=1}^{N}\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) d_{j\star,k,n} \cdot \sum_{j=1}^{J}\left(\sum_{n=1}^{N}\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) d_{j,k,n}\right)^{-1}\right)^{-1} \tag{5.2.7}$$

as update rule for a global dissimilarity vector $\vec{\alpha}$. We renormalize the α_j by dividing them by their sum. In algorithm 5.2.1 all update rules are summarized and the whole batch optimization is shown.

Algorithm 5.2.1 *Batch Neural Gas for vector-based integration of mixed data with dissimilarity functions*

```
initialize the dissimilarity parameters $\alpha_j^n$ with $\sum_{j=1}^{J} \alpha_j^n = 1$
initialize the prototype positions $w_n$
repeat
  for all prototypes $w_n$ do
    determine the dissimilarity rank of $w_n$ to all data points according to the dissimilarity given by (5.1.1)
    for all feature groups $j\star$ do
      if $d_{j\star}$ is squared Euclidean then
        set feature group $[w_n]_{[j\star]}$ according to (5.2.3)
      else if $d_{j\star}$ is Kullback-Leibler-Divergence for Gaussians then
        set $\mu_{j\star}^n$ according to (A.3.3)
        set $\sigma_{j\star}^n$ according to (A.3.4)
      else if $d_{j\star}$ is $\gamma$-Divergence with given $\gamma$ then
        set feature group $[w_n]_{[j\star]}$ according to (A.2.2)
      end if
      if local updates then
        set $\alpha_{j\star}^n$ according to (5.2.6)
      else if global updates then
        set $\alpha_{j\star}$ according to (5.2.7)
      end if
      renormalize the dissimilarity parameters
    end for
  end for
until convergence
```

In conjunction with this approach we want to mention recent mathematical findings. The procedure of deriving the vb-BNG is structurally similar to the derivation of FCM or FNG (Villmann et al. 2011). Instead of using $\left(\alpha_{j\star}^{n}\right)^{2}$ it is possible to use the more general $\left(\alpha_{j\star}^{n}\right)^{\mathfrak{f}}$ with $\mathfrak{f} > 1$. All equations have to be changed accordingly.

Batch Neural Gas for matrix-based integration of mixed data with dissimilarity functions – mb-BNG

We introduce the matrix based integration given by equation (5.1.2) into the Batch Neural Gas cost function from equation (4.1.2) and yield the cost function for mb-BNG

$$E_{\text{mb-BNG}}\left(W\right) = \sum_{n=1}^{N}\sum_{k=1}^{K} h_{\sigma}\big(r_W\left(v_k, w_n\right)\big) \cdot D_{\Lambda_n}\left(v_k, w_n\right).$$

This cost function is equivalent in its structure to the Matrix Neural Gas cost function given in equation (4.1.6) in section 4.1.1. We derive the mb-BNG analogously. In batch optimization the ranks $r_W\left(v_k, w_n\right)$ are considered as hidden variables $r_{k,n}$. They are optimized iteratively for fixed Λ_n and W, and the optimal values for Λ_n and W are in turn determined given fixed assignments for $r_{k,n}$.

As in Matrix Neural Gas (Arnonkijpanich et al. 2011) we use the substitution $\Lambda_n = \Omega_n^{\top}\Omega_n$ given in equation (3.1.6). We recall that this substitution forces the symmetry and positive semi-definiteness of Λ. To ensure positive definiteness additionally $\det\Lambda \neq 0$ has to be enforced. To optimize the cost function under this last constraint, we get the following Lagrange function:

$$\mathfrak{L}\left(W, \Omega\right) = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} h_{\sigma}\left(r_{k,n}\right) \cdot d_{\Lambda_n}\left(v_k, w_n\right) - \sum_{n=1}^{N}\lambda_n\left(\det\Lambda_n - 1\right) \tag{5.2.8}$$

with Lagrange parameters $\lambda_n \in \mathbb{R}$.

The derivatives of $\mathfrak{L}\left(W, \Omega\right)$ with respect to the feature groups $\left[w_n\right]_{[j]}$ with $j = 1, \ldots, J$ are

$$\vec{\nabla}_{[w_n]}\mathfrak{L}\left(W, \Omega\right) = \sum_{k=1}^{K} h_{\sigma}\left(r_{k,n}\right)\begin{pmatrix}\frac{\partial}{\partial[w_n]_{[1]}}\\ \vdots\\ \frac{\partial}{\partial[w_n]_{[J]}}\end{pmatrix}\left[\vec{d}(k,n)^{\top}\,\Lambda_n\vec{d}(k,n)\right]$$

using the abbreviation

$$\vec{d}(k,n) = \begin{pmatrix} d_1\left(v_k, w_n\right)\\ \vdots\\ d_J\left(v_k, w_n\right)\end{pmatrix}.$$

Additionally abbreviating

$$\vec{\nabla}_{[w_n]} = \left(\nabla_{[w_n]_{[1]}}, \ldots, \nabla_{[w_n]_{[J]}}\right)$$

with $\nabla_{[w_n]_{[j]}} = \frac{\partial}{\partial [w_n]_{[j]}}$, we get the following derivation

$$\begin{aligned}
\vec{\nabla}_{[w_n]} \mathfrak{L}(W, \Omega) &= \sum_{k=1}^{K} h_\sigma(r_{k,n}) \vec{\nabla}_{[w_n]} \left(\vec{d}(k,n)^\top \Omega_n^\top \Omega_n \vec{d}(k,n)\right) \\
&= \sum_{k=1}^{K} h_\sigma(r_{k,n}) \vec{\nabla}_{[w_n]} \left(\Omega_n \vec{d}(k,n)\right)^2 \\
&= \sum_{k=1}^{K} h_\sigma(r_{k,n}) \, 2 \cdot \Omega_n \vec{\nabla}_{[w_n]} \left(\vec{d}(k,n)\right) \cdot \Omega_n \vec{d}(k,n) \\
&= \sum_{k=1}^{K} h_\sigma(r_{k,n}) \, 2 \cdot \Omega_n \begin{pmatrix} \frac{\partial d_1(v_k, w_n)}{\partial [w_n]_{[1]}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial d_J(v_k, w_n)}{\partial [w_n]_{[J]}} \end{pmatrix} \cdot \Omega_n \vec{d}(k,n) \\
&= 2 \cdot \sum_{k=1}^{K} h_\sigma(r_{k,n}) \Omega_n^\top \Omega_n \vec{d}(k,n) \begin{pmatrix} \frac{\partial d_1(v_k, w_n)}{\partial [w_n]_{[1]}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial d_J(v_k, w_n)}{\partial [w_n]_{[J]}} \end{pmatrix}
\end{aligned}$$

and get

$$\vec{\nabla}_{[w_n]} \mathfrak{L}(W, \Omega) = 2\Lambda_n \cdot \sum_{k=1}^{K} h_\sigma(r_{k,n}) \vec{d}(k,n) \begin{pmatrix} \frac{\partial d_1(v_k, w_n)}{\partial [w_n]_{[1]}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial d_J(v_k, w_n)}{\partial [w_n]_{[J]}} \end{pmatrix}. \tag{5.2.9}$$

This should be zero and has to be solved for concrete choices of the d_j.

To derive the update rules for the dissimilarity parameters, we use the structural equivalence to the MNG (Arnonkijpanich et al. 2011). For determining Λ_n in every optimization step, we calculate the derivative of $\mathfrak{L}$ with respect to Λ_n which gives us

$$\frac{\partial \mathfrak{L}}{\partial \Lambda_n} = \sum_{k=1}^{K} h_\sigma(r_{k,n}) \begin{pmatrix} d_1(v_k, w_n) \\ \vdots \\ d_J(v_k, w_n) \end{pmatrix} \begin{pmatrix} d_1(v_k, w_n) \\ \vdots \\ d_J(v_k, w_n) \end{pmatrix}^\top - \lambda_n \left(\det \Lambda_n \cdot \Lambda_n^{-1}\right).$$

We search for the stationary points of the Lagrange function – this time considering Λ_n. We obtain the following equation

$$\Lambda_n = \left(\sum_{k=1}^{K} h_\sigma \left(r_{k,n} \right) \begin{pmatrix} d_1 \left(\left[v_k \right]_{[1]}, \left[w_n \right]_{[1]} \right) \\ \vdots \\ d_J \left(\left[v_k \right]_{[J]}, \left[w_n \right]_{[J]} \right) \end{pmatrix} \begin{pmatrix} d_1 \left(\left[v_k \right]_{[1]}, \left[w_n \right]_{[1]} \right) \\ \vdots \\ d_J \left(\left[v_k \right]_{[J]}, \left[w_n \right]_{[J]} \right) \end{pmatrix}^\top \right)^{-1} \lambda_n,$$

which yields

$$\Lambda_n = S_n^{-1} \left(\det S_n \right)^{\frac{1}{K}} \tag{5.2.10}$$

with

$$S_n = \sum_{k=1}^{K} h_\sigma \left(r_{k,n} \right) \begin{pmatrix} d_1 \left(\left[v_k \right]_{[1]}, \left[w_n \right]_{[1]} \right) \\ \vdots \\ d_J \left(\left[v_k \right]_{[J]}, \left[w_n \right]_{[J]} \right) \end{pmatrix} \begin{pmatrix} d_1 \left(\left[v_k \right]_{[1]}, \left[w_n \right]_{[1]} \right) \\ \vdots \\ d_J \left(\left[v_k \right]_{[J]}, \left[w_n \right]_{[J]} \right) \end{pmatrix}^\top \tag{5.2.11}$$

The mentioned update rules are valid for $\det \Lambda_n = 1$, which has to yield in the initialization and is ensured during adaption by the constraint in the Lagrange function in equation (5.2.8). For an algorithmic implementation of mb-BNG and a general formulation of the update rules the specification of all d_j is required.

5.2.2 Supervised variants of mixed data learning using only dissimilarity functions

In this section we investigate the implementation of both vector- and matrix-based integration of mixed data into a supervised online variant for Vector Quantization, the Generalized Learning Vector Quantization as discussed in section 4.1.2.

Generalized Learning Vector Quantization for vector-based integration of mixed data with dissimilarity functions – vb-GLVQ

For this algorithm we introduce the vector-based overall dissimilarity into the cost function of the normal GLVQ, see equation (4.1.22). We use the loss function

$$L \left(\mu_k \right) = \left(1 + \exp(-\mu_k) \right)^{-1}$$

as it was given in equation 4.1.21 for GLVQ with

$$\mu_k = \frac{D_\alpha^+ \left(v_k \right) - D_\alpha^- \left(v_k \right)}{D_\alpha^+ \left(v_k \right) + D_\alpha^- \left(v_k \right)} \tag{5.2.12}$$

adapted from equation (4.1.20) with changed dissimilarity. Then the cost function of the vb-GLVQ is

$$\begin{aligned} E_{\text{vb-GLVQ}} &= \sum_{k=1}^{K} L\Big(\mu_{\alpha}^{k}\big(D_{\alpha}\left(v_k, w_+\right), D_{\alpha}\left(v_k, w_-\right)\big)\Big) \\ &= \sum_{k=1}^{K} L\left(\frac{D_{\alpha}\left(v_k, w_+\right) - D_{\alpha}\left(v_k, w_-\right)}{D_{\alpha}\left(v_k, w_-\right) + D_{\alpha}\left(v_k, w_-\right)}\right). \end{aligned}$$

Using the definition (5.1.1) for the vector-based integration in D_α we get

$$E_{\text{vb-GLVQ}} = \sum_{k=1}^{K} L\left(\frac{\sum_{j=1}^{J}\left(\alpha_j^+\right)^2 d_j^+\left(v_k\right) - \sum_{j=1}^{J}\left(\alpha_j^-\right)^2 d_j^-\left(v_k\right)}{\sum_{j=1}^{J}\left(\alpha_j^+\right)^2 d_j^+\left(v_k\right) + \sum_{j=1}^{J}\left(\alpha_j^-\right)^2 d_j^-\left(v_k\right)}\right) \tag{5.2.13}$$

where the winning prototypes w_+ and w_- are defined as before in equations (4.1.18) and (4.1.19) but now according to the overall dissimilarity function D_α. We abbreviated $D_\alpha\left(v_k, w_+\right)$ by D_α^+ and use $d_j^+\left(v_k\right)$ instead of $d_j\Big([v_k]_{[j]}, [w_+]_{[j]}\Big)$ for the dissimilarity in the single feature groups. Furthermore we abbreviate

$$\mu_{\alpha}^{k}\big(D_{\alpha}\left(v_k, w_+\right), D_{\alpha}\left(v_k, w_-\right)\big)$$

in the following by μ_α^k. The weight update for the feature group $j\star$ of prototype w_+ with respect to data point v_k is obtained as derivative of the cost function

$$\Delta[w_+]_{[j\star]} \propto -\frac{\partial L\left(v_k, W, \alpha\right)}{\partial \mu_\alpha^k} \cdot \frac{\partial \mu_\alpha^k}{\partial D_\alpha^+\left(v_k\right)} \cdot \frac{\partial D_\alpha^+\left(v_k\right)}{\partial d_{j\star}^+\left(v_k\right)} \cdot \frac{\partial d_{j\star}^+\left(v_k\right)}{\partial [w_+]_{[j\star]}}.$$

We have

$$\begin{aligned} \frac{\partial L\left(v_k, W, \alpha\right)}{\partial \mu_\alpha^k} &= \frac{\exp(-\mu_\alpha^k)}{\left(1 + \exp(-\mu_\alpha^k)\right)^2}, \\ \frac{\partial \mu_\alpha^k}{\partial D_+^\alpha\left(v_k\right)} &= \frac{2 D_\alpha^-\left(v_k\right)}{\left(D_\alpha^+\left(v_k\right) + D_\alpha^-\left(v_k\right)\right)^2} \quad \text{and} \\ \frac{\partial D_\alpha^+\left(v_k\right)}{\partial d_{j\star}^+\left(v_k\right)} &= \left(\alpha_{j\star}^+\right)^2. \end{aligned}$$

Using these equations we get

$$\Delta[w_+]_{[j\star]} = -\epsilon_w \cdot \frac{2 \cdot \exp(-\mu_\alpha^k)}{\left(1 + \exp(-\mu_\alpha^k)\right)^2} \cdot \frac{\left(\alpha_{j\star}^+\right)^2 D_\alpha^-\left(v_k\right)}{\left(D_\alpha^+\left(v_k\right) + D_\alpha^-\left(v_k\right)\right)^2} \cdot \frac{\partial d_{j\star}^+\left(v_k\right)}{\partial [w_+]_{[j\star]}} \tag{5.2.14}$$

and analog for $[w_-]_{[j\star]}$

$$\Delta[w_-]_{[j\star]} = \epsilon_w \cdot \frac{2 \cdot \exp(-\mu_\alpha^k)}{(1+\exp(-\mu_\alpha^k))^2} \cdot \frac{\left(\alpha_{j\star}^-\right)^2 D_\alpha^+(v_k)}{\left(D_\alpha^+(v_k) + D_\alpha^-(v_k)\right)^2} \cdot \frac{\partial d_{j\star}^-(v_k)}{\partial [w_-]_{[j\star]}}. \tag{5.2.15}$$

For different dissimilarities the derivatives $\frac{\partial d_{j\star}^+(v_k)}{\partial [w_+]_{[j\star]}}$ and $\frac{\partial d_{j\star}^-(v_k)}{\partial [w_-]_{[j\star]}}$ differ accordingly. A choice of dissimilarity measures and their derivatives for the prototype weights are given in appendix B.

The adaptation scheme of the weighting parameters $\alpha_{j\star}^+$ follows from the respective derivative of the cost function:

$$\Delta\alpha_{j\star}^+ \propto -\frac{\partial E_{vb-GLVQ}}{\partial \alpha_{j\star}^+} = -\frac{\partial L(v_k, W, \alpha)}{\partial \mu_\alpha^k} \cdot \frac{\partial \mu_\alpha^k}{\partial D_\alpha^+(v_k)} \cdot \frac{\partial D_\alpha^+(v_k)}{\partial \alpha_{j\star}^+}$$

with

$$\begin{aligned} \frac{\partial L(v_k, W, \alpha)}{\partial \mu_\alpha^k} &= \frac{\exp(-\mu_\alpha^k)}{\left(1+\exp(-\mu_\alpha^k)\right)^2}, \\ \frac{\partial \mu_\alpha^k}{\partial D^+(v_k)} &= \frac{2 \cdot D^-(v_k)}{\left(D^+(v_k) + D^-(v_k)\right)^2} \text{ and} \\ \frac{\partial D^+(v_k)}{\partial \alpha_{j\star}^+} &= 2 \cdot \alpha_{j\star}^+ d_{j\star}^+. \end{aligned}$$

Thus for local updates of $\alpha_{j\star}^+$ we obtain

$$\Delta\alpha_{j\star}^+ = -\epsilon_\alpha \cdot \frac{4 \cdot \exp(-\mu_\alpha^k)}{\left(1+\exp(-\mu_\alpha^k)\right)^2} \cdot \frac{\alpha_{j\star}^+ d_{j\star}^+(v_k) D_\alpha^-(v_k)}{\left(D_\alpha^+(v_k) + D_\alpha^-(v_k)\right)^2} \tag{5.2.16}$$

and analog for $\alpha_{j\star}^-$:

$$\Delta\alpha_{j\star}^- = \epsilon_\alpha \cdot \frac{4 \cdot \exp(-\mu_\alpha^k)}{\left(1+\exp(-\mu_\alpha^k)\right)^2} \cdot \frac{\alpha_{j\star}^- d_{j\star}^-(v_k) D_\alpha^+(v_k)}{\left(D_\alpha^+(v_k) + D_\alpha^-(v_k)\right)^2}. \tag{5.2.17}$$

For global updates according to the chain rule for derivation we get:

$$\Delta\alpha_{j\star} = -\epsilon_\alpha \cdot c_\alpha \cdot \left(\frac{\alpha_{j\star}^+ d_{j\star}^+(v_k) D_\alpha^-(v_k) - \alpha_{j\star}^- d_{j\star}^-(v_k) D_\alpha^+(v_k)}{\left(D_\alpha^+(v_k) + D_\alpha^-(v_k)\right)^2} \right) \tag{5.2.18}$$

with

$$c_\alpha = \frac{4 \cdot \exp(-\mu_\alpha^k)}{\left(1+\exp(-\mu_\alpha^k)\right)^2}.$$

The dissimilarity parameters are renormalized after adaption by dividing the $\alpha_{j\star}^n$ by $\sum_{j=1}^J \alpha_j^n$ and the $\alpha_{j\star}$ by $\sum_{j=1}^J \alpha_j$ respectively. All update rules are summarized in algorithm 5.2.2.

Algorithm 5.2.2 *Generalized Learning Vector Quantization for vector-based integration of mixed data with dissimilarity functions*

```
initialize the dissimilarity parameters $\alpha_j^n$ with $\sum_{j=1}^{J} \alpha_j^n = 1$
initialize the prototype positions $w_n$
repeat
  randomly choose an input vector $v_k$ from $V$ with its label $z_{v_k}$
  determine $w_+$ and $w_-$ according to $v_k$, $z_{v_k}$ and the overall dissimilarity given in
  equation (5.1.1)
  for all feature groups $j\star$ do
    determine new prototype position for $[w_+]_{[j\star]}$ according to (5.2.14)
    determine new prototype position for $[w_-]_{[j\star]}$ according to (5.2.15)
    if local updates then
      determine new dissimilarity parameter $\alpha_{j\star}^+$ according to (5.2.16)
      determine new dissimilarity parameter $\alpha_{j\star}^-$ according to (5.2.17)
    else if global updates then
      determine new dissimilarity parameter $\alpha_{j\star}$ according to (5.2.18)
    end if
    renormalize the dissimilarity parameters
  end for
until convergence
```

Generalized Learning Vector Quantization for matrix-based integration of mixed data with dissimilarity functions – mb-GLVQ

For the cost function of the matrix-based integration of mixed data extending the GLVQ we combine the matrix-based overall dissimilarity given in equation (5.1.2) with the cost function of the normal GLVQ

$$\begin{aligned} E_{\text{mb-GLVQ}} &= \sum_{k=1}^{K} L\left(\mu_\Lambda^k\right) \\ &= \sum_{k=1}^{K} L\left(\frac{D_\Lambda^+(v_k) - D_\Lambda^-(v_k)}{D_\Lambda^+(v_k) + D_\Lambda^-(v_k)}\right) \end{aligned} \tag{5.2.19}$$

with $L\left(\mu_\Lambda^k\right)$ and μ_Λ^k as defined for vector-based integration in vb-GLVQ with the accordingly changed dissimilarity. We substitute $\Lambda = \Omega^\top\Omega$. In the derivation of mb-GLVQ we use the structural equivalence to the GMLVQ and use the regularization constraint $\operatorname{trace}(\Lambda) = 1$ following the suggestion of (Schneider 2010) for GMLVQ (cf. section 4.1.2). The derivative of the mb-GLVQ cost function with respect to the nearest prototype of the same class is given by

$$\frac{\partial E_{\text{mb-GLVQ}}}{\partial w^{+}_{[j\star]}} = \frac{\partial L\left(v_k, W, \alpha\right)}{\partial \mu^k_\Lambda} \cdot \frac{\partial \mu^k_\Lambda}{\partial D^{+}_{\Lambda}\left(v_k\right)} \cdot \frac{\partial D^{+}_{\Lambda}\left(v_k\right)}{\partial d^{+}_{j\star}\left(v_k\right)} \cdot \frac{\partial d^{+}_{j\star}\left(v_k\right)}{\partial \left[w_{+}\right]_{[j\star]}}$$

with

$$\begin{aligned}\frac{\partial L\left(v_k, W, \alpha\right)}{\partial \mu^k_\Lambda} &= \frac{\exp(-\mu^k_\Lambda)}{\left(1+\exp(-\mu^k_\Lambda)\right)^2},\\ \frac{\partial \mu^k_\Lambda}{\partial D^{+}_{\Lambda}\left(v_k\right)} &= \frac{2\cdot D^{-}_{\Lambda}\left(v_k\right)}{\left(D^{+}_{\Lambda}\left(v_k\right)+D^{-}_{\Lambda}\left(v_k\right)\right)^2} \text{ and}\\ \frac{\partial D^{+}_{\Lambda}\left(v_k\right)}{\partial d^{+}_{j\star}\left(v_k\right)} &= 2\cdot\Omega^\top\Omega.\end{aligned}$$

According to this the update rule for the prototype position is given as

$$\begin{aligned}\Delta[w^{+}]_{[j\star]} =& -\epsilon_w \cdot \frac{4\cdot\exp(-\mu^k_\Lambda)}{\left(1+\exp(-\mu^k_\Lambda)\right)^2} \cdot \frac{D^{-}_{\Lambda}\left(v_k\right)}{\left(D^{+}_{\Lambda}\left(v_k\right)+D^{-}_{\Lambda}\left(v_k\right)\right)^2}\\ &\cdot \sum_{j=1}^{J} \lambda_{j\star j} d^{+}_{j}\left(v_k\right) \cdot \frac{\partial d^{+}_{j\star}\left(v_k\right)}{\partial \left[w_{+}\right]_{[j\star]}}\end{aligned} \tag{5.2.20}$$

and analog for the nearest prototype of a different class, we yield

$$\begin{aligned}\Delta[w^{-}]_{[j\star]} =& \ \epsilon_w \cdot \frac{4\cdot\exp(-\mu^k_\Lambda)}{\left(1+\exp(-\mu^k_\Lambda)\right)^2} \cdot \frac{D^{+}_{\Lambda}\left(v_k\right)}{\left(D^{+}_{\Lambda}\left(v_k\right)+D^{-}_{\Lambda}\left(v_k\right)\right)^2}\\ &\cdot \sum_{j=1}^{J} \lambda_{j\star j} d^{-}_{j}\left(v_k\right) \cdot \frac{\partial d^{-}_{j\star}\left(v_k\right)}{\partial \left[w_{-}\right]_{[j\star]}}.\end{aligned} \tag{5.2.21}$$

According to the derivation in GMLVQ (Schneider 2010), for the update rule of the dissimilarity parameter matrix we calculate the derivate of the cost function with respect to the parameter Ω_{lm}

$$\frac{\partial E_{\text{mb-GLVQ}}}{\partial \Omega_{lm}} = \frac{\partial L\left(v_k, W, \alpha\right)}{\partial \mu^k_\Lambda} \cdot \frac{\partial \mu^k_\Lambda}{\partial \Omega_{lm}}$$

which for local updating of Ω^{+}_{lm} yields

$$\begin{aligned}\Delta\Omega^{+}_{lm} =& -\epsilon_\Omega \frac{2\cdot\exp(-\mu^k_\Lambda)}{\left(1+\exp(-\mu^k_\Lambda)\right)^2} \cdot \frac{D^{-}_{\Lambda}\left(v_k\right)}{\left(D^{+}_{\Lambda}\left(v_k\right)+D^{-}_{\Lambda}\left(v_k\right)\right)^2}\\ &\cdot \left(d^{+}_{m}\left(v_k\right) \cdot \left[\Omega \begin{pmatrix} d^{+}_{1}(v_k) \\ \vdots \\ d^{+}_{J}(v_k) \end{pmatrix} \right]_l \right)\end{aligned} \tag{5.2.22}$$

and accordingly for Ω^{-}_{lm} changes to

$$\begin{aligned}\Delta\Omega^{-}_{lm} =& \epsilon_\Omega \frac{2\cdot\exp(-\mu^k_\Lambda)}{\left(1+\exp(-\mu^k_\Lambda)\right)^2}\cdot\frac{D^{+}_{\Lambda}(v_k)}{\left(D^{+}_{\Lambda}(v_k)+D^{-}_{\Lambda}(v_k)\right)^2}\\ &\cdot\left(d^{-}_{m}(v_k)\cdot\left[\Omega\begin{pmatrix}d^{-}_1(v_k)\\ \vdots\\ d^{-}_J(v_k)\end{pmatrix}\right]_l\right).\end{aligned} \tag{5.2.23}$$

For global updates of Ω we get

$$\begin{aligned}\Delta\Omega_{lm} = -\epsilon_\Omega c_m\cdot &\left\{c_+\cdot\left(d^{+}_{m}(v_k)\cdot\left[\Omega\begin{pmatrix}d^{+}_1(v_k)\\ \vdots\\ d^{+}_J(v_k)\end{pmatrix}\right]_l\right)\right.\\ &\left.-c_-\cdot\left(d^{-}_{m}(v_k)\cdot\left[\Omega\begin{pmatrix}d^{-}_1(v_k)\\ \vdots\\ d^{-}_J(v_k)\end{pmatrix}\right]_l\right)\right\}\end{aligned} \tag{5.2.24}$$

with

$$\begin{aligned}c_m &= \frac{2\cdot\exp(-\mu^k_\Lambda)}{\left(1+\exp(-\mu^k_\Lambda)\right)^2},\\ c_+ &= \frac{D^{-}_{\Lambda}(v_k)}{\left(D^{+}_{\Lambda}(v_k)+D^{-}_{\Lambda}(v_k)\right)^2} \text{ and}\\ c_- &= \frac{D^{+}_{\Lambda}(v_k)}{\left(D^{+}_{\Lambda}(v_k)+D^{-}_{\Lambda}(v_k)\right)^2}.\end{aligned}$$

These matrices have to be renormalized after every step with respect to the condition trace $(\Lambda) = 1$. In this special case $\frac{1}{\sum_x\sum_y\Omega_{xy}}$ can be used as normalization factor (Schneider 2010). All the update steps are summarized in algorithm 5.2.3.

5.3 Learning mixed data containing also relational dissimilarities

In this section we will discuss how to handle feature groups that are given as relational data, i.e. as indexed data points with a corresponding dissimilarity matrix. For relational data we assume the matrix of dissimilarities to be error-free embeddable into a finite-dimensional Euclidean space. We refer to section 3.4 for details. Under

Algorithm 5.2.3 *Generalized Learning Vector Quantization for matrix-based integration of mixed data with dissimilarity functions*

```
initialize the dissimilarity parameter matrix $\Omega^n$ with trace $\left(\Omega^\top \Omega\right) = 1$
initialize the prototype positions $w_n$
repeat
  randomly choose an input vector $v_k$ from $V$ with its label $z_{v_k}$
  determine $w_+$ and $w_-$ according to $v_k$, $z_{v_k}$ and the overall dissimilarity given in (5.1.2)
  for all feature groups $j\star$ do
    determine new prototype position for $[w_+]_{[j\star]}$ according to (5.2.20)
    determine new prototype position for $[w_-]_{[j\star]}$ according to (5.2.21)
    if local updates then
      determine new dissimilarity parameter $\Omega^+_{lm}$ according to (5.2.22)
      determine new dissimilarity parameter $\Omega^-_{lm}$ according to (5.2.23)
    else if global updates then
      determine new dissimilarity parameter $\Omega_{lm}$ according to (5.2.24)
    end if
    renormalize the dissimilarity parameters correspondingly
  end for
until convergence
```

this assumption we can represent the relational feature groups of the prototypes by linear combinations of these groups in the data points. We combine the results from the last section with the new update rules for relational feature groups to yield algorithms that are able to handle both kinds of data. We consider both unsupervised and supervised variants.

5.3.1 Unsupervised variants for learning mixed data containing also relational dissimilarities

We will introduce the integration of the vector-based overall dissimilarities into the batch learning variant of Neural Gas for relational data, Relational Neural Gas, see section 4.2.1 for details. As the incorporation of the matrix-based integration into the Batch Neural Gas for learning using a dissimilarity function did not yield general update rules, we refrain from the additional matrix-based integration for Relational Neural Gas.

Relational Neural Gas for vector-based integration of mixed data containing also relational data – vb-RNG

For handling mixed relational and dissimilarity function based (dfb) data in unsupervised learning we combine vb-BNG from the last section with the idea of Relational Neural Gas for the feature groups that are given by relational data. The update rules for the dfb data remain unchanged. For the relational feature group updates we adapt the Lagrange function for vb-BNG

$$\begin{aligned}\mathfrak{L}(W,\alpha) &= \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} h_\sigma(r_{k,n})\cdot\sum_{j=1}^{J}\left(\alpha_j^n\right)^2\cdot d_j(v_k,w_n)\\ &\quad-\sum_{n=1}^{N}\lambda_n\left(\sum_{j=1}^{J}\alpha_j^n-1\right)\end{aligned}$$

from equation (5.2.2).

The prototypes $[w_n]_{j\star}$ for a relational group $j\star$ are represented by coefficients $\vec{\beta}_n^{j\star}=\left(\left[\beta_n^{j\star}\right]_1,\ldots,\left[\beta_n^{j\star}\right]_K\right)$. The dissimilarity between the data point v_k and the prototype w_n for the relational feature group $j\star$ is expressed by

$$\begin{aligned}d_{j\star}(v_k,w_n) &= \left(D_{j\star}\cdot\vec{\beta}_n^{j\star}\right)_k-\frac{1}{2}\cdot\left(\vec{\beta}_n^{j\star}\right)^\top\cdot D_{j\star}\cdot\vec{\beta}_n^{j\star}\\ &= \sum_{m\in[j\star]} d^{j\star}(v_k,v_m)\left[\beta_n^{j\star}\right]_m-\frac{1}{2}\sum_{i\in[j\star]}\sum_{u\in[j\star]} d^{j\star}(v_i,v_u)\left[\beta_n^{j\star}\right]_i\left[\beta_n^{j\star}\right]_u.\end{aligned} \tag{5.3.1}$$

which is in accordance to the definition in equation (3.4.8). For the derivative $\frac{\partial\mathfrak{L}(W,\alpha)}{\partial\left[\beta_n^{j\star}\right]_l}$ we get

$$\frac{\partial\mathfrak{L}(W,\alpha)}{\partial\left[\beta_n^{j\star}\right]_l}=\left(\alpha_{j\star}^n\right)^2\sum_{k=1}^{K} d(v_k,v_l)\left(h_\sigma(r_{k,n})-\sum_{i=1}^{K}h_\sigma(r_{i,n})\left[\beta_n^{j\star}\right]_k\right). \tag{5.3.2}$$

If the dissimilarity matrix D is nonsingular and for $\alpha_{j\star}^n\neq 0$ this is zero if and only if

$$\left[\beta_n^{j\star}\right]_l=\frac{h_\sigma(r_{l,n})}{\sum_{k=1}^{K}h_\sigma(r_{k,n})} \tag{5.3.3}$$

and so the update rules for the prototype representing coefficients from normal RNG persist. Combining these update rules for the prototype positions with the update rules for the dissimilarity parameters from vb-BNG we obtain algorithm 5.3.1.

Algorithm 5.3.1 *Relational Neural Gas for vector-based integration of mixed data containing also relational data*

```
initialize the dissimilarity parameters $\alpha_j^n$ with $\sum_{j=1}^{J} \alpha_j^n = 1$
initialize prototype positions $w_n$, for the relational groups initialize all prototype representing coefficient vectors $\vec{\beta}_n$ with $\sum_{k=1}^{K} [\beta_n]_k = 1$ for $n = 1, \ldots, N$ and $k = 1, \ldots, K$
repeat
  for all prototypes $w_n$ do
    determine the dissimilarity rank of $w_n$ to all data points according to the dissimilarity given by (5.1.1)
    for all feature groups $j\star$ do
      if $d_{j\star}$ is squared Euclidean then
        set feature group $[w_n]_{[j\star]}$ according to (5.2.3)
      else if $d_{j\star}$ is Kullback-Leibler-Divergence for Gaussians then
        set $\mu_{j\star}^n$ according to (A.3.3)
        set $\sigma_{j\star}^n$ according to (A.3.4)
      else if $d_{j\star}$ is $\gamma$-Divergence with given $\gamma$ then
        set feature group $[w_n]_{[j\star]}$ according to (A.2.2)
      else if $d_{j\star}$ is relational with matrix $D_{j\star}$ of dissimilarities then
        set $\left[\beta_n^{j\star}\right]_k$ according to (5.3.3)
      end if
      if local updates of dissimilarity parameters then
        set $\alpha_{j\star}^n$ according to (5.2.6)
      else if global updates then
        set $\alpha_{j\star}$ according to (5.2.7)
      end if
      renormalize the dissimilarity parameters
    end for
  end for
until convergence
```

5.3.2 Supervised variants for learning mixed data containing also relational dissimilarities

In this section we introduce the integration of the vector-based overall dissimilarities as well as the integration of the matrix-based overall dissimilarity into the variant of Learning Vector Quantization for relational data, Kernel Learning Vector Quantization discussed in section 4.2.2.

Kernel Learning Vector Quantization for vector-based integration of mixed data containing also relational data – vb-KLVQ

To handle a mixture of data with dissimilarity functions and relational data the vector-based integration of dissimilarities as introduced in section 5.2.2 into GLVQ is combined with the idea of handling relational data as given by KLVQ (section 4.2.2). We use the assumption that the total dissimilarity of a prototype vector w_n and a data point v_k in the mixed feature space is the sum of weighted dissimilarities in the different feature groups. As mentioned in vb-RNG, for relational data we can substitute the prototype vectors for the relational groups by a linear combination of the values of the corresponding feature groups in data point vectors.

The update rules for the dfb groups remain unchanged, whereas the update rules for the relational feature groups change from the equations in section 4.2.2 by exchanging the dissimilarities and their derivatives

$$\left[\beta_{+}^{j\star}\right]_l(t+1) = \begin{cases} \left[1 - c \cdot \frac{\left(\alpha_{j\star}^{+}\right)^2 \cdot D_{\alpha}^{-}(v_k)}{\left(D_{\alpha}^{+}(v_k) + D_{\alpha}^{-}(v_k)\right)^2}\right] \cdot \left[\beta_{+}^{j\star}\right]_l(t) & \text{if } v_l \neq v_k \\ \\ \left[1 - c \cdot \frac{\left(\alpha_{j\star}^{+}\right)^2 \cdot D_{\alpha}^{-}(v_k)}{\left(D_{\alpha}^{+}(v_k) + D_{\alpha}^{-}(v_k)\right)^2}\right] \cdot \left[\beta_{+}^{j\star}\right]_l(t) & \\ +c \cdot \frac{\left(\alpha_{j\star}^{+}\right)^2 \cdot D_{\alpha}^{-}(v_k)}{\left(D_{\alpha}^{+}(v_k) + D_{\alpha}^{-}(v_k)\right)^2} & \text{if } v_l = v_k \end{cases} \tag{5.3.4}$$

and

$$\left[\beta_{-}^{j\star}\right]_l(t+1) = \begin{cases} \left[1 + c \cdot \frac{\left(\alpha_{j\star}^{-}\right)^2 \cdot D_{\alpha}^{+}(v_k)}{\left(D_{\alpha}^{+}(v_k) + D_{\alpha}^{-}(v_k)\right)^2}\right] \cdot \left[\beta_{-}^{j\star}\right]_l(t) & \text{if } v_l \neq v_k \\ \\ \left[1 + c \cdot \frac{\left(\alpha_{j\star}^{-}\right)^2 \cdot D_{\alpha}^{+}(v_k)}{\left(D_{\alpha}^{+}(v_k) + D_{\alpha}^{-}(v_k)\right)^2}\right] \cdot \left[\beta_{-}^{j\star}\right]_l(t) & \\ -c \cdot \frac{\left(\alpha_{j\star}^{-}\right)^2 \cdot D_{\alpha}^{+}(v_k)}{\left(D_{\alpha}^{+}(v_k) + D_{\alpha}^{-}(v_k)\right)^2} & \text{if } v_l = v_k \end{cases} \tag{5.3.5}$$

where $c = \epsilon_w \cdot \frac{4 \cdot \exp(-\mu_{\alpha}^{k})}{(1+\exp(-\mu_{\alpha}^{k}))^2}$.

The update rules and normalizations for the dissimilarity parameters α_j^n remain unchanged. All update steps, including those for the dfb data, are summarized in algorithm 5.3.2.

Kernel Learning Vector Quantization for matrix-based integration of mixed data containing also relational data – mb-KLVQ

Mixed data with dfb and relational data can be integrated with a matrix-based approach into GLVQ, combining the idea of handling relational data as given by

Algorithm 5.3.2 *Kernel Learning Vector Quantization for vector-based integration of mixed data containing also relational data*

```
initialize the dissimilarity parameters $\alpha_j^n$ with $\sum_{j=1}^{J} \alpha_j^n = 1$
initialize prototype positions $w_n$, for the relational groups initialize all prototype representing coefficient vectors $[\beta_n]_k$ with $\sum_{k=1}^{K} [\beta_n]_k = 1$ for $n = 1, \ldots, N$ and $k = 1, \ldots, K$
repeat
  randomly choose an input vector $v_k$ from $V$ with its label $z_{v_k}$
  determine $w_+$ and $w_-$ according to $v_k$, $z_{v_k}$ and the overall dissimilarity given in (5.1.1)
  for all feature groups $j\star$ do
    if dissimilarity in $j\star$ is defined by dissimilarity function $d_{j\star}$ then
      determine new prototype position for $[w_+]_{[j\star]}$ according to (5.2.14)
      determine new prototype position for $[w_-]_{[j\star]}$ according to (5.2.15)
    else if dissimilarity in $j\star$ is defined by dissimilarity matrix $D^{j\star}$ then
      determine new prototype representation for all coefficients $[\beta_+]_l\,(t+1)$ according to (5.3.4)
      determine new prototype representation for all coefficients $[\beta_-]_l\,(t+1)$ according to (5.3.5)
    end if
    if local updates then
      determine new dissimilarity parameter $\alpha_{j\star}^+$ according to (5.2.16)
      determine new dissimilarity parameter $\alpha_{j\star}^-$ according to (5.2.17)
    else if global updates then
      determine new dissimilarity parameter $\alpha_{j\star}$ according to (5.2.18)
    end if
    renormalize the dissimilarity parameters
  end for
until convergence
```

KLVQ (see section 4.2.2) with the findings for mb-GLVQ. We substitute the prototype vectors for the relational feature groups by a linear combination of data point vectors for these feature groups.

The update rules for the dfb groups remain unchanged. The update rules for the relational components change from the equations in section 4.2.2 by exchanging the dissimilarities and their derivatives.

$$\left[\beta_{+}^{j\star}\right]_{l}(t+1)=\begin{cases}\left[1-c\cdot\frac{\sum\limits_{j=1}^{J}\lambda_{j\star,j}d_{j}^{+}(v_k)\cdot D_{\Lambda}^{-}(v_k)}{\left(D_{\Lambda}^{+}(v_k)+D_{\Lambda}^{-}(v_k)\right)^{2}}\right]\cdot\left[\beta_{+}^{j\star}\right]_{l}(t) & \text{if } v_l\neq v_k\\ \\ \left[1-c\cdot\frac{\sum\limits_{j=1}^{J}\lambda_{j\star,j}d_{j}^{+}(v_k)\cdot D_{\Lambda}^{-}(v_k)}{\left(D_{\Lambda}^{+}(v_k)+D_{\Lambda}^{-}(v_k)\right)^{2}}\right]\cdot\left[\beta_{+}^{j\star}\right]_{l}(t) & \\ +c\cdot\frac{\sum\limits_{j=1}^{J}\lambda_{j\star,j}d_{j}^{+}(v_k)\cdot D_{\Lambda}^{-}(v_k)}{\left(D_{\Lambda}^{+}(v_k)+D_{\Lambda}^{-}(v_k)\right)^{2}} & \text{if } v_l=v_k\end{cases}\tag{5.3.6}$$

and

$$\left[\beta_{-}^{j\star}\right]_{l}(t+1)=\begin{cases}\left[1+c\cdot\frac{\sum\limits_{j=1}^{J}\lambda_{j\star,j}d_{j}^{-}(v_k)\cdot D_{\Lambda}^{+}(v_k)}{\left(D_{\Lambda}^{+}(v_k)+D_{\Lambda}^{-}(v_k)\right)^{2}}\right]\cdot\left[\beta_{-}^{j\star}\right]_{l}(t) & \text{if } v_l\neq v_k\\ \\ \left[1+c\cdot\frac{\sum\limits_{j=1}^{J}\lambda_{j\star,j}d_{j}^{-}(v_k)\cdot D_{\Lambda}^{+}(v_k)}{\left(D_{\Lambda}^{+}(v_k)+D_{\Lambda}^{-}(v_k)\right)^{2}}\right]\cdot\left[\beta_{-}^{j\star}\right]_{l}(t) & \\ -c\cdot\frac{\sum\limits_{j=1}^{J}\lambda_{j\star,j}d_{j}^{-}(v_k)\cdot D_{\Lambda}^{+}(v_k)}{\left(D_{\Lambda}^{+}(v_k)+D_{\Lambda}^{-}(v_k)\right)^{2}} & \text{if } v_l=v_k\end{cases}\tag{5.3.7}$$

where c is defined as for vb-KLVQ.

The update rules and normalizations for the dissimilarity parameters Λ_n remain unchanged. All update steps, including those for dfb data, are summarized in algorithm 5.3.3.

Algorithm 5.3.3 *Kernel Learning Vector Quantization for matrix-based integration of mixed data with dissimilarity functions and relational data*

```
initialize the dissimilarity parameter $\Lambda_n = \Omega_n^\top \Omega_n$ with trace $(\Lambda_n) = 1$
initialize prototype positions $w_n$, for the relational groups initialize all prototype
representing coefficient vectors $\vec{\beta}_n$ with $\sum_{k=1}^{K} [\beta_n]_k = 1$ for $n = 1, \ldots, N$ and
$k = 1, \ldots, K$
repeat
  randomly choose an input vector $v_k$ from $V$ with its label $z_{v_k}$
  determine $w_+$ and $w_-$ according to $v_k$, $z_{v_k}$ and the overall dissimilarity given
  in (5.1.1)
  for all feature groups $j\star$ do
    if dissimilarity in $j\star$ is defined by dissimilarity function $d_{j\star}$ then
      determine new prototype position for $[w_+]_{[j\star]}$ according to (5.2.14)
      determine new prototype position for $[w_-]_{[j\star]}$ according to (5.2.15)
    else if dissimilarity in $j\star$ is defined by dissimilarity matrix $D^{j\star}$ then
      determine new prototype representation for all coefficients $[\beta_+]_l (t+1)$
      according to (5.3.6)
      determine new prototype representation for all coefficients $[\beta_-]_l (t+1)$
      according to (5.3.7)
    end if
    if local updates then
      determine new dissimilarity parameter $\Omega_{lm}^+$ according to (5.2.22)
      determine new dissimilarity parameter $\Omega_{lm}^-$ according to (5.2.23)
    else if global updates then
      determine new dissimilarity parameter $\Omega_{lm}$ according to (5.2.24)
    end if
    renormalize the dissimilarity parameters
  end for
until convergence
```

Chapter 6

Building a cognitive support system in the breast cancer research project Exprimage

In this chapter we introduce an application example for a cognitive support system in biomedical research. We explain the rationales behind the design decisions and discuss the necessary information processing. The context of the example development is the breast cancer project Exprimage. This project was conducted in close cooperation of the working groups

- at the Fraunhofer Institute for Applied Information Technology,
- the Information Systems Department of the RWTH Aachen and
- the Institute for Diagnostic Histopathology and Cytology at the Pathology Hamburg-West.

Additional support in the mathematical foundation was given by

- the Computational Intelligence and Technomathematics Group at the University of Applied Sciences Mittweida,
- the Department for Theoretic Computer Science at the University of Bielefeld and
- the Intelligent Systems Group at the University of Groningen.

The work was funded by the German Federal Ministry of Education and Research (Fkz 13N9873).

6.1 Objective of the Exprimage project

The objective of the Exprimage project was to improve the adjuvant therapy[1] suggestions in breast cancer by incorporating information from several biomedical domains.

[1] There are two main kinds of therapy: on the one hand possibly neoadjuvant chemotherapy made before surgery, in order to shrink the tumor. On the other hand, there are postoperative, adjuvant chemo- and hormone therapies and radiation.

For good therapy suggestions a detailed individual diagnosis based on the analysis of the resected tumor tissue is an indispensable basis. The Exprimage project was conceived as a retrospective study to analyze the relation between tumor properties and the corresponding most suitable therapy. A comprehensive data base with reliable information concerning the therapeutic course of the single patients is needed for this purpose. Due to several reasons this data base was not established during the project. We did not have reliable information about the therapies the patients underwent. Furthermore the amount of 2000 patient samples that was advised for the study was not achieved. Our data set comprised 93 patients.

We shifted the focus of our subproject according to this data-poor situation. We developed a cognitive support system that according to the methodological requirements is capable of supporting the research on improvements of the therapy suggestions. Instead of answering the actual pathological research question we applied the system in research to improve the prognosis by a detailed diagnosis. This shift was acceptable for two reasons:

- The detailed diagnosis is indispensable for achieving the original objective of improved therapy suggestions.
- In the application of the cognitive support system its principle functionality could be demonstrated.

Our system was based on the image analysis of digitized slice images of breast cancer tissue and the available clinical data for the patients.

6.1.1 Traditional breast cancer diagnostics and its insufficiency

The available selection of cases, called *cohort*, in our subproject was a collection of matched pairs. These are cases where the current standard diagnostics did not generally succeed. For standard diagnosis these cases look quite similar, but they differ in the clinical course and the follow-up of the patient.

The current standard processing of breast cancer diagnosis in diagnostic laboratories is characterized by complex human expert performance in a tight schedule. The tissue that was surgically removed from the patient's breast is sliced and stained with the standard pathological stain that marks structural properties of the tissue. The upper row of figure 6.1 shows this process. Then the following procedure is executed:

1. The pathological expert chooses one single tissue slice.
2. He/she analyzes it by scanning the overview and instantly identifying a few regions of interest (ROIs) with a normal optical microscope.

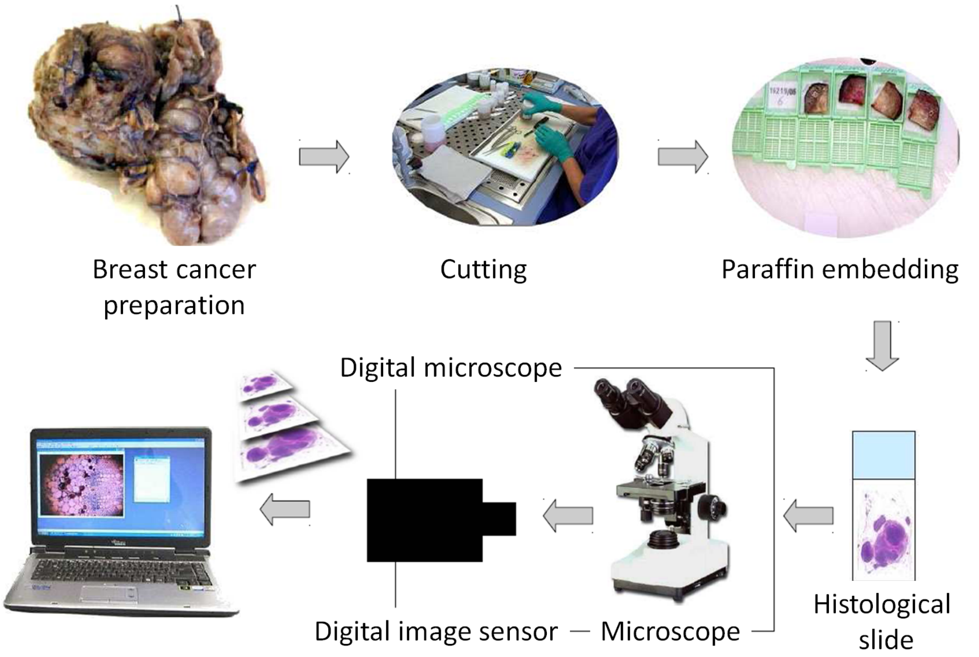


Figure 6.1: *Schematic representation of the clinical probe preparation. In the current standard diagnostic the preparation process ends at a normal optical microscope. In the Exprimage project we used a digital microscope to store the images for computer analysis. This scheme was adapted from (Bornemeier 2011).*

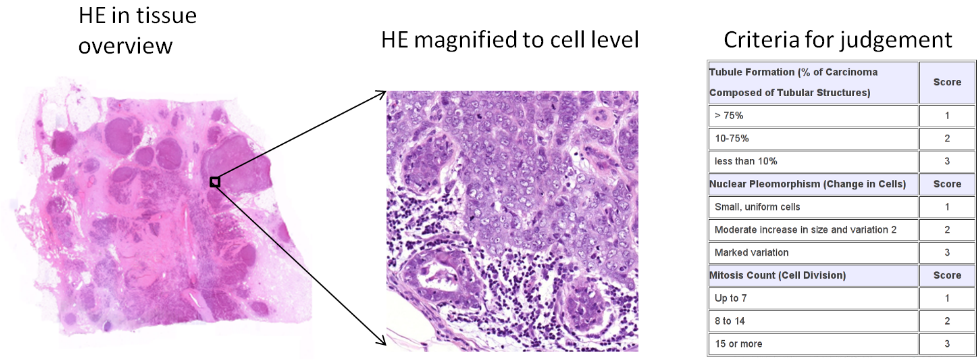
3. In the regions of interest he/she magnifies the slice to a cell detail level.

4. On this level, a collection of commonly accepted analytic items[2] is checked that leads to a *prognostic index*. In Exprimage the pathologists used the current world-wide standard: the Nottingham Index (Galea et al. 1992).

Figure 6.2 shows an example of this first part of the current standard diagnostic process and the corresponding analytic scheme. It also indicates the size of the analyzed region in relation to the whole slice. To complete the diagnostic process, the expert carries out the last step:

5. To gain hints for personalized therapy suggestions, the expert considers selected slices stained with functional markers. They indicate for example the

[2] Such items are e.g. cell nucleus abnormality and mitosis rate. These analytic items are identified in a regular consensus process at the international breast cancer conference in Sankt Gallen. http://www.oncoconferences.ch/dynasite.cfm?dsmid=98911



Tubule Formation (% of Carcinoma Composed of Tubular Structures)	Score
> 75%	1
10-75%	2
less than 10%	3
Nuclear Pleomorphism (Change in Cells)	**Score**
Small, uniform cells	1
Moderate increase in size and variation 2	2
Marked variation	3
Mitosis Count (Cell Division)	**Score**
Up to 7	1
8 to 14	2
15 or more	3

Figure 6.2: *Example of a current standard diagnostic process emphasizing subjectiveness*

expression level of the hormone receptor for estrogen. The expert qualifies this expression level into a categorical scheme analyzing selected ROIs on a cell detail magnification.

Figure 6.3 shows an example of a breast cancer slice stained with a functional marker highlighting the expression of the hormone receptor for estrogen on a tissue detail magnification. The ellipses mark regions with different expressions inside one tumor region. The expert has to compress this information into a singular expression category for the whole patient, according to non-quantitative categories yes/no. This yields an information loss. To summarize, we see that the current standard diagnostic process in breast cancer is subjective especially in two perspectives:

- Selection: The pathologist chooses the slices as well as a few, small ROIs on these slices and only considers the cell detail level. This gives a subjective, selective opinion on the patient's situation.

- Qualification: There is no reproducible, reliable mapping of the patient's situation displayed in the ROIs in the chosen slices onto the scores used as analytic items for the prognostic index. The same holds for the categories indicating the functional marker expression level. The result is a subjective, qualitative description of the patient's situation.

6.1.2 Clues for improvement

In pathological literature there is a critical trend showing that improvements of the standard diagnostic process are possible. An exhaustive and quantitative analysis using automatic image analysis of digitized slices of the pathological probes leads to

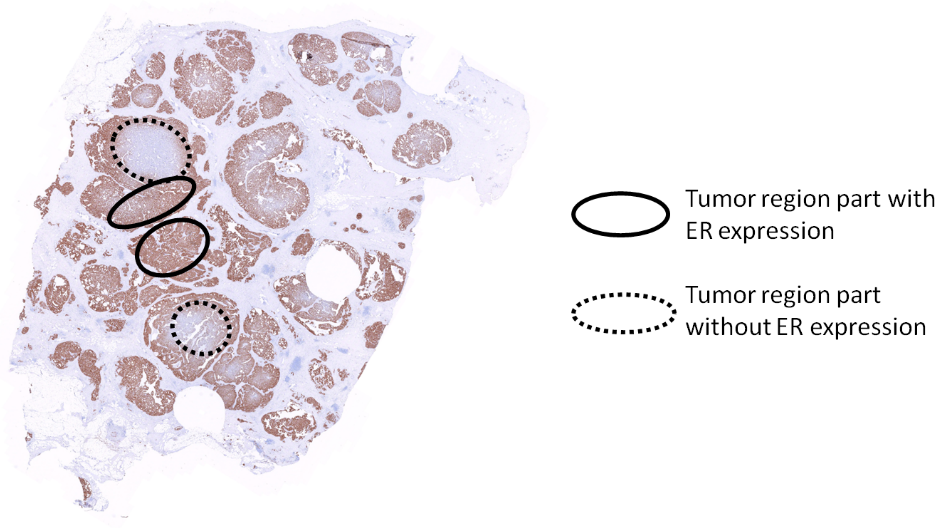

Tumor region part with ER expression
Tumor region part without ER expression

Figure 6.3: *Example of a hormone receptor expression level image emphasizing heterogeneity in functional marker expression*

a significant improvement in diagnostic precision and therapy suggestions for breast cancer by

- widening the selection of the material used e.g. by incorporating information from the tissue overview like in (Rangayyan et al. 1997). In these studies measures describing the acutance and shape of the tumor achieved a correct prediction rate of 95% for the malignancy or benignity of a breast tumor.

- quantifying important parameters e.g. the hormone receptor status like (Rexhepaj et al. 2008) using the commercial Aperio image analysis system (Olson 2007). This automatic image analysis on digitized, selectively stained slices gives better thresholding for prognosis. In the case of the estrogen receptor the thresholding found in the automated analysis improved the therapy response prediction.

Incorporating clues from different biomedical domains like biomolecular findings offers an unrealizable variety and amount of clues. We concentrated on image analysis as it gives a direct link to the experience of the pathological experts. Furthermore we focused on properties of the patient's tumor that can be identified on images of the tissue magnification level. They give information about the embedding of the tumor into its surrounding as well as its supplementary situation. These items play an important role e.g. in the tumor's response to chemotherapy, cf. (Tannock 2001).

The tissue level clues were used to support the current standard diagnostics. Therefore the information on the different levels – patient, tissue, cell and biochemical properties – had to be integrated in *multi-layer models*, cf. (Klipp et al. 2009). Advanced biomedical research uses such representations of objects that are collections of complex and heterogeneously structured characteristics.

6.1.3 Validation of additionally extracted clues

The suggested potentially relevant clues for improving diagnosis and prognosis have to be validated over many patient cases. An evaluation of singular indications is of limited significance as biological processes are driven by a network of influence factors. Integrated multi-layer models are used as basis for an ecologically valid evaluation. As these complex constructs are not cognitively manageable by human experts anymore, approaches for automatic relevance analyses are needed. These approaches can provide an initial orientation for clinical trials.

6.2 Selection and mapping of medical challenges onto technical solutions

As the variety of possible clues for improving breast cancer diagnostics is incalculable, we had to select medical challenges and identify possibilities of mapping these challenges onto technical solutions.

6.2.1 Medically motivated suggestion of possibly relevant clues

Among the clues for diagnosis and prognosis improvement that are discussed in biomedical literature, we concentrated on clues that are identifiable on images of the tumor slices. Further focusing, only clues that are defined on the tissue level are considered for the mapping onto automatically extractable feature groups.

In agreement with the pathological experts, we focused on characteristics that are related to fields of tumor activity (Collins and Barker 2007): the distribution pattern of the tumor, hormone receptor expression, inflammation processes and defense of immune activity. This is in accordance with the influence factors on therapy response discussed in literature. An additional level of information is given by the heterogeneity of tumor properties in a patient's slice. Especially the distribution of the hormone receptors is important for the therapeutic success in hormone therapy, see e.g. (Horsfall et al. 1989).

6.2.2 Mapping the selected clues to machine extractable feature groups

The pathological expert knowledge is often implicit, experience-based, hard to structure and focused on single case decisions rather than on differences between several cases. The assumption that the pathological experts are able to explicitly annotate relevant clues or categories is unrealistic. We introduced a stepwise process of information extraction.

The process started with tissue characterizations based on machine learning algorithms. The results were mapped onto the images and color coded. The experts validated the relevance and appropriateness of the analytical results. This type of evaluation was in accordance with their typical form of case based reasoning. Critical annotations could be used in refinements of the technical analyses. The results of the tissue characterization served as a basis for more complex feature groups like the co-occurrence of tumor tissue with functional markers in a specific geometrical constellation.

Extraction steps, building upon each others results, are able to adequately bridge the *semantic gap* (cf. e.g. (Smeulders et al. 2000)) between the real biomedical clues and their representations for information processing. Together with the pathological experts we mapped the previously discussed clues for the tumors' fields of activity to sets of machine extractable feature groups. We focused on two main concepts: heterogeneity and distribution patterns. These two concepts can be analyzed under two perspectives: highlighting structural or functional aspects. This collection seemed to account for many of the diagnostic indications needed to differentiate the matched pairs given in our cohort. Table 6.1 shows a choice of these pairs and shows the aspects that were considered to be clues for the different follow-up status.

6.2.3 Evaluating the relevance of feature groups in multi-layer models

To yield a suitable, holistic representation of the patient's tumor, it is essential to evaluate the relevance of the chosen feature groups in every extraction step. To reduce the complexity of these integral analyses we first checked the spectrum of singular image and clinical feature groups with respect to their soundness and their discriminative power with respect to the follow-up status of the patients.

The relevance of the chosen feature groups for the overall representation has to be evaluated integrally. The mathematical integration requires an algorithmic approach that is able to deal with different forms of quantitative and qualitative data. The integration process is basically a kind of information aggregation according to

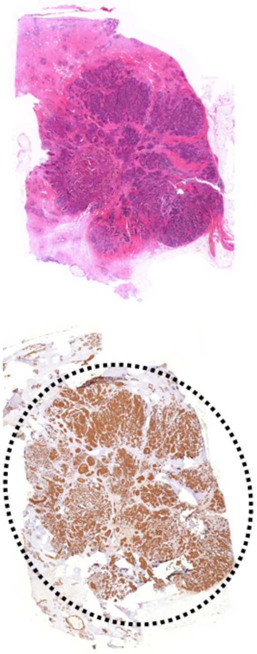
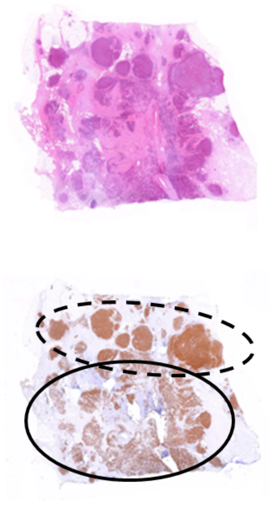
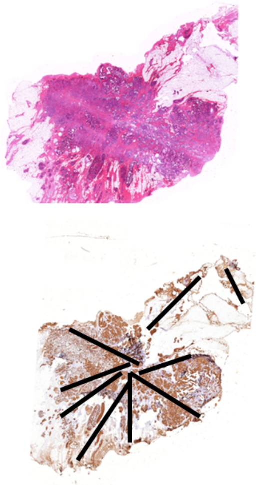
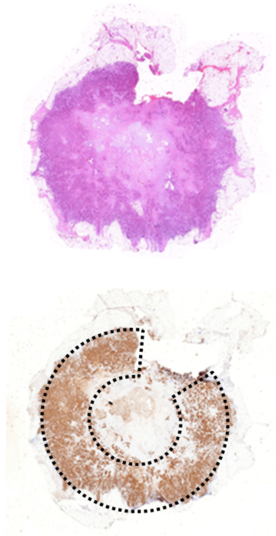
Image examples of good clinical course (healthy)	Image examples of bad clinical course (dead)	Potentially relevant clues
		*Heterogeneity characterization and quantification:* In the patient case with good clinical course, the dotted ellipse highlights a tumor area that is completely heterogeneous with a mix of dense spots and loose regions. In the patient example of a bad clinical course, within the dashed ellipse areas that are homogeneous dense are shown. This tumor in addition had relatively homogeneous loose regions marked with the solid ellipse.
		*Tumor distribution pattern characterization and quantification:* The example for the good clinical course shows a filamentous distribution pattern that is highlighted by the solid lines. In the patient case with a bad clinical course the tumor grew in a ring like structure marked by the dotted ring section

Table 6.1: *Image examples for matched pairs and potentially relevant clues*

various forms of dissimilarity. Some feature groups require the use of non Euclidean measures. The algorithms are described in chapter 5. The pattern related aggregation results and their potential relevance for the patients' prognosis are presented to the pathological experts in an interactive evaluation environment. Together with the basic data, these relevance results support evaluation, correction and domain specific insight.

6.3 Available data

The cohort that was available for our studies in the Exprimage project consisted of 93 patient cases. The Exprimage project was conceived as a retrospective study. That means that the investigated patients' tissue is older than five years[3]. The clinical data for the patients were collected in the clinical routine during diagnosis and therapy. The details about the disease course, especially whether

- the patient is still alive without disease or
- alive with disease (has a relapse) or
- dead from the disease,

are known. This follow-up status of the patients (alive, relapse, dead) was chosen as medically relevant label, which we refer to in classification. The distribution of the follow-up status in the data set was:

- Follow-up status one (alive): 50 patients
- Follow-up status two (relapse): 7 patients
- Follow-up status three (dead): 36 patients

We have a high imbalance between the classes. For all analyses the follow-up status two was neglected as there were not enough data samples.

6.3.1 Clinical data

The clinical data are a necessary part of the representation of a patient's situation. They reflect pertinent findings for every patient according to the current state of the art in diagnosis and prognosis. Most of the features collected in the clinical data are related to the cell detail level. They are a complementary part of a multi-layer model

[3] The time interval of five years is the pertinent clinical frame to evaluate the further perspective of the disease.

of the patient's situation with respect to information that is gained from the images on the tissue level.

Table 6.2 summarizes the general clinical data that were available for the patients in our cohort. For every feature a short explanation is given. We abbreviate the types of features by: nu for numerical, ca for categorical, no for nominal and bo for boolean features. For the categorical features the number of categories is given. A flag in the table indicates whether the corresponding feature was chosen for our analysis. The explanation for the choice is given in the following paragraphs. In table 6.3 we show the categorization of the clinical feature representing the size and kind of a tumor. This example shows the qualitative nature of the features. In that special case different aspects of tumor properties are mixed within one feature.

For some of the clinical features there were missing values in the documentation[4]. The selection of features for the analysis was done in accordance with the biomedical experts.

The first criterion was that the selected features should be equally available for patients with different outcomes. This was not the case for the time to follow-up and for the survival time. We did not consider these features. As the number of distant metastases was equal for most patients and missing for the others, we neglected this feature.

According to the pathological experts, the medical features have different levels of diagnostic reliability and impact on the disease progression. Currently the most important prognostic feature is the grading according to the Nottingham-Index (Galea et al. 1992). It is composed of factors like the creation of tubular structures, polymorphisms of nuclei and the mitosis rate. For the available cohort this feature had no good statistical correspondence with the follow-up status of the patients. Table 6.4 shows the contingency table between grading and follow-up status. We emphasize that the gradings are not used to predict special follow-ups. Especially a relapse is not prognosticated by grading two. Rather the tendency for survival or death is expressed by the grading. Grading one expresses a good prognosis whereas grading three is considered a bad prognosis. From grading two no reliable prognosis is given.

In the considered cohort 60% of the patients with grading one survived. Given grading three, 41% of the patients survived. In the patients with grading two 64% survived whereas 32% deceased. Only considering grading and follow-up status one

[4] The quality of clinical documentations is often insufficient. To use incompletely documented features, computational methods that can cope with missing values are indispensable. The methods introduced in this thesis are based on Vector Quantization. This can be extended to handling missing values. (Heskes 2001) e.g. introduced this extension by deriving an expectation maximization formulation of Vector Quantization. This extension was outside the scope of this thesis. We did not implement this potential but postponed it to future improvements of the system. Features with missing values were neglected in our analyses.

Medical feature	Explanation	Type of feature	Number of categories	Selected for analysis
Age at surgery	Age of the patient at time of the surgery	nu	-	yes
Grading	Grading of histo-pathological differentiation of tumor according to the Nottingham-Index on the basis of creation of tubular structures, polymorphism of nuclei and mitosis rate	ca, no	3	yes
Size and kind of tumor	Categorizing the size and kind of tumor	ca	6	yes
Number of affected lymph nodes	Categorizing the number and kind of affected lymph nodes	ca	5	yes
Number of distant metastases	Categorizing the number of distant metastases	ca, no	3	no
Invasion of lymphatic vessels	Categorizing the grade of lymphatic vessel invasion	ca, no	3	yes
Invasion of veins	Categorizing the grade of invasion of veins	ca, no	4	yes
Residual tumor	Categorizing the existence of residual tumor	ca, no	4	yes
Estrogen receptor	Immune-histochemical detection of ER (more than 10%)	ca, bo	2	yes
Progesterone receptor	Immune-histochemical detection of PR (more than 10%)	ca, bo	2	yes
Human epidermal growth factor receptor 2	Immune-histochemical detection of Her2	ca, no	4	yes
Time to Follow-up [years]	Time from OP to alive follow up	nu	-	no
Survival time [years]	Time from OP to death	nu	-	no

Table 6.2: *Overview of prediction features from clinical data*

Category	Subcategory	Explanation
TX		primary tumor can not be analyzed
T0		no clue for some primary tumor
Tis		carcinoma in situ
	Tis(DCIS)	ductal carcinoma in situ
	Tis(LCIS)	lobar Carcinoma in situ
	Tis(Paget)	M.Paget of mamilli without verifiable tumor
T1		tumor 2 cm or smaller in its peak open volume
	T1mic	micro invasion 0.1 cm or smaller in its peak open volume
	T1a	more than 0.1 cm, but not more than 0.5 cm in peak open volume
	T1b	more than 0.5 cm, but not more than 1 cm in peak open volume
	T1c	more than 1 cm, but not more than 2 cm in peak open volume
T2		tumor more than 2 cm, but not more than 5 cm in peak open volume
T3		tumor more than 5 cm in peak open volume
T4		tumor of every size with direct extent to the chest wall or skin, as far as described in T4a to T4d
	T4a	extent to the chest wall
	T4b	edema (including orange peel skin) or ulceration of the breast skin or satellite nodes of the skin at the same breast
	T4c	criteria of 4a and 4b together
	T4d	inflammatory carcinoma

Table 6.3: *Example for clinical categories representing combined tumor size and kind*

	Grading one	Grading two	Grading three
Follow-up status one	6	28	16
Follow-up status two	3	2	2
Follow-up status three	1	14	21

Table 6.4: *Contingency table for grading to follow-up status for the cohort given in Exprimage, explanation see text on page 120*

and three respectively yielded a Cohen's κ_c of 0.23 which can be interpreted as slight to fair agreement, see section 4.5.2 for details. For this configuration – neglecting the grading two, which we will later refer to as clinical configuration – we determined the recall of class one to be 27.3% and the precision of this class as 85.7%. The recall of the third class for this data set was 95.5% and the precision 56.8%. The overall recognition rate was 61.4%. As the grading is the standard diagnostic feature, it was integrated into the pattern analysis.

Summarizing, the clinical features we used for the analysis were:

- the age of the patient at the time of surgery
- the grading according to the Nottingham-Index
- the size and kind of tumor
- the number and kind of affected lymph nodes
- the invasion of lymphatic vessels
- the invasion of veins
- the presence of residual tumor
- the expression of the estrogen receptor
- the expression of the progesterone receptor
- the expression of the human epidermal growth factor receptor 2.

Neglected features were:

- the time to follow-up (only available for alive and relapse patients)
- the survival time (only available for dead patients)
- the number and kind of distant metastases (equal value in all featured patients).

There was no information available on the therapeutic course of the patients. This limits the medical expressiveness of the analysis model. The relevance of the feature groups with respect to therapy response could not be examined. In the medical perspective we could only generate tentative hypotheses that have to be evaluated by pathologists. In order to become medically relevant, the results have to be systematically analyzed in clinical trials. Our goal was to develop a computer based analysis system that is able to integrate mixed patient data and that builds an individual profile for diagnosis and therapy out of this integrated data. These profiles have the potential of pointing towards hypotheses for medical relations.

6.3.2 Image data

For every patient we had two kinds of stained tissue slice images: structural and functional stains. The starting point for image analysis in Exprimage were raw digitized microscopic images of stained tissue. These images showed the whole object slide. The original images were given to correspond to a 40 or 20 times optical magnification. We standardized the magnification of all images to correspond to a 20 times optical resolution. One pixel in the image thereby was equivalent to a physical size of 5.2 µm × 5.2 µm. We reduced the processed area of the probes to the actual tissue sample by hand. Thereby surrounding noise was excluded. After this procedure the images sizes varied according to the tumor size. Our smallest sample comprised 1398 × 1036 pixel which corresponds to an area of about 7.3mm × 5.4mm, whereas the biggest sample size was 2138 × 2971 pixel corresponding to an area of about 11.1mm × 15.4mm.

Structural stains

The structural marker stains highlight principle cellular modules that built the cells' structure. In the Exprimage project, we considered the following structural stains:

HE The commonly used histological HE (hematoxylin and eosin) stain roughly speaking stains cell nuclei blue and some cell plasma proteins in various shades of red.

VIM The mesenchymal marker Vimentin highlights connecting tissue in shades of brown. It has a blue counter stain.

AE1AE3 The pan-cytoceratin stain AE1AE marks epithelial tissue that is lining cavities and forming glands. The main stain is brown, the counter stain is blue.

These structural stains were available for all patients. Within these different stains the HE stain marks a broader spectrum of structural information than all other stains used in Exprimage. Figure 6.4 shows the original digitized images of the structural stains for one patient.

Functional stains

The functional markers stain structures that are associated to special functions within the tissue. For all patients we analyzed the following functional stains:

CD45 The CD45 stain highlights leukocytes and is used to detect inflammatory processes in the tumor slice.

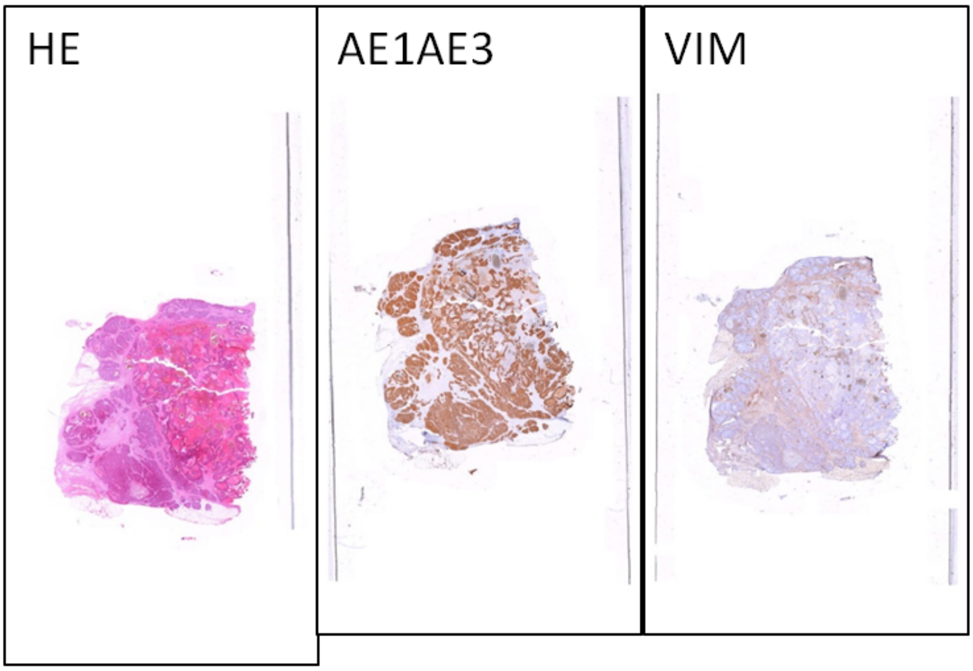


Figure 6.4: *Example of the structural stains for one patient as they are available as digitized images.*

ER Using the ER marker cells that express the receptor for estrogen are identified.

PR The PR stain highlights cells expressing the progesterone receptor.

The inflammatory processes are known to influence the therapy response of tumors, e.g. (Grivennikov et al. 2010). Hormone therapy is suggested based on the analysis of the hormone receptor status of the patients. Figure 6.5 shows the original digitized images of the functional stains for one patient.

6.4 Information processing for setting up multi-layer models

To build a suitable cognitive support system for breast cancer research in Exprimage our working group developed three information processing complexes:

Image processing and feature calculation This first complex of processing steps comprised the network of preparation and extraction steps for groups of image

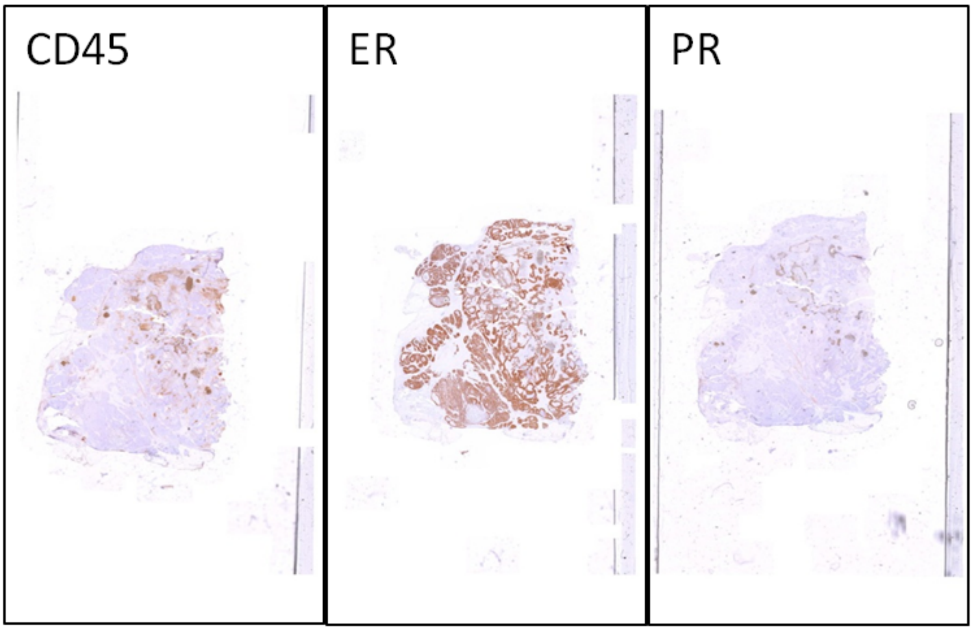


Figure 6.5: *Example of the functional stains for one patient as they are available as digitized images.*

features that represent the potentially relevant medical clues of a patient's tumor situation.

Isolated relevance analysis The extracted feature groups were, in a second complex of information processing steps, evaluated according to their relevance for the description of a patient's situation with respect to prognosis prediction. Therefore the single feature groups were evaluated by the human expert as well as in automatic evaluation procedures to get a selection of possibly relevant candidate feature groups for a multi-layer model.

Integrated relevance analysis The selection of feature groups identified in the second step was analyzed integrally according to the relevance of the feature groups within the context of the multi-layer model. Together with the basic data and the interim results, the integral analyses results were displayed to the pathological expert. The objective was to allow the evaluation of the results and induce possible insights.

The image processing for the feature extraction is introduced in section 6.5. As it is not in the focus of this thesis we will not give a detailed scientific comparison to other possibilities to solve the image processing problems. The discussion of the feature groups is necessary to understand their heterogeneity and their embedding into the overall patient's tumor representation.

In an ideal situation all feature groups would be analyzed integrally for their relevance in the multi-layer model. This yields a complex model with a high number of free variables. There was a limited amount of patient data samples available in our project for the adaptation of the free variables. It was possible that this amount of data was too small to reliably estimate the free variables from it. We interposed the isolated relevance analysis to get a reduced set of candidate feature groups.

In section 7.2 the isolated relevance analysis for the selection of single feature groups as candidates for a holistic multi-layer model is discussed. The focus of this thesis is on the integrated relevance analysis and its evaluation. We will describe it section 7.3.

6.5 Image processing and feature calculation

Together with the pathologists we conducted a requirement analysis, identifying which medical clues have to be mapped to automatically calculable feature groups to yield suitable representations of tumor situations. As mentioned before, we selected two main concepts of tumor description – heterogeneity and distribution patterns – and analyzed them under structural or functional perspectives. Table 6.5 shows these tumor properties together with the conceptual description of a set of automatically computable feature groups that we extracted for their representation.

To map these potentially relevant medical clues to machine extractable feature groups, a network of image and information processing steps was needed that we discuss in the following sections. We show a coarse schematic representation of the network of image processing steps in figure 6.6. For sake of clarity we dropped some details of interconnection between the single processing steps as well as the detailed input and output characterization. They will be discussed for the single processing steps in the following sections.

The image processing and feature calculation in the Exprimage project was mainly implemented and conducted by two students. The subject orientation and technical supervision of the students was given by the author. The students work resulted in two theses: Elionora Khabirova's master thesis on "Image processing descriptors for inner tumor growth patterns" (Khabirova 2011) and Jan Bornemeier's Diplomarbeit on "Development of descriptors for the determination of spatial distribution patterns in histopathological tissue slides of the mammary carcinoma" (Bornemeier 2011). We

Potentially medically relevant tumor property	Conceptual description of automatically computable feature groups
Structural heterogeneity	Differentiation of inner tumor growth structures
Functional heterogeneity	Co-occurrence analysis for functional marker expression
Structural tumor distribution patterns	Graph and morphometry based analysis of tumor distribution patterns
Functional tumor distribution patterns	Relation based analysis of tumor distribution patterns in correlation to functional marker expression

Table 6.5: *Selected medical tumor describing aspects and their mapping to conceptually described automatically computable features*

will base our descriptions of the image processing network in the following sections on these theses.

Our report starts with the tissue type differentiation that in figure 6.6 is shown at the center because it was the basis for all other feature extraction tasks.

6.5.1 Basic recognition task: tissue type differentiation

To characterize the general tissue situation on a patient's probe, a discrimination of tissue types like tumor or healthy tissue was needed. In this section we discuss an intuitive evaluation possibility of tissue type characterizations. We introduce our supervised approach for tissue type discrimination and explain why this approach failed in evaluation procedures. We further focus on the finally applied procedure – a clustering approach. This was chosen for the working system as it proved to be reliable in evaluation. The fourth part of this section is concerned with the question of triggering domain knowledge by the evaluation of the tissue type characterization. At least we will shortly sketch the extraction of tissue regions from the tissue type characterization.

Evaluation of tissue type differentiation by pseudo-colored images (PCIs)

VQ based algorithms are frequently used for segmenting images into regions of similar properties, e.g. similar texture or intensity. Often a feature representation is calculated for every single pixel in the images. These pixel features are clustered or

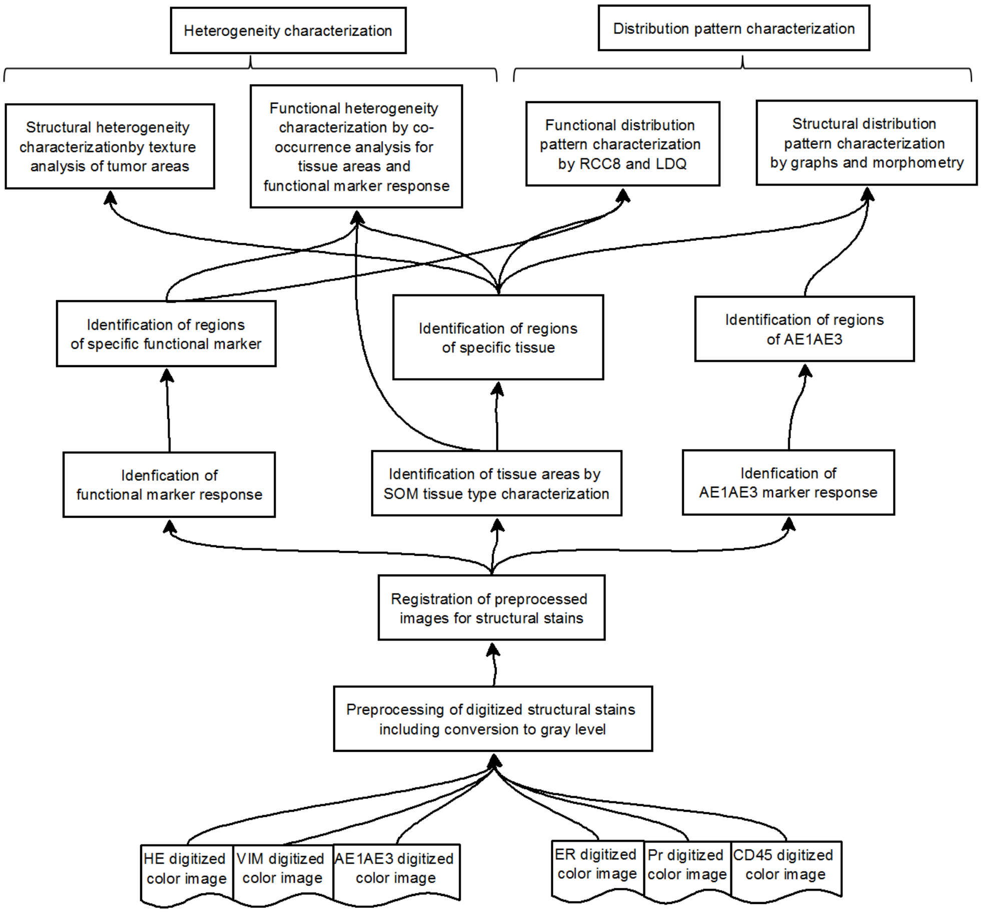

Heterogeneity characterization
Distribution pattern characterization
Structural heterogeneity characterizationby texture analysis of tumor areas
Functional heterogeneity characterization by co-occurrence analysis for tissue areas and functional marker response
Functional distribution pattern characterization by RCC8 and LDQ
Structural distribution pattern characterization by graphs and morphometry
Identification of regions of specific functional marker
Identification of regions of specific tissue
Identification of regions of AE1AE3
Idenfication of functional marker response
Identification of tissue areas by SOM tissue type characterization
Idenfication of AE1AE3 marker response
Registration of preprocessed images for structural stains
Preprocessing of digitized structural stains including conversion to gray level
HE digitized color image
VIM digitized color image
AE1AE3 digitized color image
ER digitized color image
Pr digitized color image
CD45 digitized color image

Figure 6.6: *Schematic representation of the processing steps in image analysis for the feature group extraction in the Exprimage project.*

classified. New images are created with the same dimensions as the input images. Every pixel of these evaluation images is colored according to the prototype index or classification result of the corresponding pixel feature representation. This process is also called pseudo-coloring.

For evaluating a *clustering* the prototype indices are assigned to colors. For the evaluation of a LVQ based *classification* the assigned classes are mapped to colors. The choice of the color set can reflect conceptual aspects. It has to be done carefully. From the psychological point of view misleading, suggestive color representations are possible. (Flatla and Gutwin 2010) introduced in 2010 a possibility for automatic individual adjustment of colors to improve the interpretability of information visualization.

We mapped the classes or the prototype indices respectively to colors according to the default colormap in MATLAB. The resulting pseudo-colored evaluation images were overlaid to the corresponding input images and by adjusting the transparency the images were compared according to their inherent structures of the regions. Figure 6.7 shows an example of the evaluation of a clustering conducted on image pixel representations using texture features. The pseudo-colored evaluation image was overlaid to the histological input image. The figure shows a variation of the transparency from 100% to 0% transparency of the evaluation image.

Figure 6.8 shows an example of an image pixel classification evaluation using the same method. The pseudo-colored evaluation image was overlaid to the histological input image with different transparency settings from 100% to 0% transparency. The legend of the class to color mapping is given on the right. This visualization allowed an easy evaluation by the domain expert.

If the result visualization differs from the expectation, this can be due to several causes: failure in learning, inadequate visualization method or wrong expectation. Every potential explanation has to be checked with the domain experts.

Supervised tissue differentiation approach

To establish a tissue differentiation we needed examples for suitable classes as well as for features that are able to discriminate these classes. We asked the pathological experts to annotate pertinent regions on a single HE stain. We also asked for potentially computer graspable image features like texture or distribution of edges that appropriately distinguish the tissue regions that were marked. The criteria that the experts highlighted were the graininess of the tissue as well as the intensity of the stain, i.e. the texture of the tissue. We chose the simplest way of representing the texture for each pixel – calculating the mean and standard deviation of the gray values within an area of 19×19 pixel around each pixel. This area corresponds to a sample area of approximately 0.1 mm $\times$ 0.1 mm and is adjusted to the size of the expected differentiating textures. For the smallest image that makes approximately 1.5 million feature samples with a feature dimension of two each. For the biggest image there are over 6 million samples.

In the annotation of example areas for classes the experts were free to choose the number and type of the tissue classes. We used these annotations to train a Linear Discriminant Analysis (LDA) using squared Euclidean distances[5]. As the annotations were not done with pixel accuracy we did not evaluate the LDA performance with

[5] For details about the LDA we refer to section 4.3 of (Hastie, T. et al. 2003).

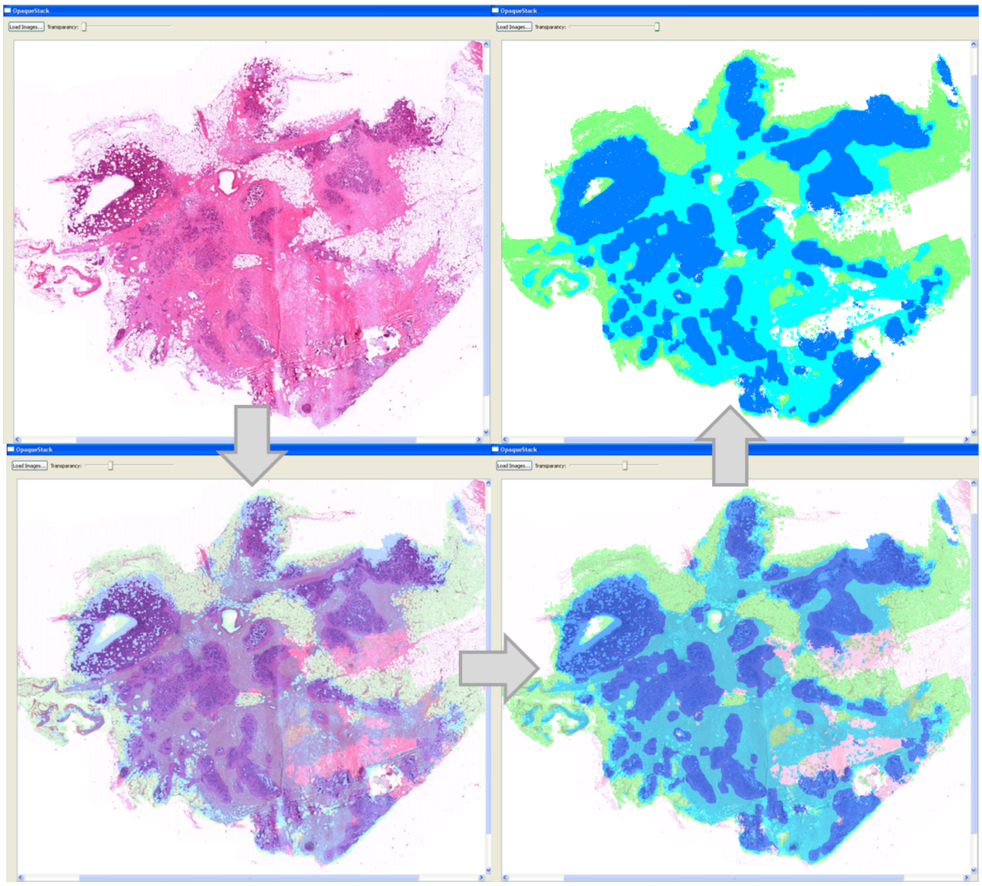


Figure 6.7: *Example for the evaluation of a clustering of image regions, in this case tissues, using a pseudo-colored image (PCI). The pseudo-colored evaluation image was overlaid to the image using different transparency settings.*

an accuracy measure. We rather used a visual evaluation by the pathological experts. The classes annotated by the pathologists were mapped to artificial colors for an evaluation using pseudo-colored images as described in the last section. Figure 6.9 shows the evaluation of a classification result, overlaying the pseudo-colored image to the HE image with a transparency of 50%.

The experts confirmed the correctness of the results for the annotated image. The generalization using the learned parameters on other patients' images was poor, as highlighted in figure 6.10. The rectangles mark examples of tumor tissue that was wrongly classified as healthy tissue. This observation is in accordance with the

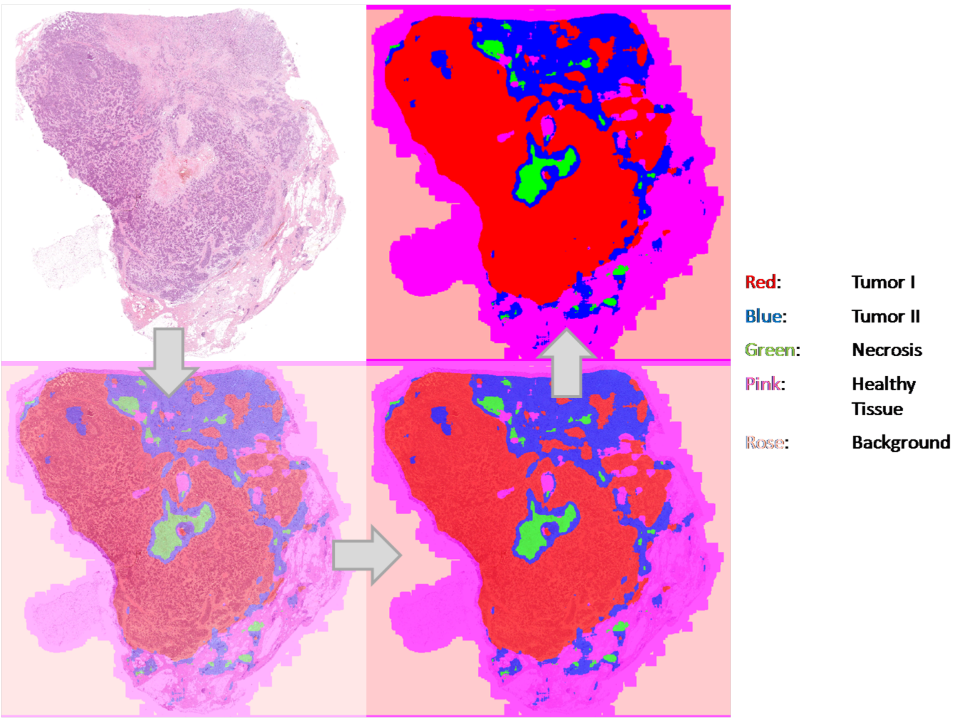


Figure 6.8: *Example of an evaluation process for a classification of image regions, in this case tissues. The pseudo-colored evaluation image is overlaid over the microscopic image of the tissue using different transparency settings.*

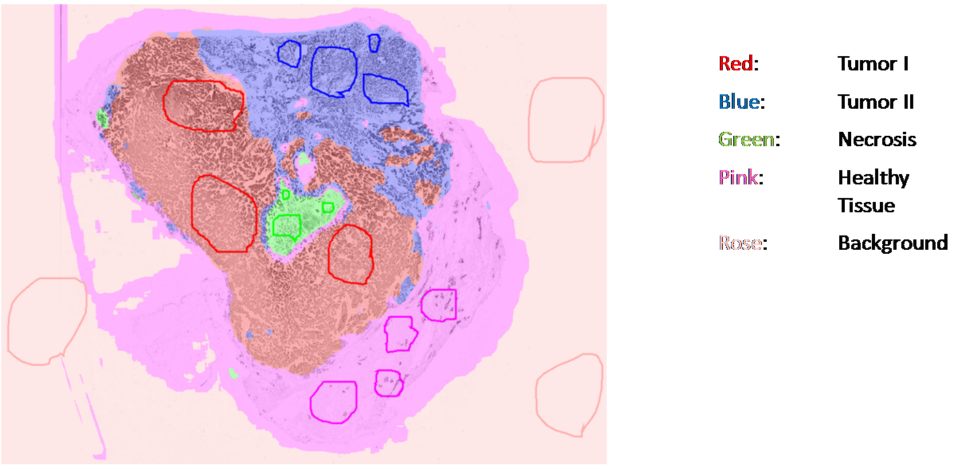


Figure 6.9: *Result of a LDA on a single patient's image learned on expert annotations given on this HE stain.*

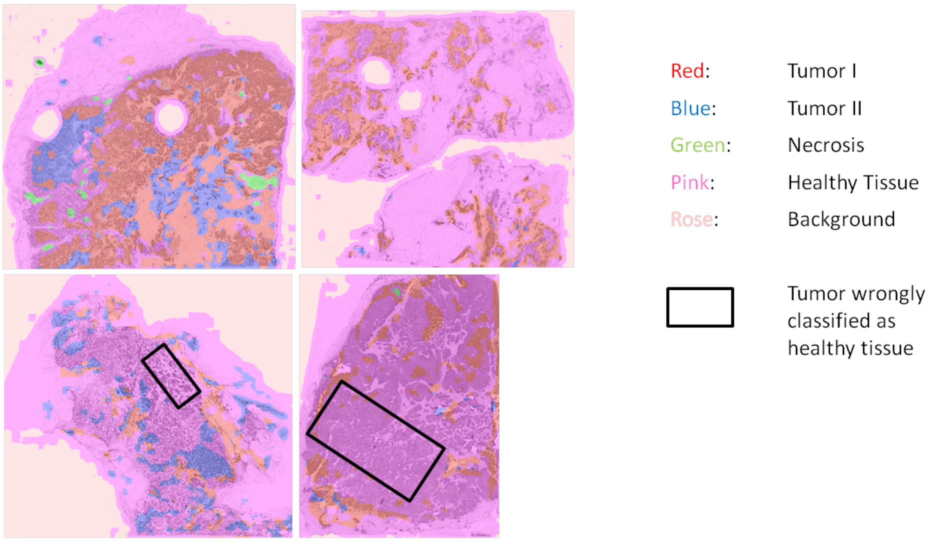


Figure 6.10: *Generalization result of a LDA that was trained on one single patient's image, see figure 6.9. In this example the LDA is applied to four other patients' images with unsatisfying results.*

expectation that the tissue texture in one single patient is not representative for the tissue texture in other patients. Consistent annotations over a larger number of cases are needed to establish a good generalization ability for the LDA. The annotations are necessary to cover the variety of manifestations of different tissue classes.

We asked one expert to apply the annotation scheme used for the first patient image to ten patient images. When we analyzed this annotation procedure it was obvious that it was very difficult for the expert to define consistent classes over more than one patient. He often wanted to introduce specialized classes for single patients. The degree of detail in the class definition was not clear and he asked for the possibility of a hierarchical annotation scheme. These observations led us to the assumption that the annotations were not reliable in terms of inter-observer reliability.

We conducted a reliability test: We asked one expert to mark regions of similar tissue in different probes and name them – as described above. Then we presented the marked regions isolated from the complete probe image to another expert and asked her to choose one of the following options:

- Name the given tissue region using one of the predefined names.
- Mark the tissue region as ambiguous.

	no. of cases			percentage (%)	
class	validated	conflict	consistent	conflict	consistent
Fat tissue	53	0	53	0.0	100.0
Necrosis	45	20	25	44.4	55.6
DCIS	9	2	7	22.2	77.8
Tumor-solid	41	30	11	73.2	26.8
Tumor stroma	93	17	76	18.3	81.7
Tumor parenchyma	84	48	36	57.1	42.9
Inflammation-normal	15	10	5	66.7	33.3
Adenoid tissue	4	2	2	50.0	50.0
Vessels	72	9	63	12.5	87.5
Tumor-dissociated	15	6	9	40.0	60.0
Tumor-tubular	21	20	1	95.2	4.8
Inflammatory tumor	39	19	20	48.7	51.3
Normal mammary parenchyma	26	10	16	38.5	61.5

Table 6.6: *Class-wise statistic of an inter-observer reliability test for the annotations given by two experts over* 10 *patient images*

For some classes the annotations were often marked as ambiguous. This was especially the case for: solid tumor (22%), tubular tumor (19%) and inflammatory tumor (18%). In contrast to this, the annotations of the classes DCIS, adenoid tissue and normal mammary parenchyma were never marked as ambiguous.

For the inter-coder reliability statistics we counted the number of consistent and conflicting codings. Regions marked as ambiguous were not considered for this statistic. The results for the reliabilities in the single classes are shown in table 6.6. Fat/healthy tissue was consistently classified in 100% of the cases. In contrast to this, normal tumor parenchyma was only consistently marked in 36 out of 84 cases, yielding a reliability of less than 50%. Averaged over all annotations the consistency percentage was 62.7%. The overall reliability is shown in table 6.7.

The low agreement rate between the experts' codings had two apparent reasons:

- the large number of coded classes and
- the difficulty for the human experts to overlook ten cases.

Pathological experts are not used to integrative thinking over many cases as their analysis usually is concentrated on a single case. We assumed that suggestions of general tissue types which were created via unsupervised learning could help the pathological experts in the identification and naming of common tissue types for

total number of validation cases	517
total number of consistent cases	324
total number of conflict cases	193
total conflict percentage	37.3 %
total consistency percentage	62.7 %

Table 6.7: *Overall statistic of an inter-observer reliability test for the annotations given by two experts over* 10 *patient images*

many cases. In a modified approach we grouped the tissue using an unsupervised clustering procedure. The resulting groups were mapped onto artificial color codings of the probes. These images were presented to the experts who post-labeled the various tissue regions. It turned out that this procedure was reliable and the results were acceptable for the experts.

In this computer based generation of annotations we used the advantage that for the computer based analyses the information of different structural stains can be integrated. We assumed that the integrated structural tissue stains in a registered stack deliver more stable features for discrimination than the HE stain alone. For this integrated analysis the stain images had to be aligned or registered.

Registration of stain image

The conceptual idea of the alignment or registration process is to find a transformation from the coordinate system of one image to the coordinate system of the other image, such that the information at a certain point in one image is most similar to the information at the corresponding point in the other image. That means, that after the transformation the informative structures in the images overlap as exactly as possible.

For the tissue type differentiation we registered the structural stains into a common stack of images. In the following, we introduce the preprocessing and the registration process that we used in Exprimage.

Preprocessing As noise or other perturbing data on the images change the gray or color value information, they can cause suboptimal registration results. We preprocessed the single images, preserving the information that is relevant for the registration process and removing perturbing data.

The stains varied significantly in their staining intensity as can be seen in the examples given in figure 6.4. The information of this variance often disturbs the registration process. Furthermore the specific color information in the images is most

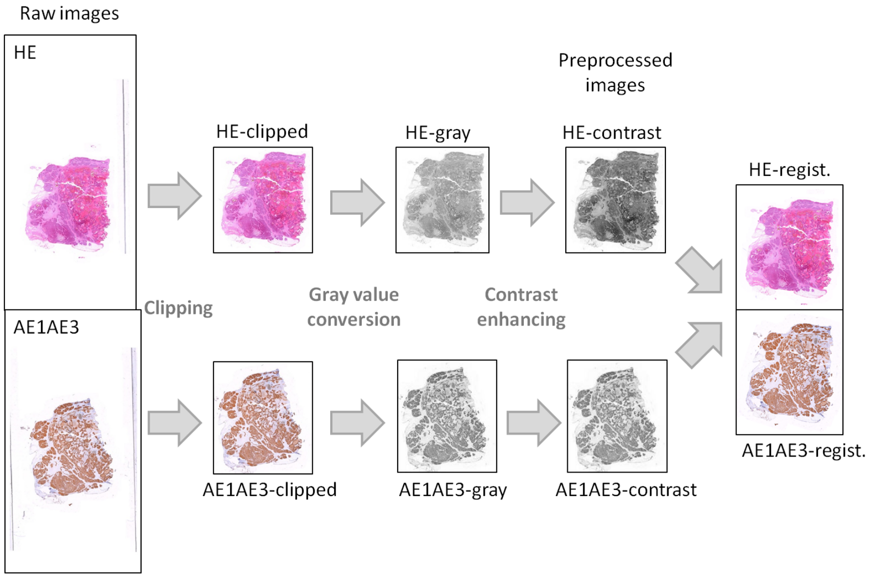


Figure 6.11: *Summary of preprocessing steps for registration displayed on an example of raw staining images to be aligned.*

often irrelevant for the registration process. The conversion of the color images to gray-scale images was a useful reduction of variance in this processing step[6]. In contrast to the intensity and color information, the structural information of the single images was used to full extent. For this purpose the contrast of the generated gray-scale image was enhanced by adjusting the images' histograms.

All preprocessing steps for the registration of the images are summarized in figure 6.11. After the preprocessing a selection of registration relevant information is preserved.

Registration process We used the enhanced gray values gained in the preprocessing to represent the information that had to be aligned in the images. As the different markers highlight different aspects of the tissue, the gray values of similar tissue were not be the same in different marker images. Thus it was not appropriate to simply compare the gray values and use their difference as a measure of correspondence in information when evaluating the overlapping quality.

Instead we used a measure evaluating the correlation between the aligned gray values in the different images. This quantitative measure is called *mutual information*.

[6] We implemented this by using the *"rgb2gray"* function from MATLAB's Image Processing Toolbox (Mathworks n.d.b).

It expresses how much information of one image can be gained from the other one, cf. (Pluim et al. 2003). If the images are optimally aligned, they contain the maximum information about each other. As the HE stain contained most structural information it was used as reference stain to which all other stains were aligned.

Additionally to the measure that was used to evaluate the overlapping quality, a procedure to optimize this measure during registration was introduced. To determine a suitable approach for the optimization of the mutual information, it was necessary to know which kinds of transformations was possible between the images that had to be aligned. Often the possible transformations are differentiated into rigid, affine and non-affine transformations[7]. Rigid transformations allow translations and rotations. In affine transformations additionally anisotropic scaling and skews are allowed. They can be represented in a matrix form and have the characteristic that under their application parallel lines stay parallel. In addition to the mentioned transformations, non-affine transformations allow any degree of freedom in the mapping.

In Exprimage, these non-affine transformations were necessary, as the tissue slice was deformable before it was fixed to the object holder. Non-affine transformations bear the risk that potentially any point in one image can be mapped to any point in the other image. To reduce the degree of freedom for the non-affine transformation, in Exprimage, it would have been necessary to model the tissue properties. This would have been error prone.

As the tumor slices experienced only slight and local non-affine deformations, aligning them using a rigid transformation was legitimate. The additional slight non-affine deformations were handled in a second process step using a locally non-affine registration. This gives the following two-stepped registration process:

1. Preregistration: rigid registration to adjust the coarse region of interest

2. Fine registration: non-affine registration to compensate small non-affine transformations.

The process was implemented as follows: For the rigid registration the preprocessed scene image was rotated and shifted against the preprocessed reference image within a certain range. In every position the mutual information of the transformed images was calculated. This was optimized using the Amoeba algorithm, that is also known as Nelder-Mead algorithm (Press et al. 2007).

For the fine registration, every preregistered scene image was divided into four non overlapping tiles. Then every tile was slided on the model image in a given range using again the Amoeba algorithm (Press et al. 2007) to find an optimal mapping. To

[7] (Hill et al. 2001) give a comprehensive survey on the basics of registration, possible transformations, registration techniques and their applications in medicine.

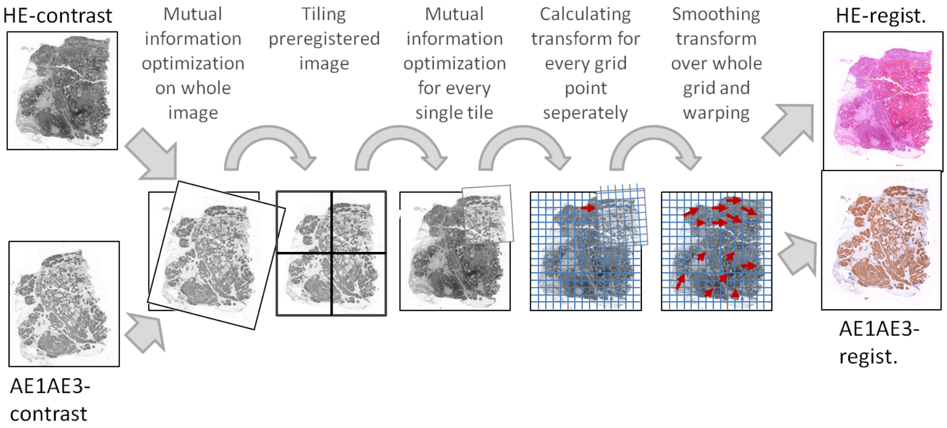

HE-contrast
Mutual information optimization on whole image
Tiling preregistered image
Mutual information optimization for every single tile
Calculating transform for every grid point seperately
Smoothing transform over whole grid and warping
HE-regist.
AE1AE3-contrast
AE1AE3-regist.

Figure 6.12: *Summary of processing steps for registration displayed on an example of preprocessed staining images to be aligned.*

integrate the different results for the tiles, we projected a grid of size 16×16 onto the reference image. For every grid point its transformation vector according to the best correlation transformation of the tiles was calculated.

To smooth the resulting vector field, we applied the interpolation approach introduced by (Fornefett et al. 1999). In this approach the interpolation transformation function was approximated by a sum of polynomials plus the sum of Radial Basis functions (RBFs). Because of the rigid preregistration we did not need the polynomials for the approximation of general shifts or rotations. The RBFs were implemented by Wendland splines, see (Wendland 1995), as suggested by (Fornefett et al. 1999). The interpolation transformation function as sum of Wendland splines was applied to all pixel in the scene image to get the registered scene image. Figure 6.12 summarizes all steps in the registration process and applies them to two example images. The result of the registration process was a stack of registered structural tissue slice images.

The preprocessing introduced before was not capable to remove all data that can disturb the registration procedure. We faced artifacts of missing tissue regions as the slices are spoiled during cooking in the staining process. Figure 6.13 shows on the left side an example of a partially disrupted HE slice image marked with an ellipse. As the number of prepared tissue slices with all background information available was small, even this slice was valuable and had to be registered and used in further analyses. For the spoiled areas the registration process could not find a valid correspondence when comparing the corresponding AE1AE3 slice image at the right side of figure 6.13. This nonexistent correspondence caused misalignments. To

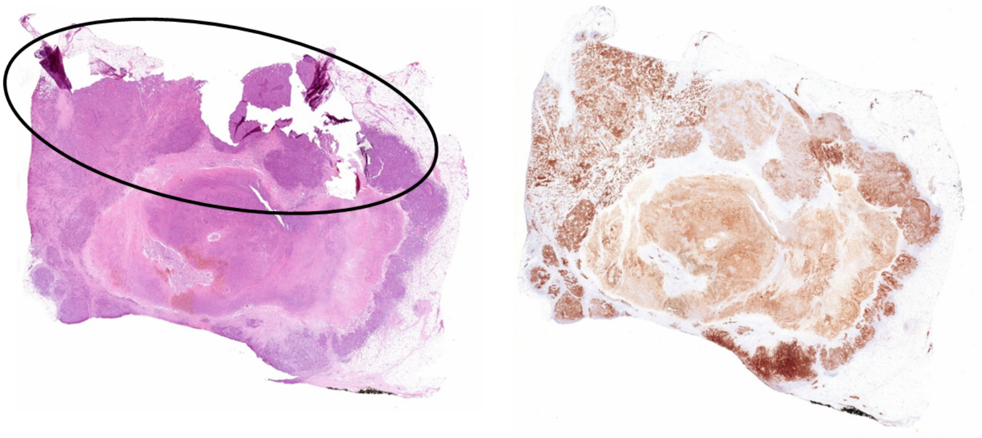


Figure 6.13: *Example of a spoiled HE stain slice image to which an AE1AE3 stain slice image should be registered*

overcome this problem, we modified the process of mutual information calculation such that missing tissue areas were identified and ignored.

Evaluation of registration The grain-size on which combined features could be reliably calculated depends on the registration quality and reliability that was achieved. Registration errors would be passed through the whole processing chain. Before proceeding we evaluated the quality of the registration. There were different factors that had a negative influence on the registration quality, e.g.:

1. The 3D tumor was cut into 2D slices. The stained slices we used in the analyses had different distances between each other. Slices with a higher distance to one another in the basic tumor showed higher differences in the tissue distribution.

2. A slice was partially spoiled as shown in figure 6.13.

3. There were recording artifacts like inhomogeneous lighting, shadows or staining artifacts that were not correctable by the automatic preprocessing.

The last two influence factors occurred seldom. In contrast to this, the first influence factor concerned every patient's probe. In registration the underlying assumption is that the considered images comprise more or less the same information. Strictly speaking this was not the case in the Exprimage project's images as on different images there were different tissue slices with different stainings. Distant slices showed bigger differences in their informative structures than consecutive ones.

In principle there are two ways of quality assurance (QA): automatic processing or manual evaluation. The mutual information of two slice images is itself a measure of quality. It is a technical term that solely builds on local gray-value correspondences of the stain images. As it is not normalized it is not pertinent for the comparison over different image pairs. We ran human evaluations of the stack of registered images.

To yield a reliable QA procedure, it was necessary to fix relevant criteria for the evaluation of the registration quality. The relevance of the criteria in our case was determined by the processing steps following the registration. A suitable registration was fine-grained enough to support further analyses of structures with pathological relevance. We divided the quality judgment into four categories with an associated measure, representing the pertinence of the registered images:

-1 – Unusable bad quality in registration and slices' preparation

0 – Critical bad slices' preparation and critical registration quality (as good as possible)

1 – Good good quality in registration and slices' preparation

2 – Perfect very good quality in registration and slices' preparation

We built an evaluation tool allowing an overlay of two images with adjustable transparency. Figure 6.14 shows a registered sample in this environment at different settings for the overlay transparency. The transparency slider is highlighted by the ellipses. To clarify the idea of registration quality, areas of bad registration quality are marked using solid rectangles whereas regions of good registration quality are highlighted using dotted rectangles.

Averaging the categorical measure over all 93 patients' slices in all structural stains, the overall registration quality was about 0.87 with a variance of 0.37. In most cases a sufficiently good registration was possible. Different stains contributed to this average with different precision averages. The average quality of registering the AE1AE3 stain to the HE stain was at a value of 1.01, whereas 0.72 was the average quality value of registering the VIM stain to the HE stain. This was expected as the VIM stain highlights less structural information than the AE1AE3 stain.

Unsupervised tissue differentiation approach

The clustering process was based on the images of the HE, VIM and AE1AE3 stain that were preprocessed as described in section 6.5.1 and registered as introduced in section 6.5.1. We used the simple texture representation by the mean and the standard deviation of the intensity that were calculated for every pixel in the images in a 19×19 pixel area around the pixel. Figure 6.15 depicts the process of feature

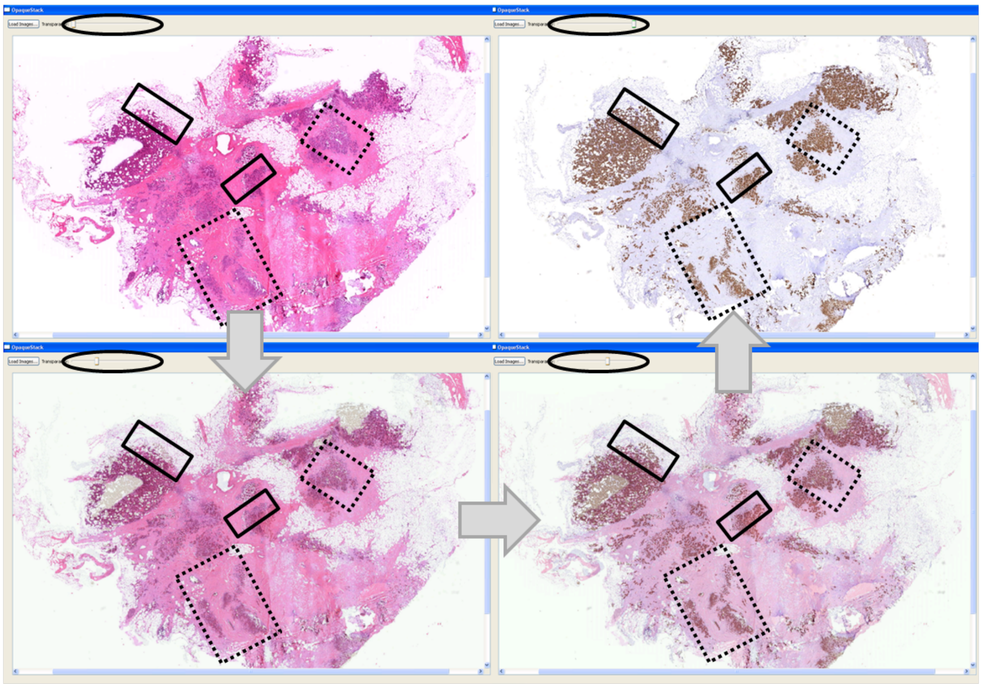


Figure 6.14: *Example of the evaluation of a HE and a AE1AE3 stain using the overlay evaluation tool.*

extraction schematically. This yielded between 1.5 and 6 million feature samples per image.

Together with the pathological experts, we selected ten patients to cover a large variety of tissue types. The feature data for this selection – approximately 37.5 million samples – were analyzed using the SOM toolbox for Matlab (Alhoniemi et al. n.d.). We trained a Self Organizing Map with a hexagonal grid in sheet form with 106×47 units. For details on the Self Organizing Map we refer to (Kohonen et al. 2001).

Together with the pathological experts, we decided to allow three clusters – and thus three tissue types. We expected this choice to yield enough information and context for further processing and to lead to a stable tissue recognition. This approach was motivated by the multi-layer processing in the human visual cortex, where several simple but stable processing steps realize a complex information processing in their suitable collaboration. Correspondingly we clustered the trained SOM using the k-means algorithm with a k of 3. For evaluation we used pseudo-colored images. Figure 6.16 shows the pseudo-colored evaluation image of the resulting clustering for the image example that was used to depict the results for classification in figure 6.9

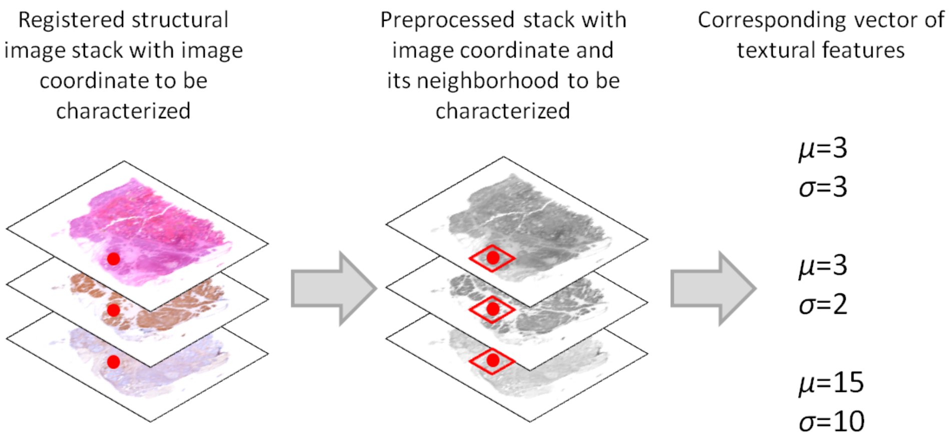

Registered structural image stack with image coordinate to be characterized
Preprocessed stack with image coordinate and its neighborhood to be characterized
Corresponding vector of textural features
μ=3
σ=3
μ=3
σ=2
μ=15
σ=10

Figure 6.15: *Schematic depiction of the feature extraction process for tissue type characterization.*

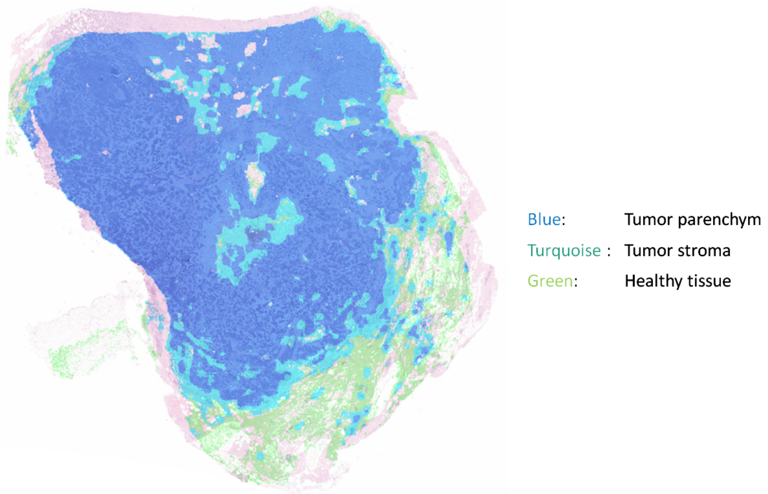
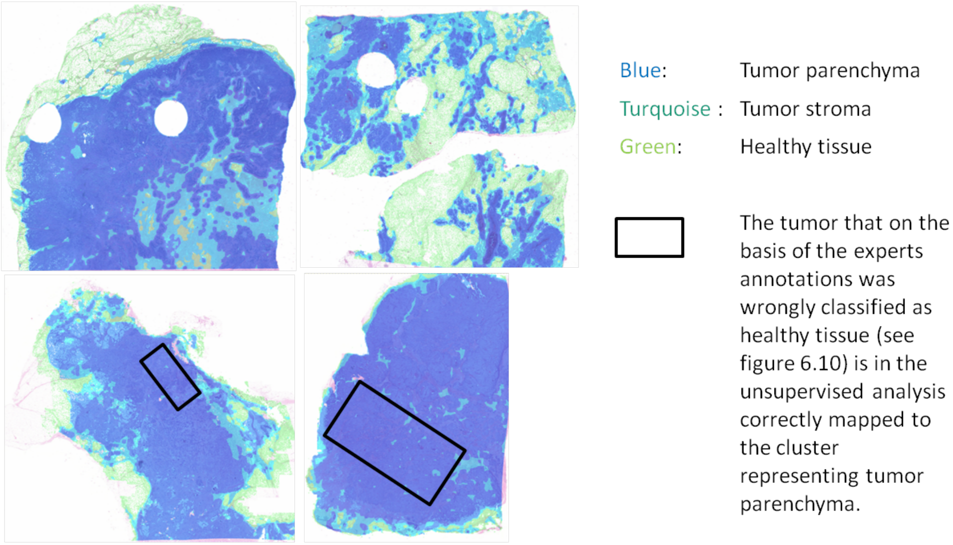


Figure 6.16: *Result of a tissue clustering trained over ten patient cases. In this example the clustering is applied to the patient case that was used in the first classification test, cf. figure 6.9.*

on page 132. The best clustering – according to the domain experts' evaluation – was analyzed in a post-labeling procedure, cf. section 2.6. It showed three tissue types, that the pathological experts identified to be:

- Tumor tissue, also called tumor parenchyma: the actual, vital part of the tumor, represented by the blue cluster



The tumor that on the basis of the experts annotations was wrongly classified as healthy tissue (see figure 6.10) is in the unsupervised analysis correctly mapped to the cluster representing tumor parenchyma.

Figure 6.17: *Generalization result of the clustering trained over ten patient cases. We applied this clustering to the four patient cases that were used in the generalization test for the classification in figure 6.10.*

- Tumor supporting tissue, also called tumor stroma: supplying the tumor with nutrients, represented by the turquoise cluster

- Healthy tissue: often fatty tissue that gets invaded by the tumor and by the tumor supporting tissue, represented by the green cluster

Figure 6.17 shows the distribution of these tissue types in the four patient cases that were used in the generalization test of the classification in figure 6.10 on page 133. In figure 6.17 for comparison we highlight the regions that were wrongly classified in the generalization test of the supervised approach based on the experts' annotations. These regions were consistently identified by the clustering.

The unsupervised tissue type differentiation procedure gave a coarse-grained and stable pixel-wise representation of the structural tissue situation of the tumor slice. We refer to the pseudo-colored images used in the evaluation as *SOM images*. They were used as basis for more complex features. For the calculation of complex features that are cognitively more intuitive for the pathological experts, regions of single tissue types and their contact lines had to be identified. These relevant regions were handled as objects for which we could calculate pertinent properties, e.g. their morphology.

Triggering specific domain knowledge by pseudo-colored images

In the tissue characterization evaluation the experts decided whether the results seemed plausible by going through the clustering results of several images. If that was the case in an overview, the experts searched for affirmations of common sense domain knowledge. If some results seemed to contradict these expectations there was a range of possible causes reaching from algorithm failure to new insight.

In our application the experts expected a tumor region within the image to be segmented into different subregions but the clustering result united these subregions. They confirmed their expectation in the overlay of the SOM images to the HE stains. When evaluating the overlay with the cytokeratin AE1AE3, the experts saw their fault in the expectation as the delineation of the tumor was more precise there.

In a situation where the result of the learning is not in accordance with the experts' expectations, the experts have to be encouraged to reassure its correctness. Perhaps the algorithm found a weak spot of the experts' normal procedure. If the parameters of the learning, e.g. the features and dissimilarities, have a strong domain motivation, the domain experts have to test the hypothesis that the details shown by the clustering are a pointer towards interesting further research perspectives.

Identification of regions

The region extraction was a processing step that was needed in nearly all feature extraction chains. This is depicted in the third row of the schematic network representation in figure 6.6. Computationally it meant to extract connected components (CCs) of pixel with equal properties, e.g. with equal tissue type from the SOM evaluation images, see section 6.5.1. To remove small noise structures, we smoothed the borders before calculating the connected components. For the details of the region calculation we refer to section 5.2.2 in (Bornemeier 2011).

Summary of processing sequence and data used

The identification of tissue areas was finally based on the digitized color images of all structural stains that we introduced in section 6.3.2: HE, VIM and AE1AE3. These images were preprocessed using the methods described in section 6.5.1 which includes a conversion into gray values. The preprocessed images were registered to an image stack in the two-stepped registration process explained in section 6.5.1. The resulting image stacks were the input for the calculation of features that were clustered using an unsupervised Self Organizing Map based approach, see section 6.5.1. In the evaluation process the clusters were mapped to tissue types. The pseudo-colored images arising from the evaluation of the clustering were the basis for quantitative analysis of tissue type presence.

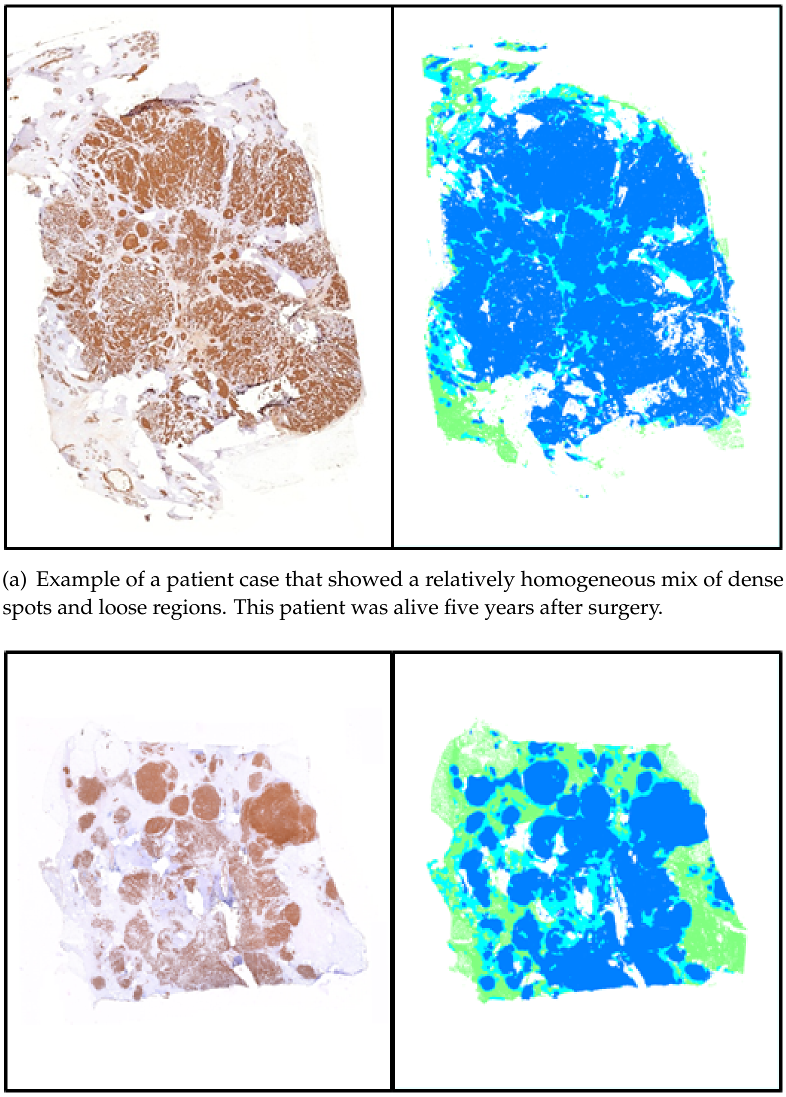
6.5.2 Structural heterogeneity: inner tumor growth patterns

The heterogeneity of a tumor as a potentially relevant medical clue also reflects the different inner growth structures within a single tumor. A characterization of the inner tumor growth structures can answer questions like: Are there areas in the tumor that are dense? Is the tumor completely loosely structured? These questions are important for the characterization of the microembedding of the tumor. It also gives hints about the tumor physiology that is known to influence the response to chemotherapy (Tannock 2001).

Figure 6.18 shows the AE1AE3 images and the SOM images, see section 6.5.1, of a matched pair that was the motivation for the characterization of the inner growth pattern of the tumor. The differences in the inner tumor structures that were apparent in the AE1AE3 stain were not discriminated in the SOM image. This was due to the registration of the structural stains that was not reliable enough to extract stable fine-grained features from the registered stacks. We conducted the detailed inner tumor growth pattern analysis using the AE1AE3 stain alone. The left column in figure 6.6 shows the corresponding information processing steps.

The characterization given by the SOM tissue type differentiation (see section 6.5.1) oriented the fine-grained inner structure characterization. pixel that in the SOM image were identified as tumor parenchyma were considered for the second tissue characterization using the AE1AE3 stain. The discrimination of the different inner tumor structures by human experts was roughly based on texture differences. For their description we used more complex texture representing features as in the SOM analysis. In different tests, that are described in detail in section 4.2 of (Khabirova 2011), we identified a suitable set of four Haralick texture features (Haralick et al. 1973): Haralick's Homogeneity, Sum of squares (variance), Sum Average and Sum Variance. By applying them to the preprocessed gray value images of the AE1A3 stains the texture of the different inner tumor structures was appropriately described over the different patient cases.

As in the SOM tissue characterization, see section 6.5.1, we used an unsupervised approach for the identification of different types of inner growth structures. We conducted different clusterings using the *"kmeans"* function of the Statistics Toolbox for Matlab (Mathworks n.d.c) with the squared Euclidean distance and k varying from 4 to 7 according to the pathological expectation of a suitable number of inner growth patterns. The clustering was trained using the feature vectors of ten chosen patient cases. The most appropriate cluster number according to Silhouette plots was



(a) Example of a patient case that showed a relatively homogeneous mix of dense spots and loose regions. This patient was alive five years after surgery.

(b) Example of a matched pair patient case with different tumor regions in one tumor that show different internal growth patterns. At the upper part of the tissue probe there were tumor regions with a very dense and compact expression of the epithelial marker AE1AE3 whereas the lower tumor regions showed looser expression. This patient was dead five years after surgery.

Figure 6.18: *AE1AE3 stain and SOM images for a matched pair that was the motivation to quantize different inner tumor growth structures. The SOM did not distinguish between the different inner tumor structures.*

five (Khabirova 2011). We show the Silhouette plots for five and seven clusters in figure 4.2. The pathological experts associated the five clusters with the following inner tumor growth structures:

- *red cluster:* solid homogeneous structures
- *blue cluster:* half-homogeneous structures
- *green cluster:* heterogeneous structures
- *yellow cluster:* sparse heterogeneous structures
- *turquoise cluster:* traces of tumor

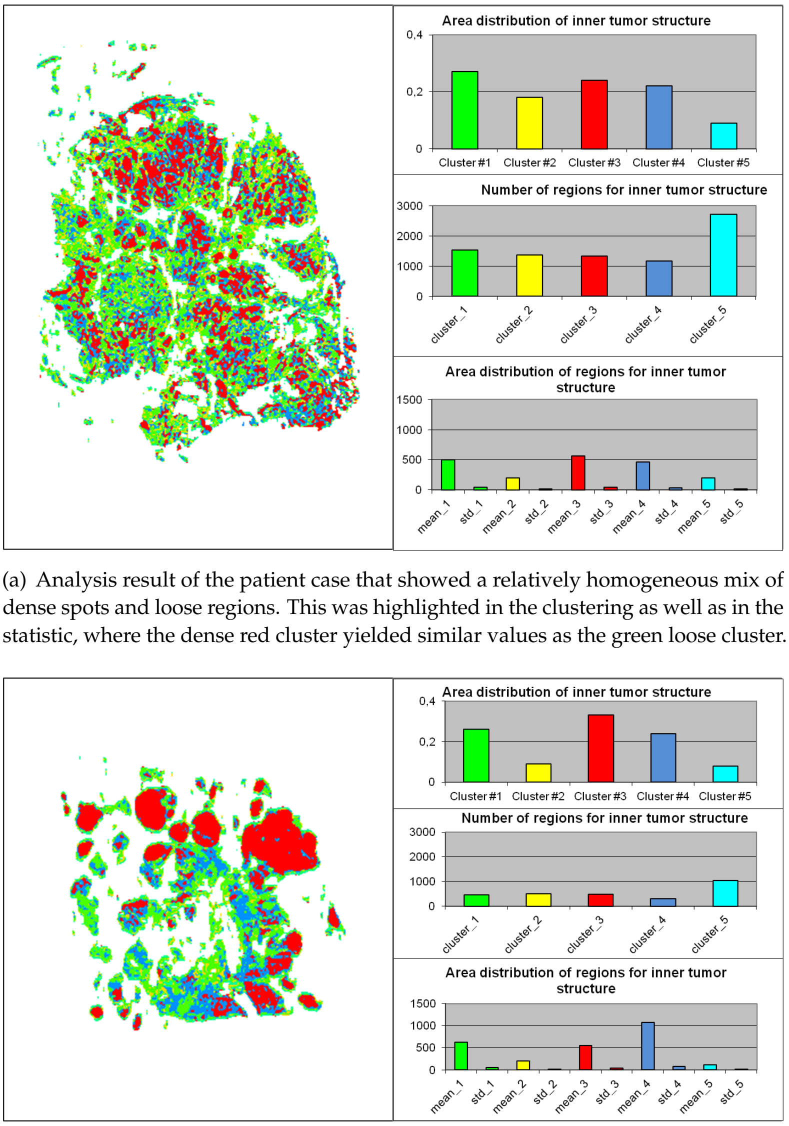
Figure 6.19 shows on the left-hand side the results of the inner tumor structure clustering in pseudo-colored images for the matched pair that was introduced in figure 6.18.

To quantify the distribution of the inner tumor growth structures we calculated how much of the tumor parenchyma was identified as belonging to the five clusters. These statistics are shown at the top of the right-hand sides for both cases of the matched pair in figure 6.19.

To yield a measure of heterogeneity we identified the connected components (CCs) for every inner growth structure cluster using the approach described before in section 6.5.1. The number of the CCs of the inner growth structure clusters gave a measure for the scattering of these inner growth structures. We show these measures in the middle of the right-hand sides for both cases of the matched pair in figure 6.19. The mean and the standard deviation of the area of the CCs were measures for the heterogeneity in the scattering of the single inner structure regions within the tumor. We show these statistics at the bottom of the right-hand sides for both cases of the matched pair in figure 6.19.

Summary of processing sequence and data used

The characterization of the structural heterogeneity was based on the digitized color image of the AE1AE3 stain and the SOM evaluation image resulting from the basic tissue type differentiation presented in section 6.5.1. These images were spatially related according to the registration that was conducted for the tissue type differentiation. The AE1AE3 stain image was preprocessed according to the processes introduced in section 6.5.1. From the SOM evaluation images the regions representing tumor parenchyma tissue were identified using the method introduced in section 6.5.1. Within the spatially corresponding regions of the preprocessed AE1AE3 image, features representing the texture of the inner tumor structure were



(a) Analysis result of the patient case that showed a relatively homogeneous mix of dense spots and loose regions. This was highlighted in the clustering as well as in the statistic, where the dense red cluster yielded similar values as the green loose cluster.

(b) Analysis result of the patient case with different internal growth pattern tumor regions in one tumor. At the upper part of the tissue probe the dense regions were highlighted in red. The red cluster also had the highest area value. In the lower part the tumor regions that showed looser expression were highlighted in yellow and green.

Figure 6.19: *Result excerpt for the inner growth structure clustering for the matched pair cases introduced in figure 6.18. The evaluation images are given at the left-hand side, the calculated statistics at the right-hand side . Details concerning the statistics can be found in the text.*

calculated. We used these features in a clustering process to identify different types of inner tumor structure. Within an evaluation process the experts mapped the clusters to inner tumor structure types. The resulting pseudo-colored inner structure image was the basis for the characterization of the heterogeneity. We also identified regions of different inner tumor structures with the corresponding approach from section 6.5.1 and analyzed them statistically.

6.5.3 Functional heterogeneity: co-occurrence analysis of functional marker expression

The co-occurrence of the functional marker expression with the identified tissue types or with other functional marker expressions is a medical clue with significance for the therapy response prediction concerning e.g. hormone therapy (Horsfall et al. 1989). In current clinical diagnosis the estrogen and progesterone receptor expression is described qualitatively. The decision "positive" is based on a human pathological expert's judgment whether more than 10% of the tumor parenchyma show hormone receptor expression. (Rexhepaj et al. 2008) showed that a precise and reliable quantification of the hormone receptor expression using automatic image analysis gives a thresholding that better corresponds with the prognosis of the patient. In the case of the estrogen receptor the improved thresholding is a valuable diagnostic differentiation for the therapy response prediction.

In our automatic analyses of the functional heterogeneity as it is depicted in the second column of figure 6.6 we considered images showing the expression of the estrogen and the progesterone receptor. Besides these images we analyzed images of the inflammation marker CD45. Inflammation with its different forms is a further relevant factor for different prognosis of the patients. For example (Jahkola et al. 1998) showed that special inflammation markers expressed in the invasion border of the tumor are predictors of local and distance recurrences. (Grivennikov et al. 2010) discuss the general influence of inflammatory processes on the therapy response.

Together with the pathological expert we divided the variety of possibly relevant questions concerning functional heterogeneity into categories that were mappable to computational analysis tasks. We considered

- the local co-occurrence using pixel statistics, as well as
- the heterogeneity of the co-occurrence in different tumor regions based on the connected component information and
- the geometrical heterogeneity analyzing in which part of the tumor regions the functional markers are expressed, e.g. in the periphery.

Figure 6.20 shows the PR stain and SOM images of a matched pair that was the motivation for the characterization of the co-occurrence analysis of functional marker expression for the tumor.

To relate the functional information of a patient to the structural information, we applied the registration process introduced in section 6.5.1 to the HE stains and the functional stains ER, PR and CD45 (figure 6.5). To model functional aspects of heterogeneity, the expression of the functional markers was localized in the tumor slice.

Localization of functional marker expression

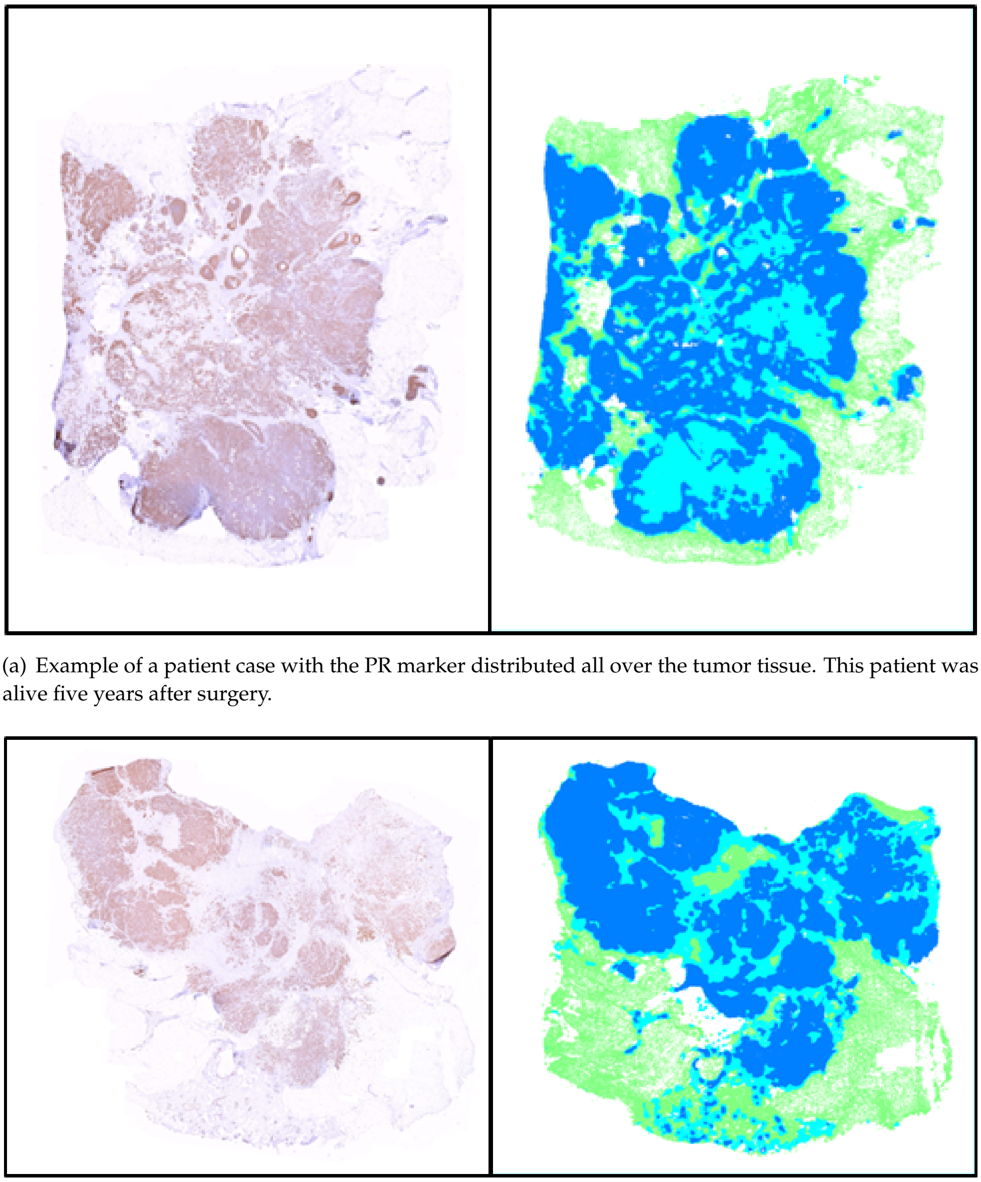
To detect and quantify the functional marker expression, we used the preprocessed registered CD45, ER and PR stain images. This was appropriate as within most cases the stain of the functional marker was significantly more intensive than the counter stain. The color information was neglected without significant information loss. Because of the high variation of the staining intensity in different patients, it was not suitable to identify the marker expression using a single intensity threshold over all stain images. We learned the marker expression identification for every functional marker in a supervised manner. The mean and standard deviation of the intensity were extracted as features representing the texture of the tissue for each pixel of the functional marker images in an area of 19×19 pixel around this pixel.

The pathological experts annotated regions of marker response on the original functional marker images and as contrast also highlighted non responding regions. In order to induce stable identification it was necessary to mark especially such regions that were near to the decision boundary of whether there is marker response or not. The annotations of four patients were used to train a Linear Discriminant Analysis with the squared Euclidean distance as dissimilarity measure.

The results of this binarization were evaluated by computer scientists as well as pathological experts using an overlay of the binarized image over the original image with adjustable transparency. If the binarized image is interpreted as a classification into background and marker response this is comparable to the pseudo-color image evaluation introduced in section 6.5.1. From this characterization we gained a *LDA image* for every probe highlighting pixel with detected functional marker expression. On the left-hand side of figure 6.21 we show the LDA images for the cases of the matched pair introduced in figure 6.20 overlaid to the corresponding SOM images.

Co-occurrence analysis based on pixel statistics

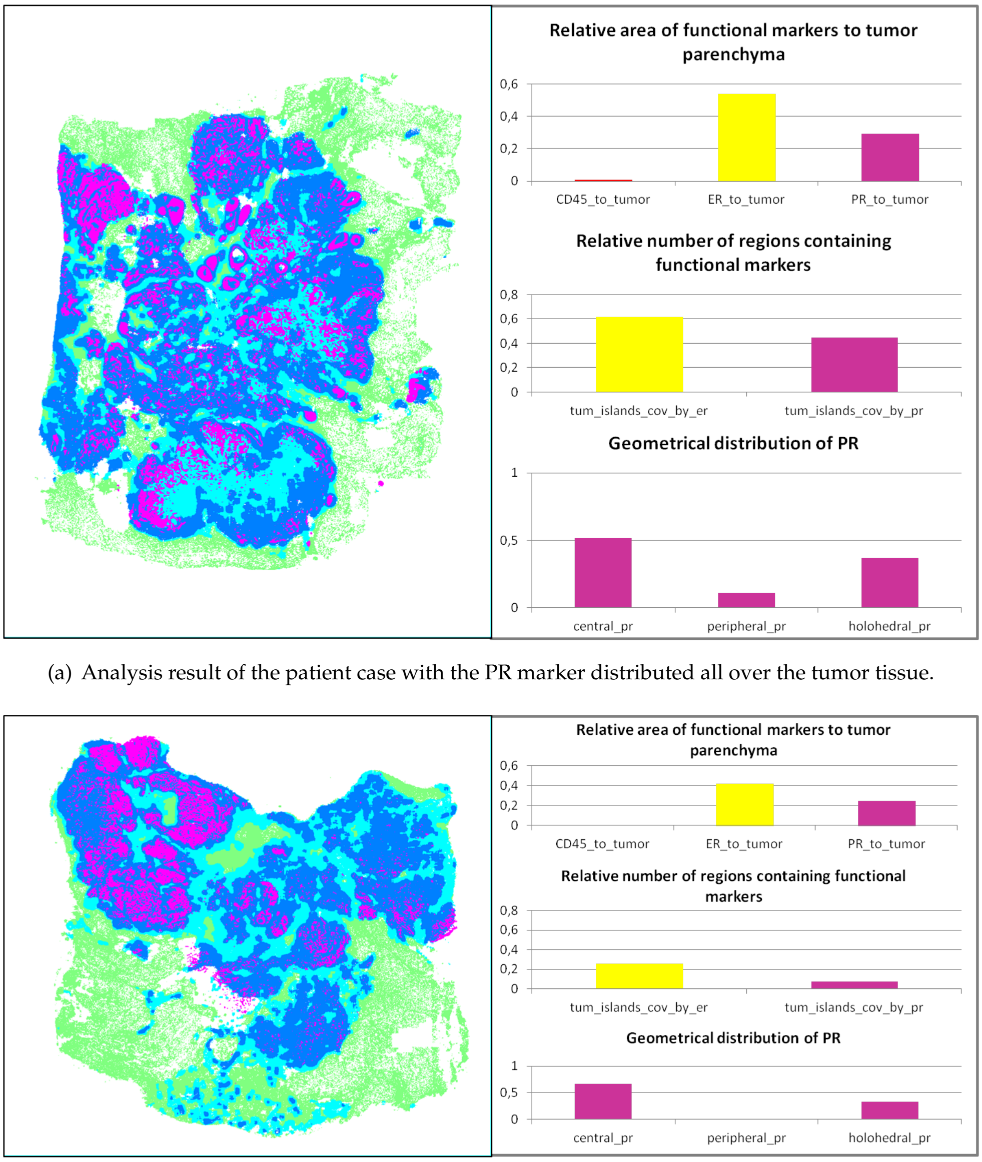
The co-occurrence feature groups based on pixel statistics described a local context of co-occurrence and answered questions like: Is there any estrogen receptor expression



(a) Example of a patient case with the PR marker distributed all over the tumor tissue. This patient was alive five years after surgery.

(b) Example of a matched pair patient case with a heterogeneous distribution of PR over the different tumor regions, some regions did not show an expression whereas others were partly expressing PR. This patient was dead five years after surgery.

Figure 6.20: *Registered PR stain and SOM images for the matched pair that was a motivation to calculate the co-occurrence of functional expression and structural information.*



(a) Analysis result of the patient case with the PR marker distributed all over the tumor tissue.

(b) Analysis result of the matched pair patient case with a heterogeneous distribution of PR over the different tumor regions.

Figure 6.21: *Result extract for the co-occurrence analysis for the matched pair cases introduced in figure 6.20. At the left-hand side the overlay of the PR marker expression localization highlighted in magenta over the SOM images is shown. At the right-hand side statistics for every co-occurrence category – pixel wise, regions based, geometrical – are shown. A detailed description is given in the text.*

Medical question	Computationally implemented feature groups
Expression of [CD45, ER, PR] relative to tumor parenchyma	Number of pixel highlighted as [CD45, ER, PR] by the number of pixel classified as tumor parenchyma
Expression distribution of [CD45, ER, PR] in relation to [tumor parenchyma, tumor stroma, normal tissue]	Number of pixel highlighted as [CD45, ER, PR] and classified as [tumor parenchyma, tumor stroma, normal tissue] by the number of pixel highlighted as [CD45, ER, PR]
Amount of co-occurrence for [CD45, ER, PR]	Number of pixel highlighted as [CD45, ER, PR] and highlighted as [CD45, ER, PR] by the number of pixel highlighted as [CD45, ER, PR]

Table 6.8: *Overview of local context co-occurrence feature groups from pixel in classified images*

in the tumor, and if so, how much? Does the progesterone receptor co-occur with the estrogen receptor? Is there inflammation in the tissue and if so, how much in which tissue type?

Table 6.8 shows an overview of the local context co-occurrence feature groups that we calculated for every patient case. It lists the medical question that characterizes the patient's situation in a potentially relevant way. Correspondingly the right part shows the mapping to machine calculable feature groups. These feature groups were derived from the tissue characterizing SOM evaluation image as well as from the LDA images representing the functional marker expression.

The amount of functional marker expression in relation to the tumor parenchyma area, is currently the base for the categorization in diagnostic parameters. We expressed this relative amount by the number of pixel identified as functional marker expression divided by the number of pixel identified as tumor parenchyma. Figure 6.21 shows this basic expression statistic for all considered functional markers at the top of the right-hand side for both cases of the matched pairs.

We quantized the distribution of the functional marker expression in different tissue types. For this purpose we related the number of pixel that showed a functional marker expression in a specific tissue to the overall number of pixel showing this functional marker expression. The co-occurrence distribution of various functional markers was gained by relating the number of pairwise combined functional marker expression pixel to the overall number of pixel for a single functional marker expression.

Medical question	Computationally implemented feature groups
Degree of scattering of [CD45, ER, PR, tumor parenchyma]	Number of CCs identified as [CD45, ER, PR, tumor parenchyma]
	Mean area of CCs identified as [CD45, ER, PR, tumor parenchyma]
	Standard deviation of area of CCs identified as [CD45, ER, PR, tumor parenchyma]
Distribution of [CD45, ER, PR] in regions classified as tumor parenchyma	Number of CCs classified as tumor parenchyma containing pixel highlighted as [CD45, ER, PR] by the overall number of CCs classified as tumor parenchyma

Table 6.9: *Overview of heterogeneity characterization for co-occurrence from regions in classified images*

Co-occurrence analysis based on connected components

This analysis handled a more complex interpretation of the concept of heterogeneity in functional marker distribution which is closer to the domain experts' view. It answered questions like: Is the distribution of the estrogen receptor expression similar in all tumor regions of the tumor excerpt?

To answer such questions a more global context of the marker expression was needed. For these medical clues the pathological experts identified different tissue regions in the patient's HE stain and the functional marker image. We calculated the connected components (CCs) gained from the tissue characterizing SOM evaluation image as well as from the functional marker LDA evaluation images for every patient case according to the approach introduced in section 6.5.1. Table 6.9 summarizes the functional questions and feature groups that we identified using the structural and functional CCs. It lists the medical questions that characterized the patient's situation in a potentially relevant way, and correspondingly gives the mapping to the computationally implemented feature groups.

With the CCs of the tissue types, the question for the degree of scattering of a functional marker was expressed in terms of feature groups on the CCs. The degree of scattering was specified through three interacting properties: the number of regions of a specific type as well as the size and the size variation of these regions. It was mapped to corresponding computational feature groups: the number of the CCs and the mean and standard deviation of the area of the CCs.

The distribution of the functional markers in the tumor regions was represented by the relative number of CCs identified as tumor parenchyma containing the specific functional marker. To avoid misinformation by noise, a CC was defined as containing

Medical question	Computationally implemented feature groups
Geometrical distribution of [CD45, ER, PR] in regions classified as tumor parenchyma	Number of CCs classified as tumor parenchyma showing [peripheral, central or holohedral] distribution of [CD45, ER, PR] by the overall number of CCs identified as tumor parenchyma

Table 6.10: *Overview of heterogeneity characterization for co-occurrence concerning geometrical constellations from regions in classified images*

a functional marker when a minimum of 0.5% of the CC's area was covered by the marker. Figure 6.21 shows the relative number of expression regions for ER and PR at the middle of the right-hand side for both cases of the matched pairs.

Co-occurrence analysis based on geometric constellations

The last category of co-occurrence feature groups was related to the geometric constellation of the functional marker expression with respect to the tumor regions, e.g. tumor regions that had peripheral inflammation. As given in table 6.10 we distinguished whether the functional marker was expressed in the periphery of the tumor region, in its center or all over the region (holohedral). These medical clues are of special importance for the prognosis as discussed e.g. in (Jahkola et al. 1998).

To map these questions onto machine extractable feature groups, we sectioned the tumor parenchyma CCs into inner and outer "rings" using the so-called *distance transform*, cf. (Fabbri et al. 2008). It gives the radial distance of a pixel to the border of a CC. At a distance of half the mean radius of the CC we determined the cut for the rings. The pixel that were highlighted by a functional marker inside the inner and outer region were counted respectively. We calculated the distribution of holohedrally, peripherally and centrally covered regions from this characterization of the functional coverage per CC. Figure 6.21 shows this distribution of the geometrical constellation for PR at the bottom of the right-hand side for both cases of the matched pair.

Summary of processing sequence and data used

For the characterization of the functional heterogeneity the digitized color images of the functional stains CD45, ER and PR were preprocessed using the methods described in section 6.5.1. The resulting functional images were spatially related to the coordinate system of the tissue type differentiation using the registration process introduced in section 6.5.1 for the HE and the functional stain images. In the spatially aligned functional images the localization of the corresponding marker expression was achieved by the LDA based classification process explained in section 6.5.3.

The pseudo-colored LDA images, arising in the evaluation process of the classification, together with the SOM evaluation image of the tissue type differentiation, see section 6.5.1, were the basis for the calculation of the co-occurrence characterization based on pixel statistics. For the second complex of feature groups we calculated the marker expression regions from the pseudo-colored LDA images using the region identification approach described in section 6.5.1. The different tissue regions were identified using this approach on the SOM evaluation image. With this information the co-occurrence analysis based on connected components was conducted. For the co-occurrence analysis based on geometric constellations the same set of marker expression and tissue regions was used. The tumor tissue regions were split additionally into an inner and an outer ring using the distance transform, see section 6.5.3, to gain a geometrical characterization of the marker expression.

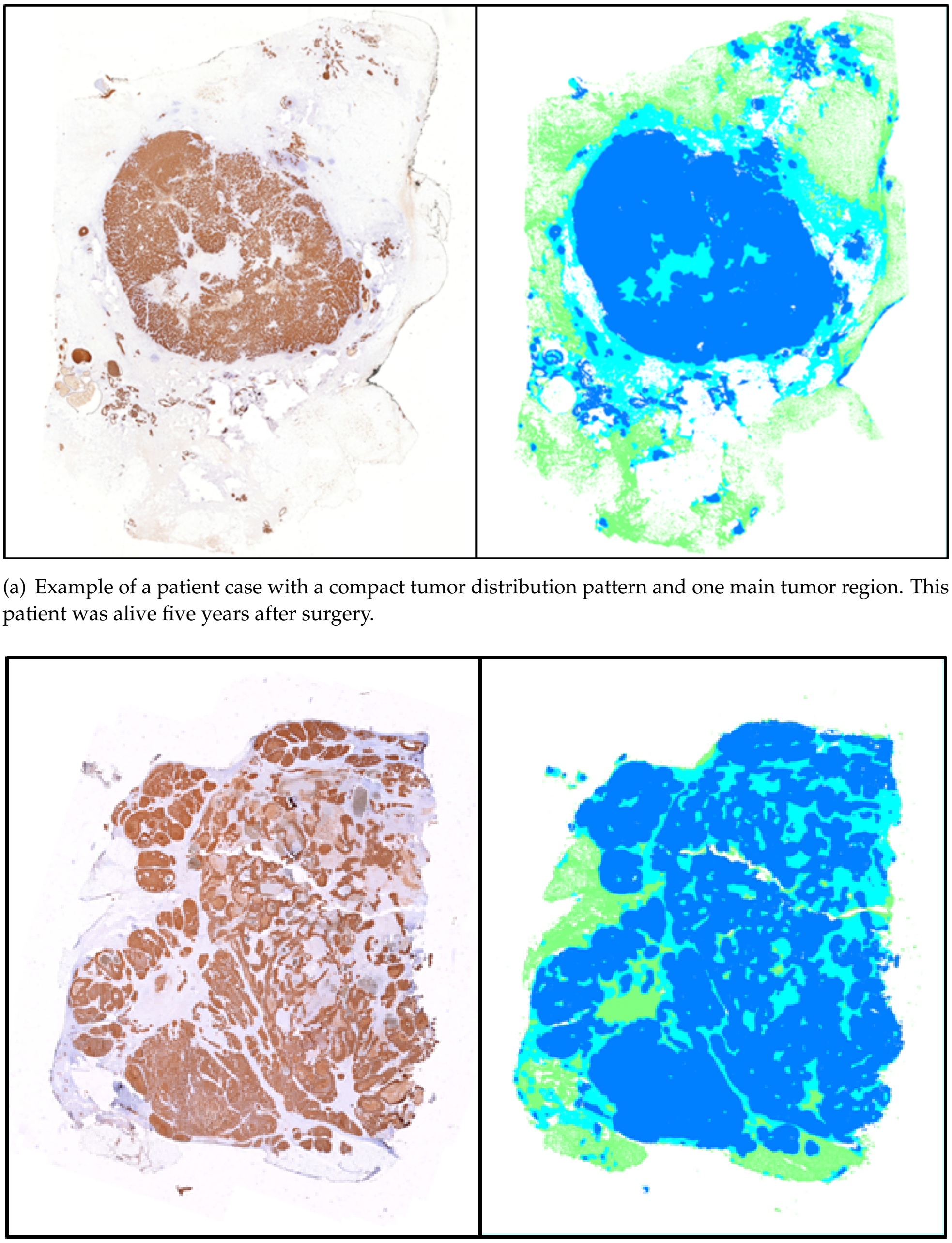
6.5.4 Structural tumor distribution patterns: morphometry and graph based analysis of tumor distribution patterns

This collection of feature groups was concerned with questions about the structural distribution pattern of the tumor such as: Is the tumor scattered or compact? Are there separated small tumor regions besides the main tumor? Figure 6.22 shows the AE1AE3 and SOM images of a matched pair that was the motivation to characterize the structural tumor distribution pattern. According to (Rangayyan et al. 1997) a compact growth pattern is less risky as compared to a tumor distribution pattern which shows invasive tumor regions outside the compact main tumor.

In our corresponding automatic feature extraction, shown in the fourth column in figure 6.6, we modeled two aspects of the structural distribution:

- For the discrimination of scattered and compact distribution patterns we used *graph based representations* of the tumor tissue distribution.
- To account for information about the single tumor regions, we analyzed the *morphometry of the regions*.

These morphometric representations were analyzed on two levels of detail. For the coarse-grained description we used the SOM tissue characterization as basis. For the finer characterization of the distribution pattern of the tumor also a finer characterization of its structures was needed. We analyzed the AE1AE3 stain images on their own, to get a fine-grained view on the tumor's structural distribution properties. For this purpose we calculated the fine AE1AE3 marker expression that was used as basis for the graph as well as for the morphometric representation.



(a) Example of a patient case with a compact tumor distribution pattern and one main tumor region. This patient was alive five years after surgery.

(b) Example of a matched pair patient case with many small regions of compact growth spreading in the tissue. This patient was dead five years after surgery.

Figure 6.22: *Registered AE1AE3 stain and SOM evaluation images for the matched pair that was a motivation to characterize the structural tumor distribution pattern.*

AE1AE3 marker expression identification

In the AE1AE3 images the area that responds to the marker is stained in shades of brown whereas the area responding to the counter stain is violet. Color gave relevant information for the identification of the fine tumor regions that was not contained in the preprocessed AE1AE3 stains. To remove the influence of irrelevant intensity variation between the images, we used the *CIE L*a*b** color space representation of the images. For the details of this color space conversion we refer to section 5.2.1 in (Bornemeier 2011).

We applied Otsu's thresholding approach (Otsu 1979) to every single channel in the L*a*b* color space and combined the results to yield the pixel characterization of the marker expression. This approach proved to be the most stable over all cases. It removed staining artifacts at the margin of the tissue. The discrimination between brown and violet was appropriate (Bornemeier 2011). The evaluation of the results was realized by overlays of the binarized image to the original image with adjustable transparency.

We calculated the fine tumor regions from this AE1AE3 expression analysis by applying the connected component identifying approach that was introduced in section 6.5.1. Figure 6.23 shows at the left-hand side the fine tumor regions as base for the graph models.

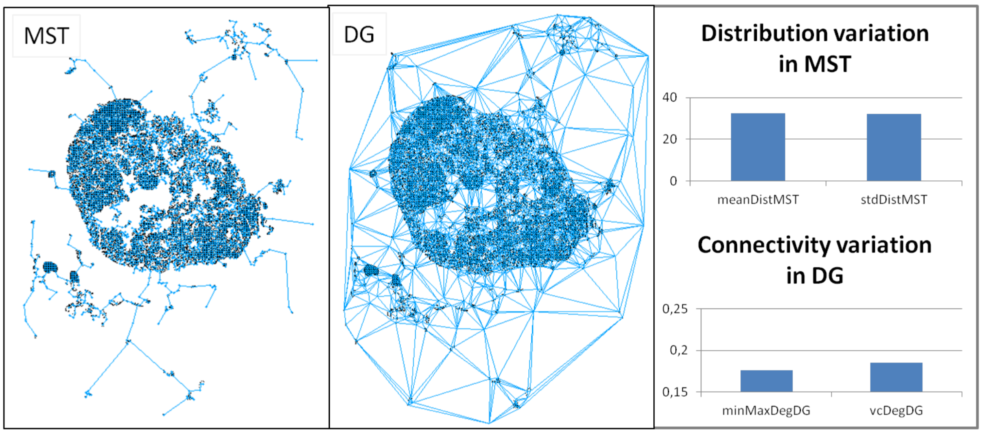
Graph based representation

The graph based analysis of the tumor distribution considered the fine distribution patterns. It was based on the fine-grained AE1AE3 expression analysis described above in the last section. To derive a basic graph representation from this tissue analysis, the nodes of the graph model had to be placed adequately among the pixel identified as tissue responding to the AE1AE3 marker. We used a grid based method to determine the position of the graph nodes that is explained in detail in section 5.4.1 of (Bornemeier 2011). Using this basic graph we induced two graph models:

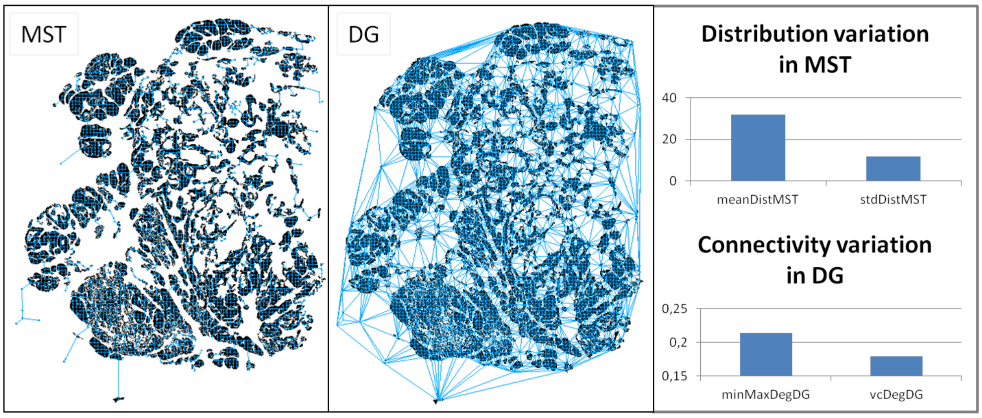
- a *minimum spanning tree* (MST) and
- a *Delaunay graph* (DG).

We calculated features that described the resulting tree models to discriminate the various forms of tumor distributions.

The MST was applied to model a rough estimate of the distribution and connectivity of the tumor regions. From the previously determined graph nodes we calculated the minimum spanning tree using Prim's algorithm (Prim 1957) as it was implemented in the Bioinformatics Toolbox for Matlab (Mathworks n.d.a). The



(a) Analysis result of the graph based analysis of the patient case with a compact tumor distribution pattern and one main tumor region.



(b) Analysis result of the graph based analysis of the matched pair patient case with many small regions of compact growth spreading in the tissue.

Figure 6.23: *Result extract for the graph based analysis of the matched pair cases introduced in figure 6.22. At the left-hand side the MST and in the middle the DG graph models were overlaid to the fine AE1AE3 tumor regions. At the right-hand sides features gained from the graph models are depicted. They are explained in the text.*

lengths of the edges were represented by the Euclidean distances of the image pixel coordinates of the graph nodes adjacent to the considered edge. Figure 6.23 shows an example of the resulting MSTs at the left-hand sides for both cases of the matched pair that was introduced in figure 6.22.

Medical question	Computationally implemented feature groups
Distribution variation of tumor parenchyma	Mean and standard deviation of the edge lengths
	Variation coefficient of the edge lengths
	Relation from minimal to maximal edge lengths
Connectivity characterization of the tumor parenchyma	Average weighted node degree (sum of edge lengths for a node to its neighbors)
	Cyclomatic number
	Randić index
Connectivity variation of the tumor parenchyma	Mean and standard deviation of the node degrees
	Variation coefficient of the node degree
	Relation from minimal to maximal node degrees

Table 6.11: *Overview of feature groups calculated from the graph models for a structural tumor distribution pattern characterization*

The DG gave a model of the connectivity and distribution that was more complete and detailed than the MST. We calculated the Delaunay graph using Matlab's build-in "*delaunay*" function. Figure 6.23 shows an example of resulting DGs at the middle for both cases of the matched pair.

For each graph model we calculated the set of feature groups that is given in table 6.11. This feature group set described properties of the structural representations by the graph models to map potentially relevant medical questions. For the motivation, definition and calculation of the single feature groups we refer to section 3.2.3 in (Bornemeier 2011).

Figure 6.23 shows at the upper part of the right-hand sides examples for the representation of the distribution variation using the mean and standard deviation of the edge lengths for the MST for both cases of the matched pair. In the lower part of the right-hand sides the connectivity variation by the relation from minimal to maximal node degrees and the variation coefficient of the node degree is depicted respectively.

(Bornemeier 2011) stated that in the case of the MST the cyclomatic number was no pertinent feature as the MST is defined to be cycle free. We will not consider this feature for the MST in the analyses.

Morphometric representation

The morphometric representation aimed at characterizing the overall tumor region properties that concern the shape and the contour of the regions. We realized two kinds of morphometric representations:

coarse-grained representation gained from the tumor regions identified by the SOM tissue characterization in section 6.5.1 and

fine-grained representation based on the fine tumor regions from the AE1AE3 expression analysis introduced in section 6.5.4.

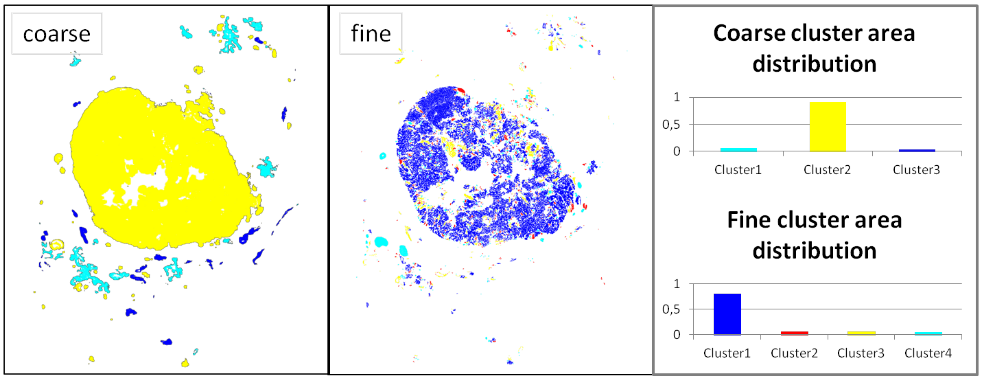
To generate these representations for the tumor regions of the SOM as well as of the AE1AE3 we calculated a variety of shape and contour describing features: Fourier descriptors, Fourier energies, moment invariants, irregularity, compactness, roundness, area and perimeter. One example for the applied contour describing features were the *Fourier descriptors*, see e.g. (Zhang and Lu 2002), that are capable of describing the contour of the regions invariant from rotation, translation and scale. For details of the applied features we refer to section 3.1 in (Bornemeier 2011).

These features were calculated for all fine and all coarse regions larger than 15 pixel in area for every patient. For the fine regions in 93 patient cases this is a total of 216196 samples. The coarse as well as the fine tumor regions were clustered according to their morphometric properties expressed in the features. As the sample size was that big we decided to train the clustering using a subset of the data. The pathological experts chose 31 patient cases for this purpose. We trained different clusterings over all regions from that choice using the *"kmeans"* function of the Statistics Toolbox for Matlab (Mathworks n.d.c) with the squared Euclidean distance varying the k from 3 to 7 according to the pathological experts' suggestion.

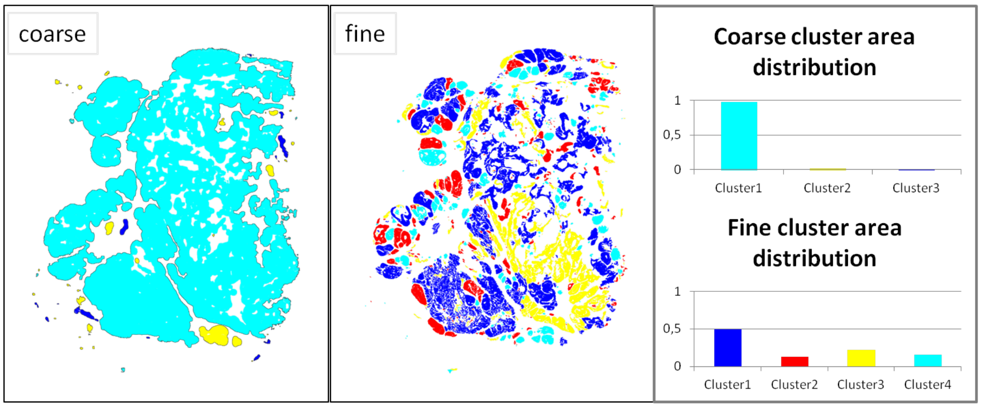
Neither the pathological nor the mathematical evaluation of the clustering yielded a clear preference of a cluster number. Only the clustering with 6 clusters did not yield pertinent results. We incorporated all other clustering variants with their corresponding statistical features into the relevance analysis described in chapter 7.

Figure 6.24 shows the results of a coarse clustering using three clusters at the left-hand sides for both cases of the matched pair that was introduced in figure 6.22. In the middle the corresponding fine clustering with four clusters for both cases is shown.

From the clustering results we calculated for every patient case the area that was covered by regions belonging to the coarse and the fine clusters. These statistics are shown at the right-hand sides of figure 6.24. In the upper part the statistic for the coarse clustering is depicted whereas the lower part represents the statistic for the fine clustering.



(a) Analysis result of the morphometric analysis of the patient case with a compact tumor distribution pattern and one main tumor region.



(b) Analysis result of the morphometric analysis of the matched pair patient case with many small regions of compact growth spreading in the tissue.

Figure 6.24: *Result extract for the morphometric analysis for the matched pair cases introduced in figure 6.22. At the left-hand side the clustering according to the morphometric properties of the coarse tumor regions using three clusters and in the middle of the fine tumor regions with four clusters is shown. The distributions extracted for the coarse and fine morphometric clusterings are depicted at the right-hand side.*

Summary of processing sequence and data used

We based the characterization of the structural tumor distribution pattern on the digitized color image of the AE1AE3 stain and the SOM evaluation image resulting from the tissue type differentiation in section 6.5.1. The AE1AE3 stain was analyzed using the Otsu based clustering approach described in section 6.5.4 for the identification of the marker response.

For the graph based analysis of the structural tumor distribution pattern the area identified as marker responding was represented by a basic graph that was used to induce a minimum spanning tree as well as a Delaunay graph, see section 6.5.4. Features calculated over these graphs were used for the tumor distribution pattern representation.

The morphometric representation was based on the pseudo-colored Otsu image created in the evaluation of the marker response identification as well as on the SOM evaluation image. We calculated the fine regions of structural marker response using the region identification approach explained in section 6.5.1 from the Otsu image. The same approach was used for the identification of coarse tumor parenchyma regions from the SOM evaluation image. For each of the fine and coarse tumor regions respectively a set of morphometric features was calculated. Separate clustering over these features for the fine and the coarse regions led to a characterization of different morphometric tumor region types. Statistics over the regions according to these region types yielded the morphometric representation of the tumor distribution pattern.

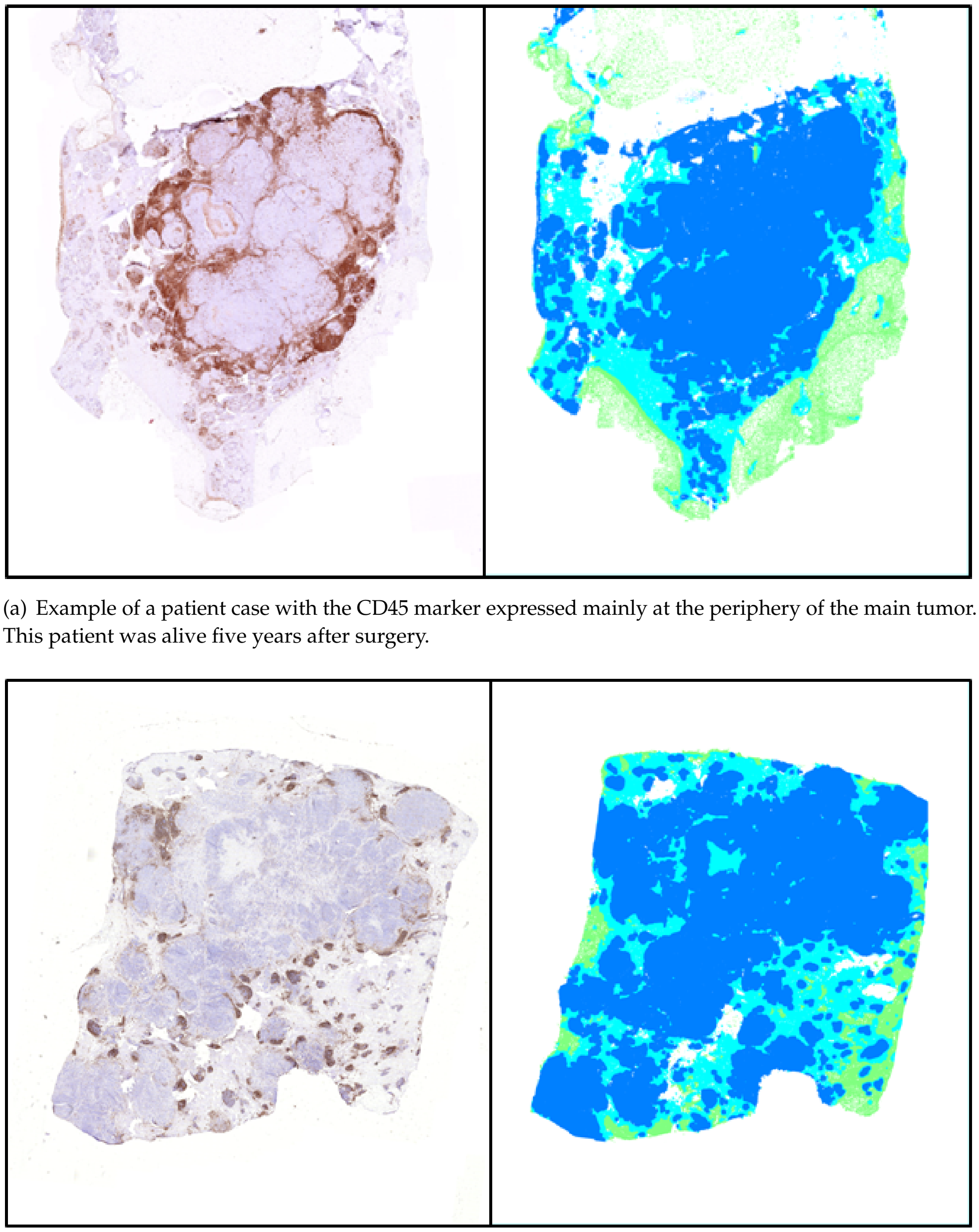
6.5.5 Functional tumor distribution patterns: relation based analysis of tumor distribution patterns in correlation with functional marker expression

This collection of feature groups gave hints for answering questions such as: What is the relation of the functional marker distribution pattern to the distribution pattern of the tumor? Are they equal? Figure 6.25 shows the CD45 and SOM images for the matched pair that motivated these analyses. To answer the corresponding questions we considered two relational approaches:

- the Region Connection calculus features, known as *RCC8* features cf. (Randell et al. 1992), as well as

- the Linear Distance Quantification, cf. section 5.5.2 in (Bornemeier 2011).

The third column in figure 6.6 depicts the corresponding processing chain. Both representations were calculated using the regions gained from the SOM tissue characterization, see section 6.5.1, and the regions from the functional marker expression analysis, see section 6.5.3. The left-hand side of figure 6.26 shows the functional marker expression results overlaid to the SOM images for both cases of the matched pair.



(a) Example of a patient case with the CD45 marker expressed mainly at the periphery of the main tumor. This patient was alive five years after surgery.

(b) Example of a matched pair patient case with peripheral and central expression of CD45 in different tumor regions. This patient was dead five years after surgery.

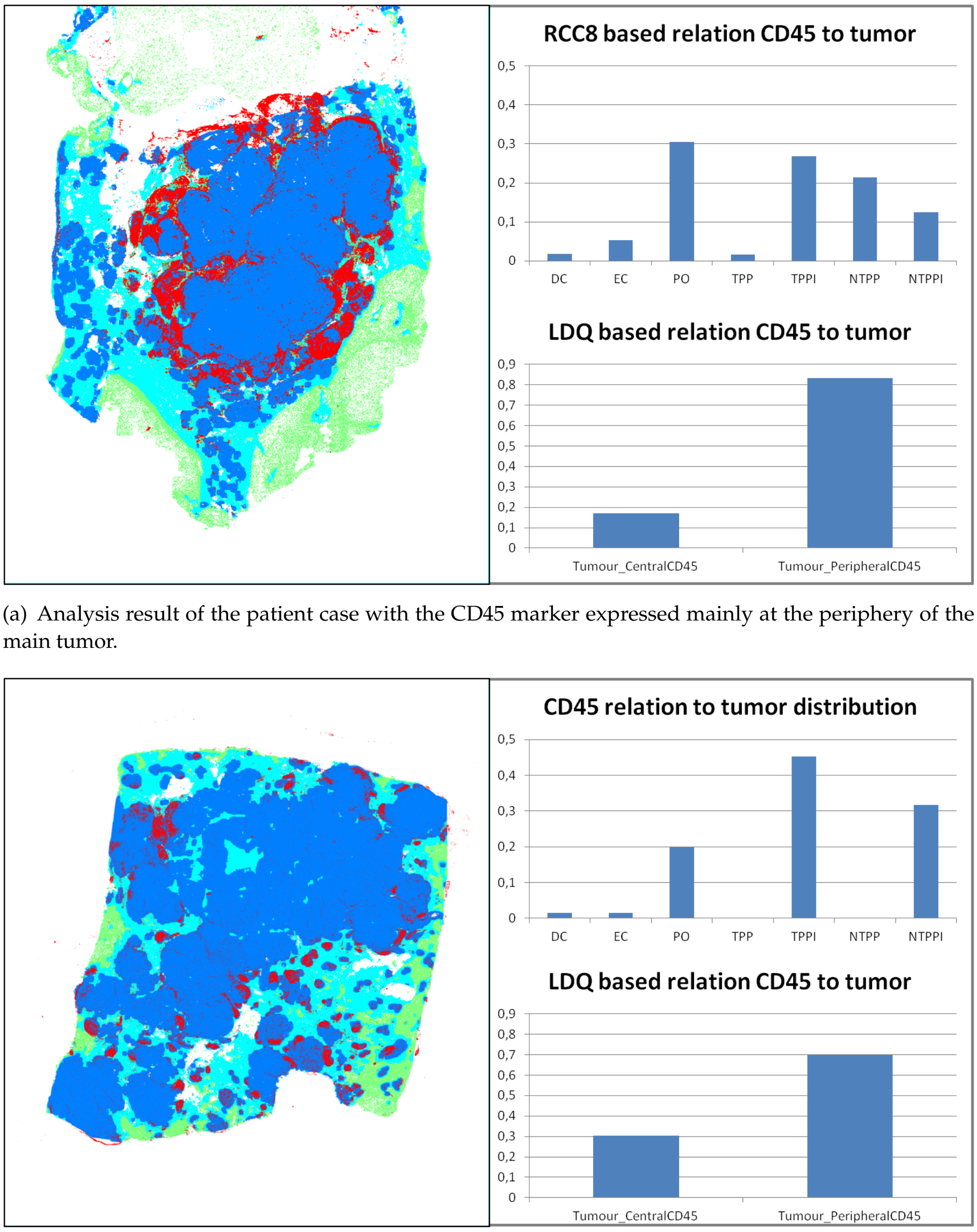
Figure 6.25: *Registered CD45 stain and SOM images for the matched pair that was a motivation to calculate the relational tumor distribution for functional expression.*

The region connection calculus features for a set of regions give the distribution of the present pairwise relations between the regions, e.g. partially overlapping, tangential proper part or disconnected. In our application, we used these features to characterize the relation between the tumor parenchyma regions and the functional marker regions of CD45, ER and PR. For details we refer to section 5.5.1 in (Bornemeier 2011). Figure 6.26 depicts at the upper part of the right-hand sides the distribution of the RCC8 relations for both cases of the matched pair.

Additionally (Bornemeier 2011) developed the Linear Distance Quantification. This approach was analog to the geometrical representation in the heterogeneity of the functional marker expression, cf. section 6.5.3. Figure 6.26 shows the results of the LDQ for the matched pair at the bottom of the right-hand side.

Summary of processing sequence and data used

The characterization of the functional tumor distribution pattern was based on the result images of the tissue type differentiation, see section 6.5.1, and the functional marker expression localization, see section 6.5.3, respectively. From all these evaluation images the corresponding regions were calculated using the region identification approach introduced in section 6.5.1. These regions were analyzed according to their relations using the RCC8 and the LDQ approach.



(a) Analysis result of the patient case with the CD45 marker expressed mainly at the periphery of the main tumor.

(b) Analysis result of the matched pair patient case with peripheral and central expression of CD45 in different tumor regions.

Figure 6.26: *Result extracts for the relational analysis for the matched pair cases introduced in figure 6.25. At the left-hand side the overlay of the CD45 marker expression localization highlighted in red over the SOM images is shown. At the right-hand side the statistics for the RCC8 relations is shown in the upper part. The relations calculated by the LDQ are shown at the bottom of the right-hand side.*

Chapter 7

Relevance analysis for medical clues – Application of VQ based framework for mixed data in breast cancer research

To analyze the relevance of the chosen medical clues in breast cancer in general it is necessary to integrate all clue representing feature groups in an overall multi-layer model. This way all possible contextual influences between the different feature groups are considered in the automatic relevance analysis.

According to the identified relevance of the feature groups in the integration the choice of the feature groups has to be reduced in order to become relevant for diagnostic processes in the clinical routine or in following pharmacological research. In every case the pathological experts have to be coupled to the system to validate and interpret the relevances and choices.

In the complete integration our set of 72 feature groups with a total of 234 dimensions yields an enormous amount of free variables. For example with only one prototype for every follow-up status and the vector-based integration of the feature groups[1] we have a total of $72 + 2 \times 234 = 540$ free variables. This number is opposed to the number of 93 patient samples in the Exprimage data set, see section 6.3, from which the free variables have to be inferred[2].

We introduce a preliminary reduction step to select a set of candidate feature groups that we use for the integrated analysis. This reduces the number of integrated feature groups and the overall dimension. We provide evidence in this *isolated relevance analysis* in a twofold manner:

Mathematical evidence We analyze every feature group singularly according to its discriminative power against the follow-up status of the patients in terms

[1] The vector-based integration has an order of $J + N \times M$ free variables which is much less than $J^2 + N \times M$ free variables for the matrix-based integration. J is the number of feature groups, N the number of prototypes and M the overall feature dimension.

[2] There are research activities in the field of statistical learning theory that work on good approximations for the needed number of samples given a known model complexity (cf. e.g. McAllester (Mcallester 2003)). As these approximations are often conservative they lead to large numbers of samples. These numbers can not be expected in our application example and thus we do not consider these approximations.

of recognition rate and Cohen's kappa. This neglects potential contextual interactions between the feature groups.

Pathological evidence The results of the automatic isolated relevance analysis are evaluated by and discussed with the pathological experts. They provide the contextual embedding of the feature groups and gather potentially interesting candidate sets of feature groups from the ones scoring best in terms of the discriminative power.

The candidate sets are intentionally chosen larger than suited for clinical routine to show the process of feature group selection in the integrated relevance analysis. In our application example this reduction is needed to reduce the number of free variables and provide a recognition that has a better generalization ability. The reduction prevents overfitting the model to the training data. In applications with a sufficient number of samples the reduction is necessary for the identification of clear and stably conductible diagnostic patterns.

The analyses as well as the remarks in this chapter aim to test the following **hypotheses**

1. Candidate sets of feature groups can be reduced using the relevances identified in the integrated relevance analysis such that the generalization of the learned classification becomes more stable and better. We estimate the stability of a system by the standard deviation of the considered evaluation measure over several runs. In our application, the quality of the results is measured by the recognition rate and/or Cohen's kappa over the test set according to the ability of the learned model to predict the follow-up status of the patients.

2. A suitably integrated combination of feature groups yields more stable and better generalization results than the use of a single feature group. Additional to the above mentioned estimation of stability, the reliable convergence of the system is subsumed in this term.

3. A suitably integrated combination of feature groups yields better results than the commonly accepted classification by the pathological grading alone.

4. A suitably integrated combination of feature groups with their conceptually adequate dissimilarities yields more stable and better generalization results than the use of a comparable classification with overall applied Euclidean distance.

5. The outliers identified by the application of a leave-one-out validation on the previously identified best integrated combination of feature groups are

salient in their pathological interpretation, e.g. they show preparation failures or exceptional distribution patterns.

The comparison with the current clinical diagnosis is not straight forward. As discussed in section 6.3 the grading is the summary of the current diagnostic process. It comprises three categories that express a good, a medium and a bad prognosis for the patient. From the second category no conclusions on the outcome can be made. Furthermore the follow-up status, which is the actual benchmark, can also be a relapse. We neglect this fact as we do not have enough sample data for this class. We return to the question of comparability in section 7.3.2.

General considerations about the settings in the application of VQ based algorithms can be found in section 4.4. In the following section we give a summary of the settings and optimizations used in our application.

Section 7.2 is concerned with the isolated relevance analyses yielding the basis for the identification of feature group candidate sets as well as the reference frame for the comparison of the discriminative power achieved with the integrated feature groups. In this section also the discussion and decision on candidate sets is reflected. It was conducted together with the pathological experts.

In section 7.3 we show the evaluation of relevances for integrated feature group sets. Furthermore the relevances are used for the reduction of the candidate feature group sets. In this section we test the hypotheses 1 to 3. We conduct tests concerning hypothesis 4 in section 7.4. In section 7.5 we discuss the possibility of testing hypothesis 5. In this chapter we give the results and a short discussion of the tests as well as the resulting changes for subsequent tests. The validity of the hypotheses is discussed in section 8.1.

Additionally to the tests of the hypotheses we analyzed the behavior of a reduction process that is only calculated from patient samples with the grading value two. As mentioned before from this grading value there is up to now no reliable prognosis possible in the current clinical prognostic process.

7.1 Vector Quantization methods in the Exprimage context

In this section we give pragmatic considerations about the settings for a pertinent use of Vector Quantization methods in our biomedical context. We do not give theoretical foundations but rather rules of thumb that proved good results in the Exprimage project.

7.1.1 Chosen algorithms for workflow

An overview of the whole set of 72 potentially relevant feature groups is given in table 7.2. For the isolated relevance analysis we used an adapted GLVQ algorithm, introduced in section 4.1.2, with the conceptually adequate dissimilarity for every single feature group. For the relational feature groups the KLVQ algorithm, see section 4.2.2, was used. The relevance of the single feature groups was judged according to their predictive ability for the clinical follow-up of the patients.

In the integrative analyses we considered the algorithms introduced in chapter 5 that use the vector-based integration of the mixed data as they have a significantly smaller number of free variables, cf. footnote 1 in this chapter, than the matrix-based integration. The relevance in the integrative analysis was determined by the dissimilarity parameters identified by dissimilarity adaptation. Dissimilarity adaptation in the context of clustering is critical as with unsuitable initializations the results are inappropriate. With small data sets the suitability of initializations in terms of representativeness is not likely. For that reason, we did not consider unsupervised dissimilarity adaptation approaches in the Exprimage application. As for median approaches more data samples would have been required we also did not implement these variants for our application. Due to the relational feature groups in our data set we used the vb-KLVQ, introduced in section 5.3.2.

The test of the complex integrative handling of mixed data against the simple Euclidean integration was done by comparing the results of using a GLVQ and the results using the vb-KLVQ on several feature group sets in parallel.

In all these tests we repeated the runs 20 times, altering the splitting between test and training samples and the order of the training sample presentation. We considered 86 patient samples: 50 patients with follow-up status one and 36 with follow-up status three, neglecting follow-up status two. These samples were split into 72 samples for training and 14 samples for testing. As we will discuss in section 7.2 we identified in first tests that the balancing of the data set can be preferable to prevent degenerated results. In later tests we additionally used a balanced setting with 60 samples for training – 30 from each class – and 12 for testing – 6 for each class.

For the outlier analyses we applied the leave-one-out validation introduced in section 4.5.2 with the vb-KLVQ for the feature group set that performed best during the integrative relevance analysis. We ran the vb-KLVQ 20 times for each left out data point. We initialized one prototype per class by a random data point from the training set.

7.1.2 Initialization, parameter setting and convergence of algorithms in our application

There are several settings necessary for applying vector quantization based methods to a specific problem. They concern the initialization, the choice of parameters and the definition of convergence. The general considerations about these settings were introduced in section 4.4. In the present section we discuss heuristics and approaches for the appropriate setting in our application domain the breast cancer research project Exprimage.

Number of prototypes

To use the prior domain knowledge for the choice of the number of prototypes reliable expert annotations are required, see section 4.4.1. The experience in our application showed that the prior knowledge in the form of pathological annotations was not reliable, for details cf. section 6.5.1. We did not use it to determine the number of prototypes. A preliminary clustering was not applied as there was not enough representative training data available to cope with the freedom of the integrated dissimilarity measure. Due to the small number of training data as a starting point we restrict our model to one prototype per follow-up status.

Initialization of prototype positions

In section 4.4.2 we introduced several initialization possibilities for the prototype positions. Due to the insufficient reliability of the prior biological knowledge in the Exprimage project we chose a non oriented version of initialization. To reduce the influence of randomness in the tests with random training and test set splitting, we used the initialization in the center of mass from the Euclidean point of view. Tests using initializations with an adequate center of mass for the different dissimilarities are outside the scope of this thesis but postponed to further improvements of the systems. In the leave-one-out validation, see section 4.5.2, there is no random training and test set splitting. We applied the initialization by random data point choice in this case.

Initialization of dissimilarity parameters

In our framework for the learning of mixed data we distinguish two types of integration introduced in section 5.1.1:

- the *vector*-based integration of dissimilarities given in equation (5.1.1) and

- the *matrix*-based integration of dissimilarities given in equation (5.1.2).

From the initialization methods proposed in section 4.4.3 we used similar values for all parameters in one run.

Learning rate for the prototype positions

The learning rate of the prototypes was adapted according to Papari's method introduced in section 4.4.4. We used the batch implementation of the method where the prototype updates are calculated and collected during one epoch but not applied directly. The collected update vector is normalized and scaled with the step size given by the learning rate. We worked with an initial learning rate of 0.5 and a reduction factor of $\frac{2}{3}$ in the case of a jump. The condition for jumping was controlled every three epochs.

Learning rate for the dissimilarity parameters

The learning rate of the dissimilarity parameters was chosen as a fifth of the prototypes' learning rate. The need for adiabatic updates is reduced as the dissimilarity is stationary over one epoch, see section 4.4.5.

Convergence

We ran the tests for 600 epochs initially. Using the visualization introduced in figure 4.1 we identified non-converged tests. In this case we repeated the 20 runs for the corresponding setting with 1200 epochs. In later tests with higher stability we reduced the number of epochs to 300.

7.1.3 Evaluation of learning results

In section 4.5 we introduced measures and approaches for the numerical evaluation of the learning results. In the tests described in this chapter we evaluated the recognition rates for the training and test set, the achieved cost function value and the Cohen's kappa between the actual labels of the samples and the predicted ones. Furthermore we evaluated the recall and the precision of the recognition for the single follow-up status classes. For the analysis of the last hypothesis we used the leave-one-out validation.

To couple the pathological experts to the results, we introduced visual feedback methods for the learned model. They were discussed in section 4.6. This coupling was necessary for the proof of the ecological validity of the results as well as for insight possibilities. In the combined analysis we used RFDDs and lDPPs to display the results of the best run for the best combination to the medial experts. This evaluation could induce further medical experiments and studies. We decided to use the non-classical multidimensional scaling as it is available in the Matlab Statistics Toolbox together with various visualization and evaluation possibilities.

7.1.4 Determining appropriate dissimilarities for the feature groups from image analysis and clinical data

The grouping of features to semantically valid feature groups as well as the determination of the dissimilarities just within these groups was induced by judgments of the human domain experts, see section 7.1.4. Grouping is suitable to reduce the complexity of the computations. It significantly reduces the number of entries in the dissimilarity matrices for relational data and with this the computational complexity of the learning calculations. The single feature groups were tested independently in the isolated relevance analysis.

Categories of feature groups

In the algorithms and the application in this thesis, we considered four different categories of feature groups. These categories are associated with properties concerning the choice of dissimilarities and the normalizations that were used for the corresponding feature group. The four feature group categories with their properties are given in table 7.1.

For the *simple numerical descriptors* the squared Euclidean distance, cf. equation (3.1.2), is a suitable dissimilarity measure. In this metric every feature dimension contributes to the overall distance with the same weight. If some feature dimensions have high variances they have a major influence on the overall distance. As this was not intended in our application, a normalization of the features was useful. We tested the values in each feature dimension whether they were distributed according to the normal distribution using the Jarque Bera test (Jarque and Bera 1980). If that was the case, the corresponding feature was normalized to fit a normal distribution with zero mean and unit standard deviation. Otherwise the values of the feature dimension were linearly scaled to be within a minus one to one range.

If the considered feature group was representing a *distribution*, we measured the dissimilarity in this group according to the Cauchy-Schwarz divergence, cf. equation (3.2.5), which is a γ-divergence with $\gamma = 1$. A normalization in the sense as for the Euclidean distance was not necessary. The values of distributions according to theory sum to one within their feature group. During learning a renormalization had to be done after every adaption step to ensure this condition.

Feature group dissimilarity	Dissimilarity measure	Initial normalization	Normalization in adaptation
Simple numerical descriptors	Squared Euclidean distance	yes	no
Distributions	Cauchy-Schwarz-divergence	no	yes
Gaussian distribution representatives (mean, standard deviation)	Kullback-Leibler for Gaussian	yes	no
Relational data	Pairwise dissimilarities	no	yes

Table 7.1: *Overview of considered feature group categories and their properties concerning the choice of dissimilarity and normalization*

The feature groups that represented *Gaussian distributions* via mean and standard deviation values were compared using the specialized Kullback-Leibler divergence given in equation (3.2.6). For this category normalization is a necessary step as this divergence is sensitive to variations in the features. Numerical instabilities can occur if the features are not normalized. A normalization that considers mean and standard deviation as independent from each other is conceptually inadequate. An adequate normalization strategy requires the normalization of the values from which the mean and standard deviation are calculated. From these normalized basic data the corresponding mean and standard deviation have to be calculated and used as adjusted features. If the feature extraction and the normalization are separated process steps this normalization is not possible. Because of the work flow design in our application we used a separate normalization of mean and standard deviation according to the Jarque Bare test described above for the numerical descriptors.

For the forth considered category, the *relational data*, pairwise dissimilarities were used for the comparison. They were either gained in the investigation of expert domain knowledge, as shown in section 7.1.4 for the process in Exprimage, or calculated from the value distribution in the data, cf. section 3.3. As relational data represents a distribution, the single feature values within a group have to sum up to one. Renormalization of the feature values during the adaptation assured this property.

Determining appropriate dissimilarities for features extracted from tissue level images

The groups for the image based features were defined by the different representations that were realized to map possibly relevant medical clues. The feature groups were categorized as numerical descriptors, distributions, representations of Gaussian distributions or relational data. The category of the feature group is related to properties concerning a suitable dissimilarity measure and normalization method as discussed in the the previous section. The mapping of the extracted feature groups in the image analyses to their corresponding category is given in table 7.2. We abbreviate the category of the groups by N for simple numerical descriptors, by D for distributions, by G for Gaussian distributions. For the relational feature groups we used R1 and R2 as abbreviation for the different statistically calculated dissimilarities and RH for the dissimilarities based on the human expert knowledge.

Determining appropriate dissimilarities for clinical data

For the clinical data that we used in the analysis, cf. section 6.3.1, there are established groups in clinical routine:

1. the tumor characterization by tumor size and number of affected lymph nodes (T, pN) that are in clinical practice grouped together with the metastases status (neglected here) to the so called TNM characterization[3] that is used for prognosis,
2. the tumor characterization by hormone receptor status (ER, PR, HER2) that is used for therapy suggestion and
3. the tumor characterization by invasion and residual tumor (L,V, R) that is used in addition to the other groups in the Sankt Gallen consensus for therapy suggestion.

The grading was considered as a singleton relational data feature. The age was also handled separately as it has a natural influence on the death rate of the patients. We used the difference as dissimilarity measure for the age.

The relevance analysis approaches realized an information aggregation according to various forms of dissimilarities. We needed dissimilarities between different constellations within the clinical data feature groups. The features were categorical. Commonly in computational approaches the dissimilarities between constellations

[3] This characterization was suggested by the Union for International Cancer Control (http://www.uicc.org/) for determining the stage of the tumor. This characterization is not specific for breast cancer. It can be used in all cancer diseases.

of categorical features are gained from the available data using statistical measures, see section 3.3.

To improve the conceptual validity of the dissimilarities and the information aggregation we determined a dissimilarity measure for the available data that uses the biomedical domain knowledge. For pathological experts the determination of dissimilarities across many cases was counter intuitive. They are used to a case based judgment of the riskiness of tumors in the context of single findings expressed in the clinical features. An extensive explanatory conversation with the pathological experts was indispensable introducing the need of the dissimilarity determination and the necessary abstract perspective respectively.

In the following section we present an attempt to mediate between the case based human expert thinking and the need for dissimilarities in the learning approaches. In the Exprimage project we conducted this dissimilarity determination approach with two pathological experts who had the possibility to discuss their contributions during the process. The experts emphasized that their contributions are vague and incomplete. With every new patient case that is handled in an enabling system the biomedical background knowledge grows. It is necessary to adapt the dissimilarities when the background knowledge indicates it[4].

The statistically gained dissimilarities model different evidence for the semantical dissimilarity than the human dissimilarity judgment based on riskiness. It would probably yield the best ecological validity to suitably combine these both dissimilarity measures. This is outside the scope of this thesis. For the moment we decided to incorporate all dissimilarity aspects by using copies of each feature group with every dissimilarity measure. The relevance analysis over all feature groups provided evidence for the adequate choice of dissimilarity for the clinical data in different constellations. In the first section of table 7.2 we summarize the clinical feature groups with the corresponding dissimilarities.

Dissimilarity determination by experts knowledge An abstract dissimilarity determination is not connectible to the pathological procedures of case based reasoning. We did not ask the domain experts to determine the dissimilarities. Rather we let them judge the riskiness of specific constellations in the clinical data. From the differences of the riskiness for different constellations we calculated the dissimilarities.

[4] In an ideal system an incrementally growing ontology of the biomedical relations with their contextual influences would be represented. It would grow in informative value by an adaptation process that results from the human evaluation and interpretation of patient's analysis in the system. This idea is far from realization as many formalizations cannot cope with the structured complexity of the real-world phenomena.

To gain the abstract riskiness judgment for the data in the feature groups we conducted a stepwise approach. For every feature within a feature group we listed the feature values that were present in the data. They were sorted according to their riskiness, see figure 7.1(a). The experts had the possibility to adjust the riskiness of specific feature values in the sorted list, see figure 7.1(b).

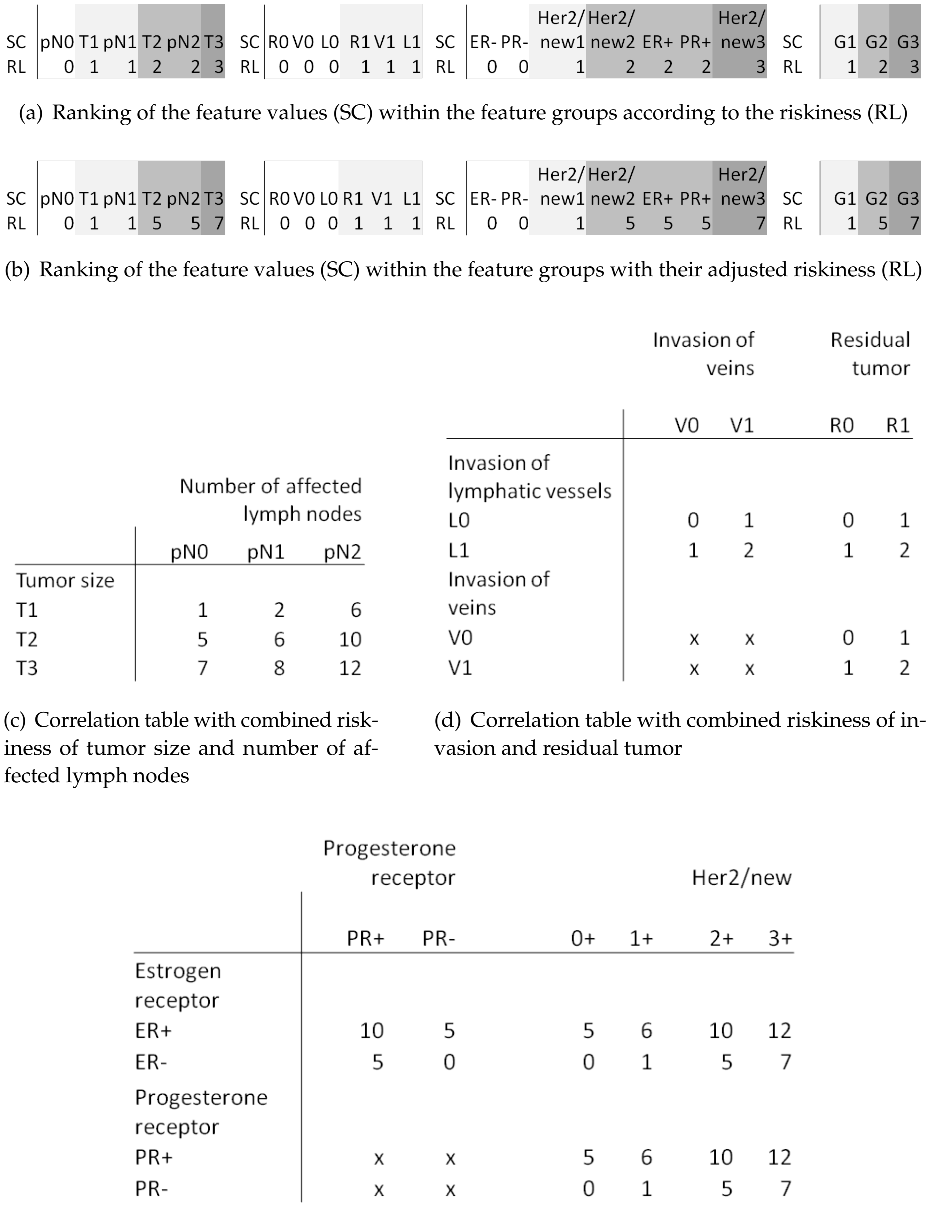
Based on this isolated riskiness judgment we asked the experts to judge combinations of two feature values within a group. For the constellations present in the data, the experts chose whether the riskiness of the feature values add or multiply or cancel each other out. The figures 7.1(c) – 7.1(e) show such correlation tables with the riskiness that the experts related to the considered combinations. In the cases where three features were evaluated for one feature group the domain experts decided to combine the risk levels of the single feature values via addition. For every feature group the dissimilarity matrix was calculated from the differences between the riskiness of the feature value constellations present in the data, see figure 7.2.

Automatically determined dissimilarities In different tests we used both approaches introduced in section 3.3 to determine the dissimilarities in the groups of categorical clinical features. We considered the combinations of categories inside the groups that were found in the training data set. This would have been inappropriate if the training data was not representative for all combinations that appear in real data.

7.2 Isolated relevance analysis

In chapter 6 we introduced a collection of potentially relevant medical clues and mapped them to computationally implemented feature groups that are summarized in table 7.2. These features were analyzed according to their relevance for the overall tumor representation in the discrimination of different cancer subtypes correlating to the known follow-up status of the patients. We introduced the preliminary selection step by an isolated relevance analysis as the number of patient samples is small and the results of an overall integration using all implemented feature groups may not lead to generalizable results.

In this isolated relevance analysis we tested the single features according to their discriminative power in the prediction of the follow-up status of the patients. We used the results of this analysis to guide the selection process of candidate sets according to medical and computational aspects, cf. section 7.2.1. The candidate sets formed a multi-layer model of the patient's tumor situation. The relevance of the feature groups within the multi-layer model was analyzed integrally, cf. section 7.3.



																	Her2/	Her2/			Her2/				
SC	pN0	T1	pN1	T2	pN2	T3	SC	R0	V0	L0	R1	V1	L1	SC	ER-	PR-	new1	new2	ER+	PR+	new3	SC	G1	G2	G3
RL	0	1	1	2	2	3	RL	0	0	0	1	1	1	RL	0	0	1	2	2	2	3	RL	1	2	3

(a) Ranking of the feature values (SC) within the feature groups according to the riskiness (RL)

																	Her2/	Her2/			Her2/				
SC	pN0	T1	pN1	T2	pN2	T3	SC	R0	V0	L0	R1	V1	L1	SC	ER-	PR-	new1	new2	ER+	PR+	new3	SC	G1	G2	G3
RL	0	1	1	5	5	7	RL	0	0	0	1	1	1	RL	0	0	1	5	5	5	7	RL	1	5	7

(b) Ranking of the feature values (SC) within the feature groups with their adjusted riskiness (RL)

	Number of affected lymph nodes		
	pN0	pN1	pN2
Tumor size			
T1	1	2	6
T2	5	6	10
T3	7	8	12

(c) Correlation table with combined riskiness of tumor size and number of affected lymph nodes

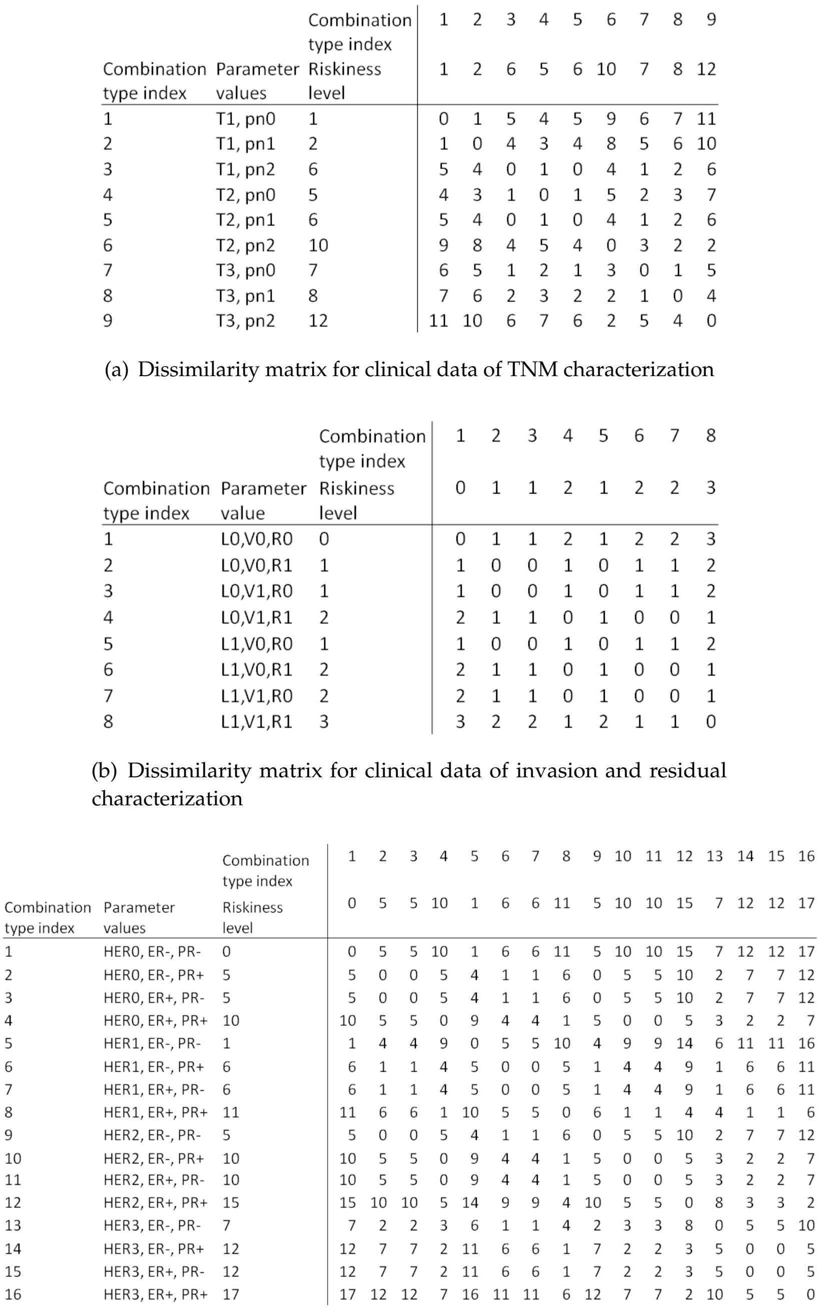
	Invasion of veins		Residual tumor	
	V0	V1	R0	R1
Invasion of lymphatic vessels				
L0	0	1	0	1
L1	1	2	1	2
Invasion of veins				
V0	x	x	0	1
V1	x	x	1	2

(d) Correlation table with combined riskiness of invasion and residual tumor

	Progesterone receptor		Her2/new			
	PR+	PR-	0+	1+	2+	3+
Estrogen receptor						
ER+	10	5	5	6	10	12
ER-	5	0	0	1	5	7
Progesterone receptor						
PR+	x	x	5	6	10	12
PR-	x	x	0	1	5	7

(e) Correlation table with combined riskiness of hormone receptor status

Figure 7.1: *Determination of medically motivated dissimilarities in Exprimage.*



		Combination type index	1	2	3	4	5	6	7	8	9
Combination type index	Parameter values	Riskiness level	1	2	6	5	6	10	7	8	12
1	T1, pn0	1	0	1	5	4	5	9	6	7	11
2	T1, pn1	2	1	0	4	3	4	8	5	6	10
3	T1, pn2	6	5	4	0	1	0	4	1	2	6
4	T2, pn0	5	4	3	1	0	1	5	2	3	7
5	T2, pn1	6	5	4	0	1	0	4	1	2	6
6	T2, pn2	10	9	8	4	5	4	0	3	2	2
7	T3, pn0	7	6	5	1	2	1	3	0	1	5
8	T3, pn1	8	7	6	2	3	2	2	1	0	4
9	T3, pn2	12	11	10	6	7	6	2	5	4	0

(a) Dissimilarity matrix for clinical data of TNM characterization

		Combination type index	1	2	3	4	5	6	7	8
Combination type index	Parameter value	Riskiness level	0	1	1	2	1	2	2	3
1	L0,V0,R0	0	0	1	1	2	1	2	2	3
2	L0,V0,R1	1	1	0	0	1	0	1	1	2
3	L0,V1,R0	1	1	0	0	1	0	1	1	2
4	L0,V1,R1	2	2	1	1	0	1	0	0	1
5	L1,V0,R0	1	1	0	0	1	0	1	1	2
6	L1,V0,R1	2	2	1	1	0	1	0	0	1
7	L1,V1,R0	2	2	1	1	0	1	0	0	1
8	L1,V1,R1	3	3	2	2	1	2	1	1	0

(b) Dissimilarity matrix for clinical data of invasion and residual characterization

		Combination type index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Combination type index	Parameter values	Riskiness level	0	5	5	10	1	6	6	11	5	10	10	15	7	12	12	17
1	HER0, ER-, PR-	0	0	5	5	10	1	6	6	11	5	10	10	15	7	12	12	17
2	HER0, ER-, PR+	5	5	0	0	5	4	1	1	6	0	5	5	10	2	7	7	12
3	HER0, ER+, PR-	5	5	0	0	5	4	1	1	6	0	5	5	10	2	7	7	12
4	HER0, ER+, PR+	10	10	5	5	0	9	4	4	1	5	0	0	5	3	2	2	7
5	HER1, ER-, PR-	1	1	4	4	9	0	5	5	10	4	9	9	14	6	11	11	16
6	HER1, ER-, PR+	6	6	1	1	4	5	0	0	5	1	4	4	9	1	6	6	11
7	HER1, ER+, PR-	6	6	1	1	4	5	0	0	5	1	4	4	9	1	6	6	11
8	HER1, ER+, PR+	11	11	6	6	1	10	5	5	0	6	1	1	4	4	1	1	6
9	HER2, ER-, PR-	5	5	0	0	5	4	1	1	6	0	5	5	10	2	7	7	12
10	HER2, ER-, PR+	10	10	5	5	0	9	4	4	1	5	0	0	5	3	2	2	7
11	HER2, ER+, PR-	10	10	5	5	0	9	4	4	1	5	0	0	5	3	2	2	7
12	HER2, ER+, PR+	15	15	10	10	5	14	9	9	4	10	5	5	0	8	3	3	2
13	HER3, ER-, PR-	7	7	2	2	3	6	1	1	4	2	3	3	8	0	5	5	10
14	HER3, ER-, PR+	12	12	7	7	2	11	6	6	1	7	2	2	3	5	0	0	5
15	HER3, ER+, PR-	12	12	7	7	2	11	6	6	1	7	2	2	3	5	0	0	5
16	HER3, ER+, PR+	17	17	12	12	7	16	11	11	6	12	7	7	2	10	5	5	0

(c) Dissimilarity matrix for clinical data of hormone receptor status characterization

Figure 7.2: *Resulting medically motivated dissimilarity measures for clinical data in Exprimage.*

7.2.1 Overview over implemented feature groups

The whole variety of feature groups that were implemented to represent the clinical clues is given in table 7.2. For every feature group we give the full name as well as an abbreviation that we will use in the discussion of the tests in the next sections. At the right side of the table in the left column the type of the feature groups is given, cf. section 7.1.4. We abbreviate numerical descriptors by N, representatives of Gaussian distributions by G and representatives of distributions by D. For the relational feature groups we used different types related to the dissimilarities mentioned in the previous section: R1 and R2 for the different statistically calculated dissimilarities and RH for the dissimilarities based on the human expert knowledge. In the right most column we show the dimensionality of the corresponding feature group that gives the number of features within the group.

Clinical data			
Feature group full name	Abbreviation	Type	Dim
TN characterization of the tumor	Clinical TN	R1	7
		R2	7
		RH	9
LVR characterization of the tumor	Clinical LVR	R1	5
		R2	5
		RH	8
Hormone receptor characterization of the tumor	Hormonereceptors	R1	8
		R2	8
		RH	16
Age of the patient at surgery	Age	N	1
Grading	Grading	R1	3
		R2	3
		RH	3

Basic quantification of different tissues in a patient's probe			
Feature group full name	Abbreviation	Type	Dim
Absolute tissue area	AbsoluteArea	N	2
Relative area stroma to overall tumor	RelAreaStroma	N	1
Size variation of tumor regions	RegionSize	G	2
Perimeter variation of tumor regions	RegionPerimeter	G	2
Number of AE1AE3 tumor regions	NumberOfIslands	N	1

Mean area of AE1AE3 tumor regions to tumor area	MeanAreaToTumor	N	1

Structural heterogeneity characterization: inner growth structures

Feature group full name	Abbreviation	Type	Dim
Distribution of inner tumor structure	InnerTumorStructure	D	5
Number of regions of different inner tumor structures	ClusterRegNumber	N	5
Area distribution for cluster one	AreaRegionsCluster1	G	2
Area distribution for cluster two	AreaRegionsCluster2	G	2
Area distribution for cluster three	AreaRegionsCluster3	G	2
Area distribution for cluster four	AreaRegionsCluster4	G	2
Area distribution for cluster five	AreaRegionsCluster5	G	2

Functional heterogeneity characterization: Co-occurrence analysis of functional and structural information based on pixel statistics

Feature group full name	Abbreviation	Type	Dim
Relative area of functional marker to tumor parenchyma	RelativeAreaToTumor	N	3
CD45 distribution in tissue types	CD45inTissue	D	4
ER distribution in tissue types	ERinTissue	D	4
PR distribution in tissue types	PRinTissue	D	4
CD45 co-occurrence with other functional markers	CD45co-occurrence	D	3
ER co-occurrence with other functional markers	ERco-occurrence	D	3
PR co-occurrence with other functional markers	PRco-occurrence	D	3

Functional heterogeneity characterization: Co-occurrence analysis of functional and structural information based on connected components

Feature group full name	Abbreviation	Type	Dim
Number of regions	NumberRegions	N	3
Area distribution for tumor regions	AreaRegionsTum	G	2
Area distribution for ER positive regions	AreaRegionsER	G	2

Area distribution for PR positive regions	AreaRegionsPR	G	2
Number of tumor regions covered by hormone receptors	HRTumorRegions	N	2
Functional heterogeneity characterization: Co-occurrence analysis of functional and structural information based on geometric constellations			
Feature group full name	Abbreviation	Type	Dim
Spatial distribution of CD45 in tumor regions	DistributionCD45	D	3
Spatial distribution of ER in tumor regions	DistributionER	D	3
Spatial distribution of PR in tumor regions	DistributionPR	D	3
Structural tumor distribution pattern characterization: graph based representation			
Feature group full name	Abbreviation	Type	Dim
Mean and standard deviation of the edge lengths in MST	MSTDist1	G	2
Variation coefficient of the edge lengths and relation from minimal to maximal edge lengths in MST	MSTDist2	N	2
Average weighted node degree in MST	MSTWeightedDeg	N	1
Number of nodes in MST	MSTnNodes	N	1
Randić index in MST	MSTRandicIndex	N	1
Distribution of the node degrees in MST	MSTDeg1	G	2
Variation coefficient of the node degree and relation from minimal to maximal node degrees in MST	MSTDeg2	N	2
Mean and standard deviation of the edge lengths in DG	DGDist1	G	2

Variation coefficient of the edge lengths and relation from minimal to maximal edge lengths in DG	DGDist2	N	2
Average weighted node degree in DG	DGWeightedDeg	N	1
Number of nodes in DG	DGnNodes	N	1
Cyclomatic number in DG	DGCyclNumber	N	1
Randić index in DG	DGRandicIndex	N	1
Distribution of the node degrees in DG	DGDeg1	G	2
Variation coefficient of the node degree and relation from minimal to maximal node degrees in DG	DGDeg2	N	2

Structural tumor distribution pattern characterization: morphometric representation

Feature group full name	Abbreviation	Type	Dim
Morphometric clustering on SOM regions using two clusters	SOMnCl2	D	2
Morphometric clustering on SOM regions using three clusters	SOMnCl3	D	3
Morphometric clustering on SOM regions using four clusters	SOMnCl4	D	4
Morphometric clustering on SOM regions using seven clusters	SOMnCl7	D	7
Morphometric clustering on AE1AE3 regions using two clusters	MoAE1AE3nCl2	D	2
Morphometric clustering on AE1AE3 regions using three clusters	MoAE1AE3nCl3	D	3

Morphometric clustering on AE1AE3 regions using four clusters	MoAE1AE3nCl4	D	4

Functional tumor distribution pattern characterization

Feature group full name	Abbreviation	Type	Dim
Ratio CD45 to AE1AE3	CD45Ratio	N	1
Ratio ER to AE1AE3	ERRatio	N	1
Ratio PR to AE1AE3	PRRatio	N	1
Distribution of RCC8 relations for CD45	CD45RCC8	D	7
Distribution of RCC8 relations for ER	ERRCC8	D	7
Distribution of RCC8 relations for PR	PRRCC8	D	7
Linear Distance Quantification for CD45	CD45LDQ	D	2
Linear Distance Quantification for ER	ERLDQ	D	2
Linear Distance Quantification for PR	PRLDQ	D	2

Table 7.2: Overview over all feature groups that we considered for the development of a multi-layer model for breast cancer in Exprimage

Tests of discriminative power for single feature groups

For the selection of single feature groups we tested the discriminative power of the single feature groups with respect to the follow-up status in the KLVQ for the relational feature groups and in the GLVQ for all others. We used the settings discussed in section 7.1.2.

First series of tests We conducted the twenty runs for each of the feature groups with an unbalanced setting in the training and test data. The learning was performed over 600 epochs. We evaluated the test recognition rate as well as precision and recall for all test data. These values are not comparable to the clinical evaluation values as introduced in section 6.3 because we include here the grading two. Restricting the measures to the test data yields a more valid comparison between the single test series concerning the generalization ability.

The results exhibited the following phenomena:

- In all feature groups representing Gaussian distributions there were runs that did not properly converge. They showed sustained alternations in the training and/or the test recognition rate. This was also the case for the SOMncl4 feature groups that represented a distribution.

- In the recall and precision values for the classes we saw that for all feature groups of numerical descriptors the system tended to classify the data to follow-up status one. This was in accordance with the prior distribution of the data samples in the classes. The evaluation of the prototypes showed that the prototype of follow-up status three was pushed out of the data space. A similar phenomenon was observed for some distribution representing feature groups. Due to the renormalization of the prototypes' feature values after the adaption these prototypes were pushed to the border of the data space and did not probably represent the follow-up status three.

For the feature groups that did not constantly show proper convergence we repeated the tests with 20 trials that were run 1200 epochs. That yielded constant proper convergence for two feature groups: the SOMncl4 and the AreaRegionsPR. For all other feature groups the convergence was not improved. Papari's method for learning rate control did not intervene in this behavior. Another mechanism is necessary to identify alternating behavior and force the decrease of the learning rate. We expected that this instability would not occur in combined feature group settings.

The second phenomenon is known in learning and data mining, see e.g. (Chawla 2005) for an overview on data mining for imbalanced data sets. To cope with this phenomenon we changed the setting to balanced training and test sets. The training data set comprised 30 samples for each follow-up status and the test set 6 samples each. As this also influenced the convergence properties we repeated the tests for all feature groups with this balanced setting.

Second series of tests The settings for the second test series were identical to the first test series except for the balancing of the data set. We performed 600 learning epochs for every run in all feature groups. There was no additional learning rate control included in these tests.

The runs for the Gaussian distribution representatives converged significantly more often than in the unbalanced setting. For all these feature groups there were enough converged runs to estimate their discriminative power, at least 16 out of 20 converged. All other feature groups converged reliably during the 600 epochs.

Feature group	Average test recognition rate	Standard deviation of recognition rate
Grading RH	66.7%	14.0%
SOMnCl3	65.0%	13.1%
Grading R1	66.7%	16.2%
SOMnCl4	60.0%	10.7%
Clinical TN RH	55.8%	6.7%
Clinical LVR RH	57.9%	9.9%
ERRatio	50.8%	3.7%
Grading R2	60.0%	13.1%
CD45co-occurrence	52.1%	5.3%
Hormonereceptors R1	50.8%	4.6%

Table 7.3: *Overview of results for best ten feature group categories in isolated relevance analysis*

For the numerical descriptors the representation of class three was improved. This was shown by higher recognition rates in general and higher recall values for class three. In recall and precision for class one and three we still saw a tendency of the learned models to classify the data as class one.

The second test series yielded enough suitable information for the selection of feature group combinations for the integrated analysis. Table 7.3 as an example shows the best ten feature groups according to their discriminative power measured by the average test recognition rate and its standard deviation. The feature groups were sorted according to the average test recognition rate minus its standard deviation. The overview over all feature groups can be found in table C.1 in the appendix.

The results for the clinical data feature groups showed that except for the hormone receptor feature group the dissimilarities gained using expert knowledge perform throughout better in terms of the average recognition rates. Partially the experts′ dissimilarities also yielded more stable results than the statistic ones.

Selection of candidate feature group combinations under different aspects

We chose the candidate feature group combinations from the set of feature groups according to different aspects:

pathological selection We selected the feature groups together with the pathological experts according to the medical clues to be covered in the description of breast cancer.

computational selection We selected the feature groups according to the best recognition rates in the isolated analysis. For semantically similar feature groups we skipped worse recognition rates for the benefit of other feature groups. This was done in accordance to the pathological experts' definition of semantic similarity.

random selection We randomly draw a preselected number of feature groups from the whole set. There was no previous orientation in this selection.

no selection For these test runs we used all available feature groups without any selection.

We started with the pathological selection. From these considerations we also got the magnitude for the number of feature groups to be combined. We used this number in all other selections.

Pathological selection The basic idea for the pathological selection was to cover every field of medical clues that was represented by our image analysis, see section 6.5. We selected the feature groups from the single fields according to their discriminative power in the isolated relevance analysis. We subsumed the functional heterogeneity and the functional distribution pattern as they are semantically close. From the pathological point of view the feature groups for the distribution pattern description are more promising for the breast cancer representation. We put more weight on them. Additionally the age was chosen as it is expected to give a context for the other feature groups.

This yielded the following assignment of the selected feature groups to the medical clues – **P1**:

Structural heterogeneity: AreaRegionsCluster5

Functional heterogeneity & distribution

- **for CD45:** CD45co-occurrence
- **for ER:** ERRatio
- **for PR:** PRLVQ

Structural distribution pattern

- **by graphs for macro structure:** MSTDist1
- **by graphs for micro structure:** DGDeg2
- **by morphometry for macro structure:** SOMnCl3
- **by morphometry for micro structure:** MoAE1AE3nCl4

Clinical data: Age

We selected nine feature groups for the integrative relevance analysis. Four of them were distributions, three numerical descriptors and two Gaussian distribution representatives.

Computational selection As there were nine feature groups selected according to the pathological evidence we also selected nine feature groups according to computational considerations. Starting with the most discriminative one according to the difference of the average test recognition minus its standard deviation. In the question which feature group to choose next a precise way is to calculate the correspondence between the classification given by the first feature group and the one analyzed next. For this purpose the Cohen's kappa introduced in section 4.5.2 is a suitable measure of correspondence. For a comprehensive candidate selection it is pertinent to choose feature groups with high recognition rates and low values for the Cohen's kappa as they are expected to model different aspects of the classification tasks. If already a selection of more than one feature group was chosen the classification correspondence can be measured using the Fleiss' kappa, see section 4.5.2. This precise selection approach is computationally expensive.

In our application we conducted a pragmatic selection process according to the semantic interpretation of the feature groups. We only chose feature groups that according to the pathological experts were semantically different from the ones before. This resulted in the following feature group selection – **C1**:

- Grading RH
- SOMnCl3
- Clinical TN RH
- Clinical LVR RH
- ERRatio
- CD45co-occurrence
- Hormonereceptors R1
- AreaRegionCluster5
- RegionsPerimeter

There were four relational feature groups, two distributions, two Gaussian distribution representatives and one numerical descriptor in the selected set.

Random selection To randomly select nine feature groups from the whole data set we used matlab's built-in function *"randperm"* for the feature group set size of 72. The first nine indices were chosen for the resulting feature group selection – **R1**:

- Clinical LVR RH
- CD45co-occurrence
- DistributionPR
- AreaRegionCluster4
- AreaRegionCluster1
- MSTDist1
- CD45inTissue
- AreaRegionCluster3
- ERLDQ

The set comprised one relational feature group, four distributions and four Gaussian distribution representatives.

7.3 Integrative relevance analysis – Application of algorithms from the introduced framework to the Exprimage data set

The aim of this thesis is not primarily the identification of medically relevant findings or the advanced development of the algorithms or the best discriminating result. More patient cases, comprehensive therapy information and a closer interaction with the pathological experts are necessary for these research aims. Rather we show the principle approaches for the interleaving integrative relevance analysis on the example of follow-up status prediction.

We focus on the experimental application of the algorithms together with the evaluation of the tendency of the results. The preliminary aim is to show an approach to a relevant contextual feature group combination. We conducted the necessary reduction process according to the relevance values identified by the system using different incorporation strategies for statistical and pathological evidence. Checking carefully whether dropping of feature groups improved and stabilized the results we tried to find a suitable incorporation strategy. For that reason this section is structured to give a process-oriented presentation of interim results and the reflection of their consequences in addition to the general description.

In the following section we will mention the results of tests rather than interpret them in detail. This discussion is given in chapter 8.

7.3.1 Discussion of the interpretability of the relevance values

In the following tests we interpret the dissimilarity parameter vector $\vec{\alpha}$ that is adapted during the learning process of the vb-KLVQ (see section 5.3.2) as a vector of relevance values for the corresponding feature groups. It estimates their relevance for the discrimination of different cancer subtypes. This interpretation is valid if the following conditions are met:

- The dissimilarities in the single feature groups are comparable to each other. That means that if the dissimilarity value is mathematically equal in two feature groups the dissimilarity of the underlying feature group values is conceptually equal.
- The dissimilarities in the single feature groups vary in a similar range. That means that the variance of the dissimilarities is about the same in the different feature groups.

These conditions are not commonly fulfilled in the application to biomedical data with a range of different dissimilarity measures. The verification of the conditions is no trivial task as it needs to mediate between the mathematical and the conceptual validity. This touches the issues of finding medically valid dissimilarities that we discussed in section 7.1.4.

To meet the considered conditions as good as possible in our tests, we normalized the dissimilarities for every feature group. We determined the median and the interquartile range of the pairwise dissimilarities in the single feature groups for the training data in every test run. We used these values to normalize all feature group specific dissimilarity values in the approach. The detailed mathematical and medical analysis whether this normalization was able to answer the purpose is outside the scope of this thesis. Throughout the next sections we assume that the introduced normalization procedure assured the fulfillment of the conditions and that the dissimilarity parameters can be interpreted as relevance values.

7.3.2 First test series using candidate feature group combinations

For the first tests on feature group combinations we used the principle settings described in section 7.1.2 except for the update of the prototypes. We changed this procedure such that the winning correct prototype was attracted ten times stronger to the presented data point than the winning wrong prototype was repelled. We assumed that this mechanism is capable of compensating the influence of an unbalanced data set. We used the whole data for follow-up status one and three without balancing.

For the evaluation of the results we determined the test recognition rate as well as the recall and precision of the classes for the test data set. In addition, we calculated these measures on all data that had grading values one or three. For this evaluation we did not distinguish between training and test set. Using only patient samples with grading values one and three of the test set would not have been enough data for a valid estimation. The described prediction evaluation measures are better comparable to the clinical prediction measures given in section 6.3. We refer to these values as clinical recognition rate, clinical recall and clinical precision.

We ran 20 tests for each of the selected combinations of feature groups introduced in section 7.2.1. We applied Papari's approach for learning rate control but no additional control to force a decrease in the case of alternating behavior. For the combinations all the tests converged properly during less than 600 epochs. We reduced the number of epochs to 300. In the random choice there were several feature groups that did not show proper convergence in the isolated analysis. In the combination there was proper convergence for every test run. The combination of the instable feature groups with others appeared to stabilize the learning process.

Table D.1 in the appendix summarizes the results of this first test series. There was no clear best model in these selections. For the different evaluation measures test recognition rate, test Cohen's kappa, clinical recognition rate and training recognition rate different selections scored highest. According to the test recognition rate the whole feature group set had the best prediction ability for the follow-up status with an average of 59.6% and a standard deviation of 4.8%. For comparison, the test recognition rate of the trivial case classifying all data to the dominant class one would be 57.1%.

Except for the random choice in all feature group combinations the recall and precision values revealed the tendency of the system to classify samples as class one. This tendency was the same for the clinical configuration. The clinical recall and precision values that were given by the grading for our data set showed an opposing tendency, see section 6.3. There class three was preferred.

For all feature group combinations the average recognition rate for the test set over ten runs was worse than the average for the training set. The generalization ability of the learned models was limited. We reduced the single feature group selections to decrease the number of free variables and potentially increase the generalization ability.

We conducted a relevance ranking analysis to determine the feature groups that were selected out of the feature group combinations. There are different possibilities for the relevance ranking, e.g. determining the average relevance value over all runs for every feature group and rank them accordingly. This approach is suitable if the relevance values are stable.

In our application the relevance values for the single runs were variable. To reduce this unsteadiness, we used a statistic over the ranking of the feature group relevances in the different runs. We ranked the feature groups for every run according to their relevance value, i.e. the dissimilarity parameter in the vb-KLVQ. We summed up the inverse rank number for the feature groups over all runs. If a feature group out of a set of nine feature groups constantly won the relevance ranking it got an overall ranking score of $\text{nrTrials} \cdot \text{highestInverseRankingNumber} = 20 \cdot 9 = 180$. In contrast if it always was least relevant the overall ranking score was $20 \cdot 1 = 20$.

We analyzed the ranking scores for every feature group within one selection. The differences between the ranking scores were used to determine the cut off for the reduction to the most relevant features. For example in the computational feature group selection the ranking scores were given by the values shown in table 7.4. We used the highest difference as identification point for the cut off. Feature groups that scored lower than the one above the cut-off were dropped. In this example the cut-off was between the ERRatio and the AreaRegionCluster5 feature groups. We dropped the AreaRegionCluster5, the CD45co-occurrence and the RegionsPerimeter feature groups for the next test series.

With the described relevance ranking score analysis we yielded the following reduced feature group selections:

Pathological selection – P2

- PRLDQ
- DGDeg2
- SOMnCl3
- MSTDist1

Computational selection – C2

- Grading RH
- SOMnCl3
- Clinical TN RH
- Clinical LVR RH
- ERRatio

Feature group	Ranking score	Difference to next feature group in ranking score
Clinical LVR RH	168	18
Grading RH	150	10
Clinical TN RH	140	35
SOMnCl3	105	1
Hormonereceptors R1	104	12
ERRatio	92	41
AreaRegionCluster5	51	5
CD45co-occurrence	46	2
RegionsPerimeter	44	

Table 7.4: *Relevance ranking score for the computational feature group selection. The minimum achievable ranking score was* 20*, the maximum* 180*. The cut-off for the selection of feature groups for subsequent tests was the difference of* 41 *in the ranking score between the ERRatio and the AreaRegionCluster5 feature group.*

- Hormonereceptors R1

Random selection – R2

- Clinical LVR RH
- DistributionPR

In the reduced pathological selection there are three distributions and one numerical descriptor. The reduced computational selection comprised four relational feature groups, one distribution and one numerical descriptor. In the reduced random selection there was one relational feature group and one distribution.

For the computational selection the highest scoring feature groups were in accordance with the results of the isolated relevance analysis. In the random selection the DistributionPR scored significantly better in combination with the Clinical LVR RH than the CD45co-occurrence (rank 2 to rank 6) which was the other way round in the isolated analysis (rank 9 to rank 19). In the pathological selection the four feature groups that scored best in the isolated relevance analysis were dropped according to their relevance score in the integrated analysis.

In the feature group set comprising all implemented feature groups there was no clear cut-off possible. In accordance with the feature group selection based on the isolated relevance analysis we used the nine best scoring feature groups for subsequent tests:

No selection – N2

- RelAreaStroma
- Hormonereceptors R2
- MSTDist2
- MSTWeightedDeg
- DGDist2
- DGDeg2
- DGWeightedDeg
- CD45Ratio
- ERRatio

As the systems showed the tendency to classify the samples to class one we changed the setting of the tests to the balanced data sets and repeated the trials for the basic feature group selections. Table D.2 in the appendix shows the test results in detail. For all selections except the random one the test recognition rate was smaller in the balanced setting. For the test recognition rates the pathological selection scored highest with an average of 56.3%. The quality of this result was limited as the standard deviation was 12.1%.

We applied the same relevance ranking score based reduction approach as described before to the results of the balanced tests. For the random and the pathological selection the chosen feature groups were the same as in the unbalanced tests. For the computational selection the results of the balanced runs suggested to additionally drop the ERRatio. This resulted in the following selection:

Computational selection – C3

- Grading RH
- SOMnCl3
- Clinical TN RH
- Clinical LVR RH
- Hormonereceptors R1

For the whole feature group set (no selection) the relevance ranking score results differed significantly. The Hormonreceptors R2 feature group was the only one that was under the nine best scoring feature groups for both test settings. The reduced feature groups selection from the whole feature group set according to the balanced tests was:

No selection – N4

- DistributionER
- DistributionPR
- Clinical TN R2
- Clinical LVR R1
- Hormonereceptors R2
- Grading R1
- Grading R2
- SOMnCl4
- DistributionCD45

7.3.3 Subsequent test series for the reduction of candidate feature group combinations

Using the reduced feature group sets from the first test series we conducted further tests to reduce these feature group selections. In every stage we ran balanced as well as unbalanced tests and applied the relevance ranking score approach to further reduce the selections. For some tests the described reduction was not possible as the highest score difference was between the first and the second feature group or between the last two. We used the second largest difference instead for the determination of the cut-off if it was approximately the same size as the largest difference. The reduction of the feature group selections stopped if only two feature groups were left or if the cut-off could not be determined according to the given procedure.

For the evaluation of the results all measures mentioned before were calculated. For sake of clarity we focus the following result discussion on the test recognition rate and the clinical recognition rate for the judgment of the generalization ability of the systems. For every series of tests we further analyzed the run with the best test recognition rate. For these runs additionally the Non-Euclidean Coefficient, as given in equation (3.4.3) in section 3.4.2, was calculated. Basis for this calculation was a matrix of dissimilarities that was build as data dissimilarity matrix from the dissimilarity measure that was result of the dissimilarity adaptation in the vb-KLVQ.

Reduction results for the pathological selection

Figure 7.3 shows the reduction process for the pathological selection. The feature group selections of the corresponding reduction steps are shown in the rectangles

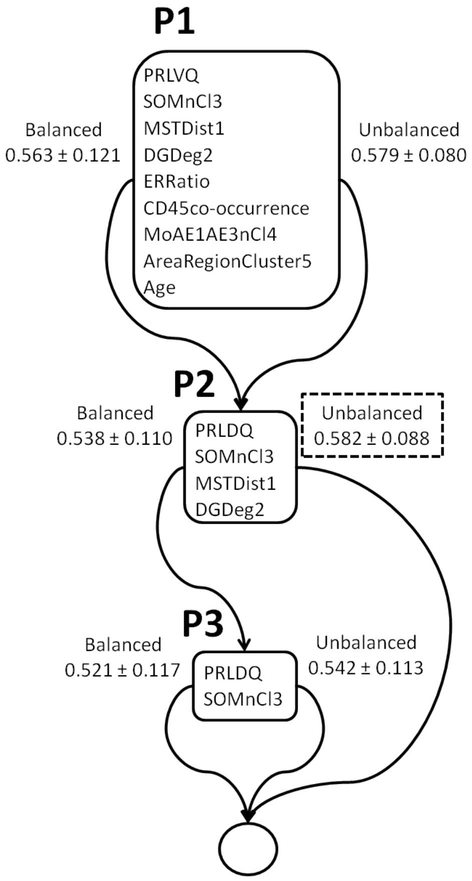


Figure 7.3: *Reduction process for the pathological feature group selection.*

with the rounded corners. For every test series the average test recognition rate over twenty runs is given with the standard deviation. The best recognition rate for the pathological selection is highlighted with a dashed outline. At the left side of the feature group selection the results for the balanced setting are given and at the right side the ones for the unbalanced setting. The arrows point towards the reduced set that is calculated from the corresponding relevance values using the relevance ranking score approach. The circle depicts the termination of the reduction process. In table D.3 in the appendix we give the detailed learning results for the reduction process.

The reduction in the first test series yielded the same reduction for the balanced and the unbalanced setting. The generalization results for the unbalanced setting in the first test series yielded a higher average recognition rate with a lower standard deviation than the for balanced setting. In the second test series the best recognition rate was achieved in the unbalanced setting. For these test runs no further reduction of the set was identifiable. In the balanced setting the results were worse than before the reduction. The subsequent reduction identified in these test runs did not lead to an improvement of the generalization ability.

There was no clear pattern in the reduction of the feature group selection for example in the type of the feature groups. From the pathological point of view no underlying concept for the reduction was identified. It is probable that in the initial pathological selection relevant feature groups were missed.

Reduction results for the computational selection

In figure 7.4 the reduction process for the computational selection is depicted. We use the same symbols as described before for figure 7.3. Additionally, we highlight the generalization result that yields a better test recognition rate than random classification by a gray background. We give the detailed learning results in table D.4 in the appendix. For the first test series the resulting reduction was similar for the balanced and the unbalanced settings with the exception that the ERRatio feature group was dropped under the balancing. With both reductions the recognition rates improved. For the reduction resulting from the unbalanced setting, higher average recognition rates and lower standard deviations were achieved. When analyzed in a balanced setting this feature group selection yielded the best generalization results compared to all other tests. Taking into account the standard deviation the test recognition rate was higher than random classification. This result showed a higher clinical recognition rate than the grading for the current prognosis discussed in section 6.3.

A further reduction as identified in the discussed balanced test led to the same reduced set as the balanced first test series. Subsequent reduction to a feature group selection of two clinical data feature groups showed no improvement. Either the model complexity fell below the necessary extend or relevant feature groups were dropped. That last phenomenon occurs if all features in the selection are relevant. As the relevance values are relative to each other, the normalization of the dissimilarity parameters can lead to low relevance values for relevant features.

Reduction results for the random selection

Figure 7.5 shows the reduction process for the random selection. The detailed learning results are given in table D.5 in the appendix. The balanced as well as the

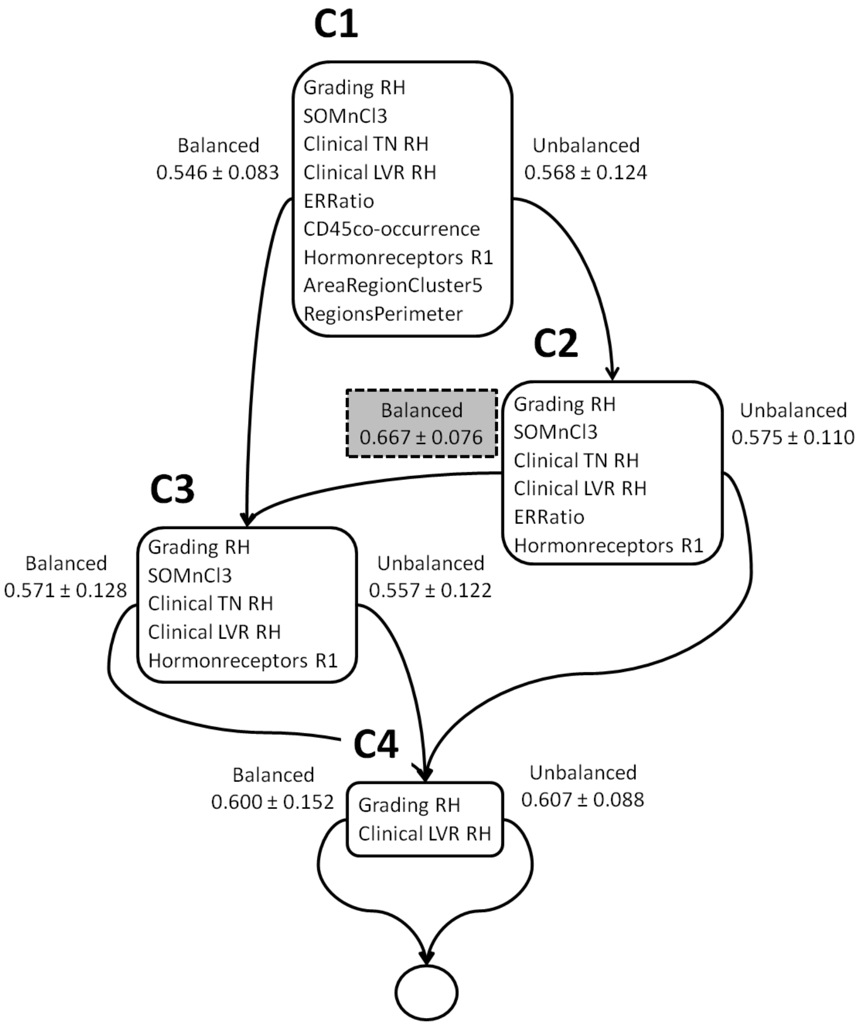


Figure 7.4: *Reduction process for the computational feature group selection.*

unbalanced test settings led to the same reduction in the two possible reduction steps. For both reduction steps the balanced setting yielded better average recognition rates over the twenty test runs but with larger standard deviations than for the unbalanced settings. In the reduction process an improvement of the generalization ability was achieved. The second reduction in the balanced setting resulted in the best test recognition rate for the random selection. Taking the standard deviation into account,

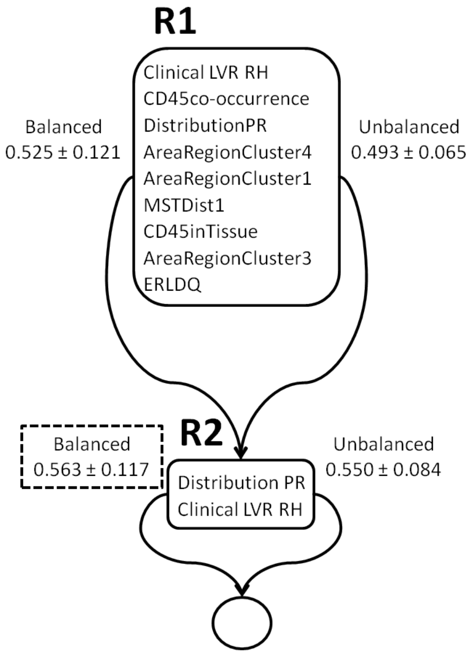


Figure 7.5: *Reduction process for the random feature group selection.*

none of the results was better than random classification or the trivial classification according to the classes' prior distribution.

Reduction results for the whole feature group set

In figure 7.6 we depict the reduction process for the whole feature group set. Both reductions according to the first test series yield a decrease in the mean test recognition rate. Analyzing the feature group set resulting from the unbalanced first test series in the unbalanced test runs ended with a trivial system classifying all data to class one.

Table D.6 in the appendix shows the detailed learning results. The reduction resulting from the unbalanced setting yielded no good results all over the reduction process. With further reduction the generalization ability got worse.

The reduction process building on the results of the balanced setting in the first test series showed a continuous improvement in the generalization ability of the system. The test recognition rate increased in every subsequent reduction step starting from the second reduction for both balanced and unbalanced settings. The standard deviation did not show a steady improvement. It varied without recognizable pattern.

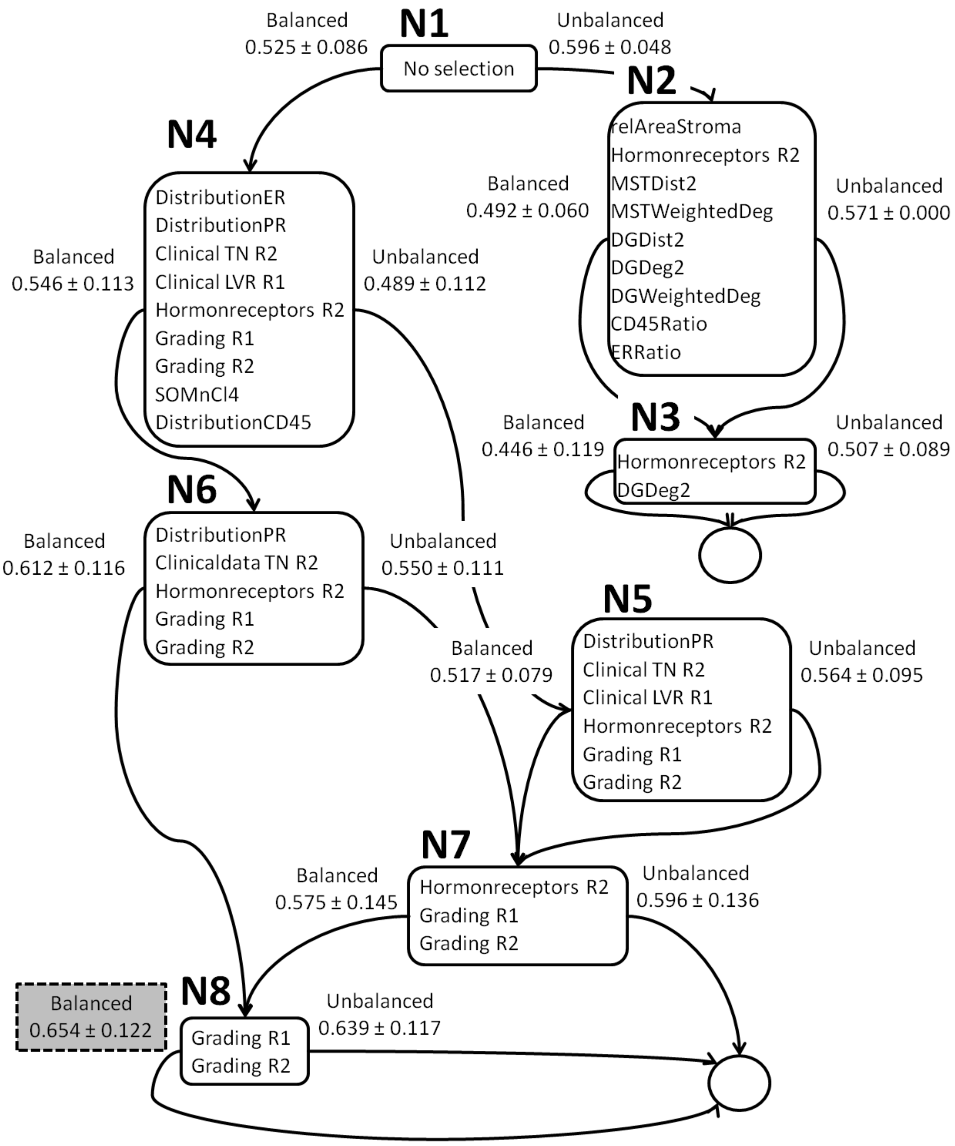


Figure 7.6: *Reduction process for the whole feature group set.*

In tendency the test with the balanced settings yielded slightly better recognition rates than the ones with the unbalanced settings. The best result was achieved in the balanced setting for the maximum reduced feature group set: Grading R1 and Grading R2. With an average recognition rate of 65.4% and a standard deviation of 12.2% it was better than random classification.

Tendencies in all the reduction results

In the reduction processes there was a tendency to improvements in the test and clinical recognition rates. Often the improvement in the average recognition rate was associated to an increase in the standard deviation of the corresponding measure. That means that the stability of the prediction decreased. This was contrary to our preliminary expectation that the reduction stabilizes the predication. In tendency this phenomenon was larger for the unbalanced data settings. One possible interpretation is that there were constellations in the data that could not be modeled sufficiently by the sharply reduced feature group selections.

The suggested procedure for the determination of the cut-off for the reduction tended to reduce the feature group selection significantly. This reduction was too drastic in some of the tests. Further approaches towards the reduction of the feature groups selections based on the relevance values for the single test runs have to be analyzed in their corresponding behavior. This is outside the scope of this thesis. A larger data set could help analyzing the phenomenon with respect to the influence of the data set size.

It is probable that the strong reduction tendency of the relevance ranking score approach prevented the reduction of the whole feature group set from achieving better generalization results. This mechanism was compensated by the introduction of pathological knowledge at the right time. For the pathological selection the incorporation of pathological knowledge was probably too early and dropped relevant features. For the computational selection the introduction of pathological knowledge by skipping semantically equivalent features came at about the right time.

Analyzing the non-Euclidean coefficient (NEC) for the best learning results in each test series did not show a clear tendency. The NEC values are given in the detailed result tables D.3 to D.6 in the appendix. The NEC did neither decrease generally during reduction nor did it generally increase. There was also no clear dependency between the generalization ability in terms of the test recognition rate and the NEC. With few exceptions there was a slight tendency that the higher the test recognition rate, the higher was also the NEC. This indicates that the modeled discrimination task is non-Euclidean in its nature.

A relevant result was that the two best generalization results incorporated the grading as a feature group. These feature group selections yielded test recognition rates that even considering their variation achieved better values than random

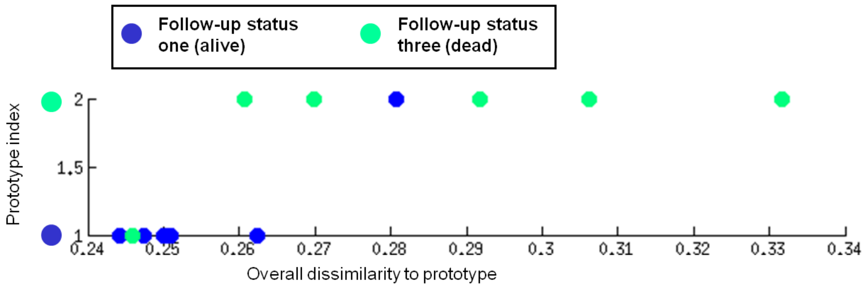


Figure 7.7: *Labeled receptive field density diagram for the test data points in the trial with the best test recognition rate for the computational selection –* ***C2*** *balanced.*

classification or classification according to the prior distribution of the classes. These feature selections were: the first reduction of the computational selection in an unbalanced setting – **C2** – and the last reduction of the whole feature group set starting with a balanced setting – **N8**. A possible strategy for the proceeding in the reduction that considers these best results would be: Start the reduction using a balanced as well as an unbalanced setting. For all subsequent reduction steps use the balanced setting.

The overall best evaluation values were achieved for the first reduction of the computational selection – **C2**. With a test recognition rate of 66.7% and its standard deviation of 7.6% as well as a clinical recognition rate of 65.8% and its standard deviation of 7.3% it yielded better results than the current clinical prediction given by the grading in section 6.3. This tendency was confirmed in the recall and precision values for the classes one and three. A further reduction of this selection decreased the generalization ability. This is an evidence that the model lost necessary complexity or feature group information in this reduction. The ERRatio may be of pathological relevance in this context. This tendency would have to be confirmed in further pathological studies involving prospective samples.

For this test runs we also evaluated the best trial using the visual evaluation methods introduced in section 4.6. This test run yielded a test recognition rate of 83.3% and a non-Euclidean coefficient of 0.500. Figure 7.7 shows the labeled receptive field density diagram for the best run over the test data points.

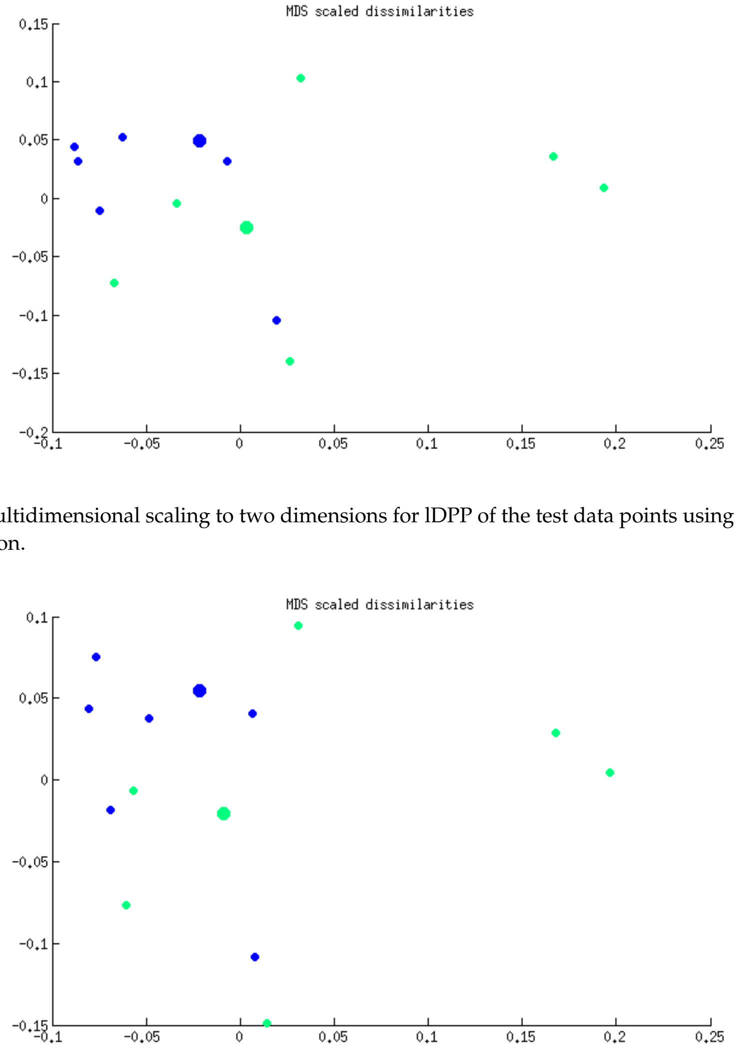
For the data point projections we conducted three MDS trials using the sammon function, the stress function and the metric stress function using the *"mdscale"* function in MATLAB from the statistics toolbox (Mathworks n.d.c). We mapped the training data points as well as the test data points. For the training data points the MDS with the sammon function did not converge within the default maximum iteration number MATLAB considers for it. We neglect this mapping. In the figures D.1 and D.2 in the appendix we show the MDS results for the training data points for stress and metric stress optimization as well as the corresponding Shepard plot. The Shepard plot shows that the distances for the stress function in tendency are overestimating whereas the distances for the metric stress are underestimating.

Figure 7.8 shows the labeled data point projection of the test data points that were generated using the stress function and the metric stress function in MDS. The abscissa and ordinate show the visualization coordinates of the mapped data points and prototypes. The prototypes are identified by larger dots. Class one is mapped to the blue color whereas class three is mapped to the green color. The multi dimensional scaling result in the upper diagram (figure D.1(a)) was achieved by optimizing the stress function. For the lower MDS diagram (figure D.1(b)) the metric stress function was applied. In figure 7.9 the Shepard plot for these two MDS optimization strategy results is shown. The distances that were found in the optimization process of MDS are plotted along the abscissa against the actual dissimilarities from the vb-KLVQ learning along the ordinate. Both optimization criteria – stress and metric stress – show satisfying consistency between the dissimilarities and the distances.

A three-dimensional example of a MDS plot as a labeled data point projection is given in figure 7.10. Here the mapped visualization coordinates were calculated to stretch over three dimensions by the *"cmdscale"* function in MATLAB's statistics toolbox (Mathworks n.d.c). Data points labeled as class one are given by blue dots and data points of class three by green dots. The larger dots are the two prototypes of the corresponding classes. In figure D.3 in the appendix we show the corresponding three-dimensional MDS result for the training data points.

7.4 Comparing results of pure Euclidean LVQ with integrative LVQ for mixed data

We conducted the following test series on the reduced feature group selections that resulted from the unbalanced first test series – **P2**, **C2**, **R2** – except for the whole feature group set that was used in its unreduced form – **N1**. This way we had a variety of suitably and unsuitably integrated feature groups to check whether this influences the results. The feature group sets were analyzed using GLVQ with a squared Euclidean distance as introduced in section 4.1.2. The settings were the same as for the combined feature group sets in the subsequent reduction tests. We expected the Euclidean GLVQ to suffer more from the unbalanced data set than the vb-KLVQ. For fair conditions we conducted the tests only for the balanced setting. Table D.7 in the appendix summarizes the results. In table D.8 we summarize the evaluation measures for direct comparison between the GLVQ and the vb-KLVQ.



(a) Multidimensional scaling to two dimensions for lDPP of the test data points using stress function.

(b) Multidimensional scaling to two dimensions for lDPP of the test data points using metric stress function.

Figure 7.8: *Labeled DPPs for the best result in integrative relevance analysis over the test data samples.*

From this comparison no clear-cut picture of the advantages or disadvantages could be derived.

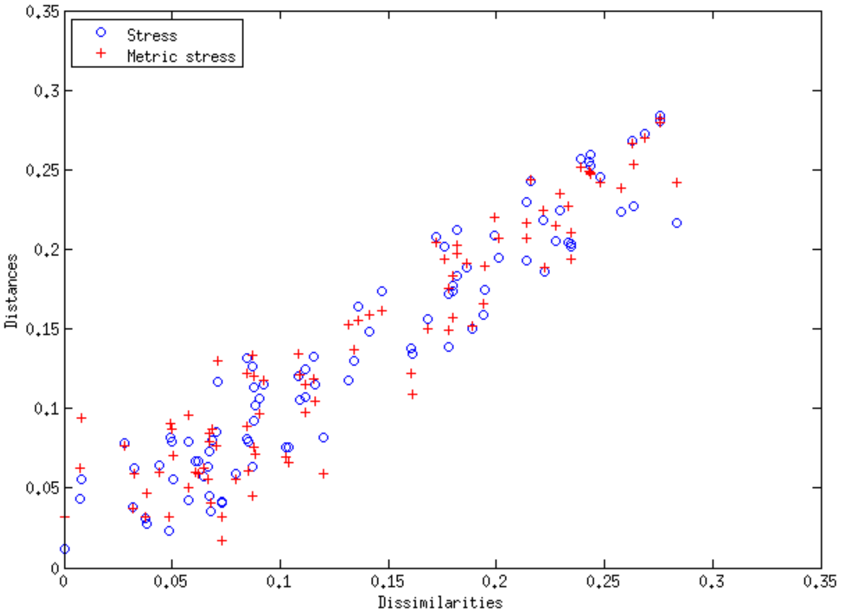


Figure 7.9: *Shepard plot for the evaluation of the corresponding mappings in figure 7.8.*

For the computational selection **C2** the training recognition rate was higher than in the vb-KLVQ with a small standard deviation. All other evaluation measures were significantly lower in the analysis based on the pure squared Euclidean distance and showed high variances between the single runs. This indicates that the vb-KLVQ in this case had a higher generalization ability over the particular feature group set than the GLVQ with the squared Euclidean distance.

In the unselected case the pure squared Euclidean analysis yielded better results except in the test recognition rate. For the pathological selection the clinical recognition rate was slightly better in the vb-KLVQ whereas all other evaluation measures were worse. In the random selection the results of the vb-KLVQ and the GLVQ were comparable to each other in all evaluation measures.

7.5 Outlier detection – Leave-one-out evaluation

Using the best feature group selection **C2** in the vb-KLVQ we conducted a leave-one-out validation as introduced in section 4.5.2 on page 77. We ran the tests twenty times

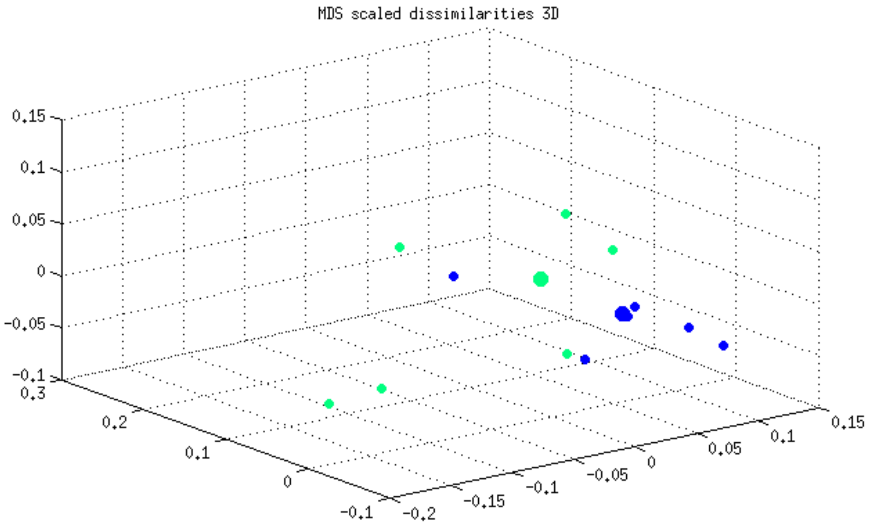


Figure 7.10: *Multidimensional scaling to three dimensions for a lDPP of the test data point using classical multidimensional scaling.*

for every data sample in the Exprimage data set except for the samples of follow-up-status two. The settings were the same as for the integrated tests on balanced data sets except for the prototype initialization. In the LOOV the randomness of the training and test set splitting is removed. With a random choice of data points as initial prototype positions we could evaluate the influence of this source of random behavior.

The results were collected in terms of the recognition ratio for every single data sample. This gave an approximate measure of how well a data sample is represented by a model learned from the feature group selection and the remaining data samples. We associated this measure with a display of the basic data for the patient samples. Figure 7.11 shows an example of the LOOV result displayed in the InfoZoom application (humanIT Software GmbH n.d.) that in the Exprimage project was used for the described tasks. In this overview possible relations to other data, e.g. the registration quality, could be identified. We did not identify a simple correlation with the registration quality or clinical data.

Figure 7.12 illustrates the detail view of the LOOV result in InfoZoom that allowed the correlation and inspection of the corresponding basic images from which the tumor representation was calculated. It shows the data section with patient samples for which the follow-up status was not suitably predicted. We highlighted two special

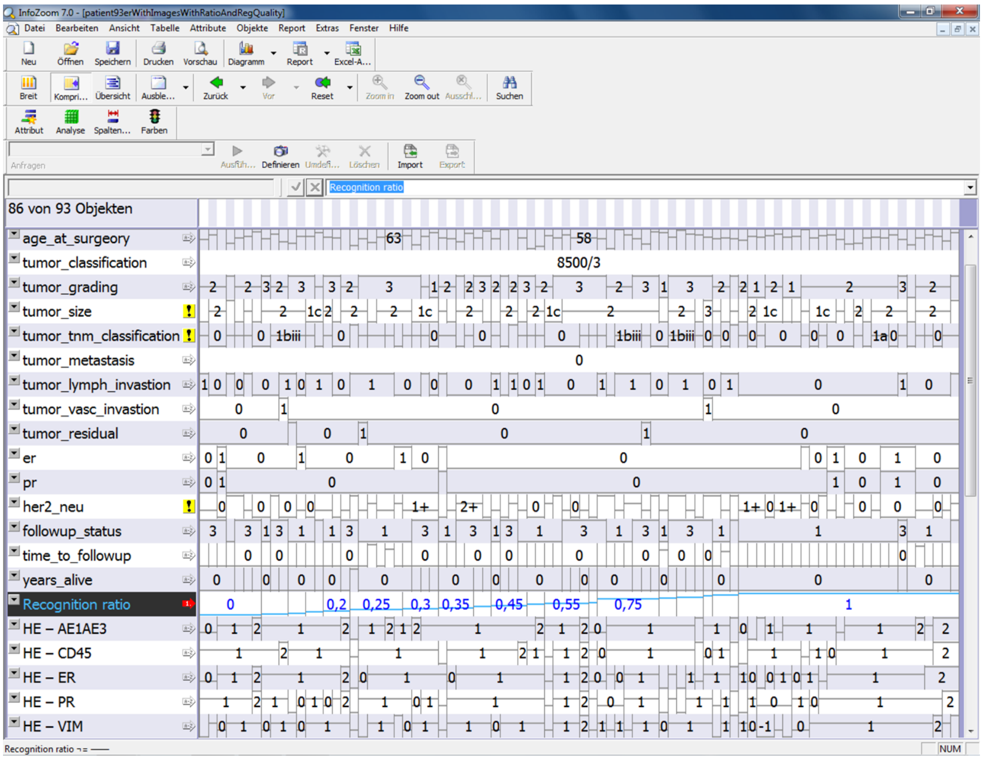
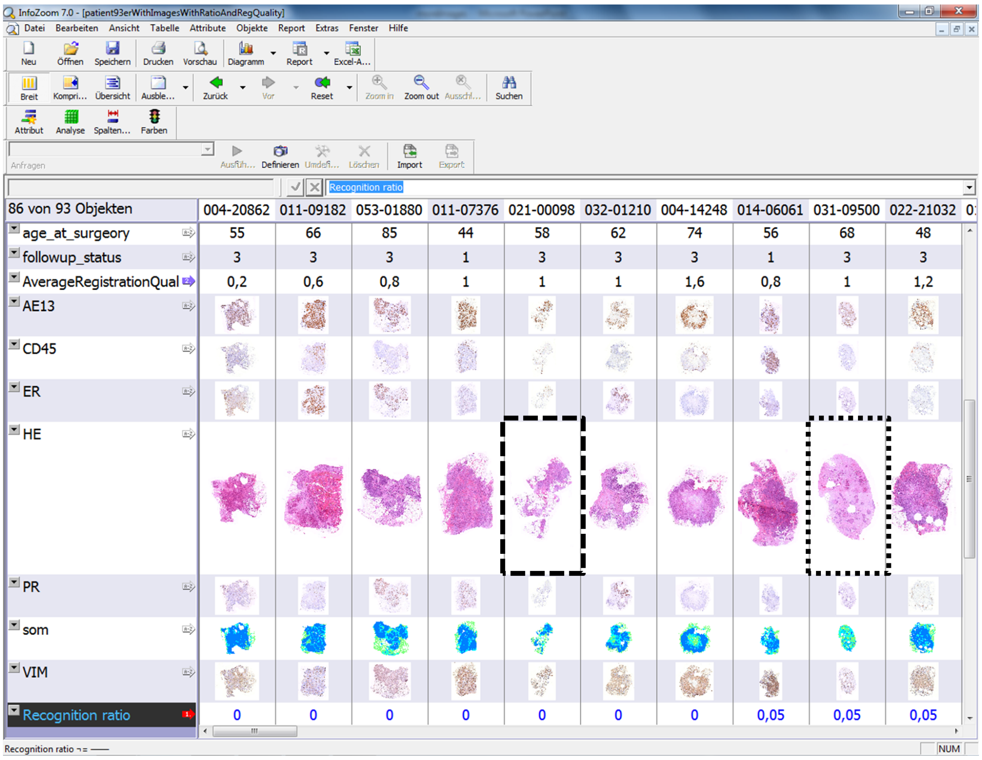

InfoZoom 7.0 - [patient93erWithImagesWithRatioAndRegQuality]
Datei Bearbeiten Ansicht Tabelle Attribute Objekte Report Extras Fenster Hilfe
Neu Öffnen Speichern Drucken Vorschau Diagramm Report Excel-A...
Breit Kompri... Übersicht Ausble... Zurück Vor Reset Zoom in Zoom out Ausschl... Suchen
Attribut Analyse Spalten... Farben
Anfragen Ausfüh... Definieren Umdefi... Löschen Import Export
Recognition ratio
86 von 93 Objekten
age_at_surgeory
tumor_classification
tumor_grading
tumor_size
tumor_tnm_classification
tumor_metastasis
tumor_lymph_invastion
tumor_vasc_invastion
tumor_residual
er
pr
her2_neu
followup_status
time_to_followup
years_alive
Recognition ratio
HE – AE1AE3
HE – CD45
HE – ER
HE – PR
HE – VIM
63
58
8500/3
0,2
0,25
0,3
0,35
0,45
0,55
0,75
1
Recognition ratio ¬= ——
NUM

Figure 7.11: *Result presentation of the leave-one-out validation for the computational feature group selection* **C2** *in the compressed view of the InfoZoom application together with clinical data.*

outlier patient samples. The dashed rectangle in figure 7.12 marks an example of bad slide preparation in histopathology. Figure 7.13(a) shows the corresponding HE image in more detail. The dotted rectangle in figure 7.12 marks a special tumor type with a high proportion of DCIS that non-invasively spread inside the milk ducts. Its detailed HE image is shown in figure 7.13(b).

7.6 Analysis of patient samples with grading value two

According to the pathological experts the grading value two gives no reliable prognosis. We conducted the reduction process with the strategy defined in section 7.3.3 starting with the whole feature group set using only samples with the grading value two. The aim was to identify feature groups that are relevant for the prognosis of


InfoZoom 7.0 - [patient93erWithImagesWithRatioAndRegQuality]
Datei Bearbeiten Ansicht Tabelle Attribute Objekte Report Extras Fenster Hilfe
Neu Öffnen Speichern Drucken Vorschau Diagramm Report Excel-A...
Breit Kompri... Übersicht Ausble... Zurück Vor Reset Zoom in Zoom out Ausschl... Suchen
Attribut Analyse Spalten... Farben
Anfragen Ausfüh... Definieren Umdefi... Löschen Import Export
Recognition ratio
Recognition ratio ¬= ——
NUM

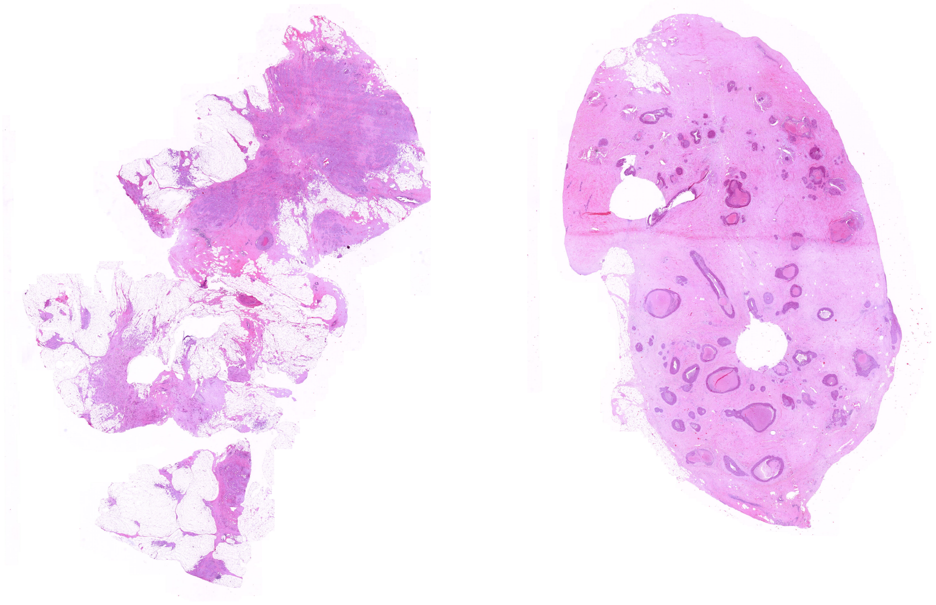
86 von 93 Objekten	004-20862	011-09182	053-01880	011-07376	021-00098	032-01210	004-14248	014-06061	031-09500	022-21032
age_at_surgeory	55	66	85	44	58	62	74	56	68	48
followup_status	3	3	3	1	3	3	3	1	3	3
AverageRegistrationQual	0,2	0,6	0,8	1	1	1	1,6	0,8	1	1,2
AE13										
CD45										
ER										
HE										
PR										
som										
VIM										
Recognition ratio	0	0	0	0	0	0	0	0,05	0,05	0,05

Figure 7.12: *Result presentation of the leave-one-out validation for the computational feature group selection* ***C2*** *in the detailed view of the InfoZoom application together with the basic image data.*

these patients. Figure 7.14 shows this reduction process and table D.9 in the appendix gives the detailed learning results.

An important finding was that the standard deviation of the test recognition rate as well as of the clinical recognition rate was significantly higher than in all other tests before, at least 20%. It is probable that there are different subtypes of surviving and deceasing patients. Further tests with different numbers of prototypes are needed to analyze this hypothesis. These tests are outside the scope of this thesis but will be considered for future work.

The pathological experts emphasized the tendency in the reduction process to prefer feature groups related to the functional markers. In the best result there were two functional marker describing feature groups, one from clinical data and one describing the heterogeneity in the geometrical distribution of the progesterone receptor in tumor regions. The pathological experts considered the identified best



(a) HE image of a patient sample that due to the bad histopathological preparation of the slide could not be suitably predicted in its outcome from models learned using all other patient samples.

(b) HE image of a patient sample that due to its special tumor type – high proportion of DCIS which non-invasively spreads inside milk ducts – could not be suitably predicted in its outcome from models learned using all other patient samples.

Figure 7.13: *Digitized color images of the HE stains for two outlier patient samples detected by the leave-one-out validation.*

feature group set, comprising the DistributionPR and the Hormonereceptors R2 feature groups, as potentially relevant starting point for further pathological research for a better prognosis in the unclear patients with grading two.

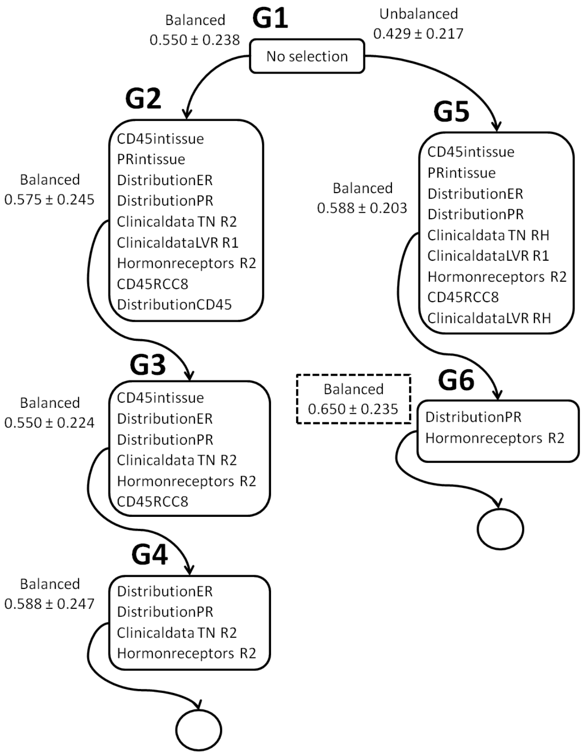


Figure 7.14: *Reduction process for the whole feature group set using only samples with a grading value of two.*

Chapter 8

Discussion and conclusion

In the following sections we will discuss the results of the work presented in this thesis. After concluding the scientific contribution of the thesis we provide ideas for future work building on the thesis' achievements.

8.1 Discussion of test series with respect to introduced hypotheses

In this section we discuss the test series with respect to the hypotheses formulated in the introduction of chapter 7. We cannot give a definite assessment of the validity of the established hypotheses. More example data and a closer long-term interaction with the pathological experts would have been necessary for a detailed analysis. The actual pathological scientific question was the prediction of the therapy success for different patients. To answer this question it is particularly important to know whether the patient was treated with chemotherapy. A combined label from follow-up status and therapy has to be analyzed for a pertinent therapy success prediction. We used the follow-up status as single label as the therapy information was not available in our cohort. The classification based on the follow-up status is no ecologically valid implementation of the actual question. In this section we provide and discuss tentative evidence of the hypothesis validity for the computational aspects of the follow-up status classification.

8.1.1 Hypothesis 1 – improvement in reduction

The first hypothesis stated that it is possible to reduce candidate feature group sets according to the identified relevance values such that the generalization of the learned classification becomes better and more stable. The generalization ability was measured in terms of the recognition rates and Cohen's kappa for classification of the test set. A detailed introduction of the test results was given in the sections 7.3.2 and 7.3.3.

We conducted reduction series according to a relevance ranking analysis over the dissimilarity parameters for four different settings:

Whole feature group ensemble – N1 For this choice we did not apply any previous knowledge from computational or pathological analysis.

Pathological selection – P1 According to current pathological research interests feature groups for every tumor typical activity field were chosen according to their computational isolated relevance analysis. Conceptually the choice was driven by pathological knowledge and corrected by computational insight. The resulting number of nine feature groups was taken as a bench mark for the other selections.

Computational selection – C1 We chose nine feature groups that scored highest in the isolated relevance analysis according to the stable generalization ability. The choice was influenced by the pathological corrective to avoid semantic redundancy in the feature groups. For this selection the computational knowledge was driving the choice of feature groups whereas the pathological knowledge corrected it.

Random selection – R1 For this selection we randomly drew nine feature groups out of the whole feature group ensemble without the influence of computational or pathological criteria.

For the whole feature group ensemble the reduction yielded generalization results that were significantly different for the first reduction step based on the balanced or the unbalanced setting. For the subsequent reduction of the reduced set from the unbalanced setting no improvement of the recognition rate was achieved. For the balanced first reduction in the subsequent tests the results became stably better with every reduction process for the test recognition rate. For the clinical recognition rate – evaluation of the recognition rate over the whole data set but just samples for grading one or three – the mean value was increasing during the reduction process but also the variation of this recognition rate. The best result according to the test recognition rate was achieved for the maximum reduction. It resulted in the selection of the feature groups: Grading R1 and Grading R2. That was a small set that also comprised a semantically narrow field of tumor description. Even taking the variation into account its recognition rate was better than for random classification.

For the pathological selection there was only a slight improvement in one reduction during the reduction process. Possible reasons for the reduction to fail are:

- The available training data is not representative enough to conclude dissimilarity parameters and consequently relevance values from it.

- The dissimilarity values in the feature groups are not comparable and consequently the dissimilarity parameters can not be interpreted as relevances.
- The relevance ranking score is not suited for the integration of the single relevance values in the trials. That means it cannot model adequately the overall relevance of a feature group.
- The selection of the cut-off in the ranking score is not adequate.
- Relevant information was dropped before.

The most probable reason is the last one. Evidence for that reason can also be found in the evaluation of the reduction steps for the other feature groups. None of the feature groups identified as most relevant in the other reduction processes were present in the basic pathological selection. This provides evidence for the pathological knowledge to be incorporated too early into the selection process. Relevant feature groups could have been dropped.

The best generalization results in the whole test series were generated by the reduction in the balanced analysis of the reduced computational selection **C2**. The same holds for the Cohen′s kappa for the classification of the test set. The clinical recognition rate was also best in the first reduction step but slightly lower than in the last reduction of the whole feature group ensemble. The computational selection in this stage comprised four clinical feature groups as well as the coarse morphometric tumor representation using three clusters and the computationally determined ratio of ER expression. The further reduction of the computational selection in the second step that removed the ERRatio feature group decreased the generalization ability in all pertinent measures. This indicates that information relevant for the classification was dropped. This phenomenon can be defined as an overselection of feature groups where no further suitable reduction of the model complexity is possible without a loss of predictive power.

In the random selection of feature groups the reduction process showed a tendency to improve the generalization ability. As the random selection did not comprise relevant features the recognition rates were not better than random classification.

Summarizing the discussion of the single reduction processes there is evidence that hypothesis 1 is valid. As another result of these tests we found that the system tends to reduce the number of feature groups too much. In order to avoid this effect it proved to be advantageous to incorporate the pathological knowledge as a corrective at the right time.

8.1.2 Hypothesis 2 – stability in combinations

In hypothesis 2 we assumed that a suitable combination of feature groups yields better and more stable results than the single feature groups on their own. We introduced the test results for the single feature groups in section 7.2.1. In section 7.3.3 we discussed the results for the feature group combinations. Using the combined feature group selections all runs converged within a smaller number of epochs than the runs for the single feature groups. Convergence was also reliable in combinations with single feature groups that did not show proper convergence in the isolated analysis.

For the best reduction result in the computational selection we achieved test recognition rates that in average were the same as for the best single feature group (Grading RH). According to the smaller standard deviation (7.6% instead of 14%) the results for the feature group combination was stably better. Thus the suitable combination yielded more stable results.

There is evidence that hypothesis 2 is valid.

8.1.3 Hypothesis 3 – improvement of grading

We formulated the hypothesis that the suitably integrated combination of feature groups yields better results than the pathological classification by the grading. We discussed the question of comparability of the results in section 7.3.2 and applied the corresponding measures in the different tests for the combined feature groups.

For the reduced computational feature group selection as well as for the reduction of the whole feature group ensemble we achieved a classification that is slightly better than the classification given by the grading in section 6.3 according to the clinical recognition rate (65.8% compared to 61.4%). The values for the recall and precision of the classes are comparable with the difference that the grading gives higher preference to class three than our classification.

This test series provides evidence that hypothesis 3 is valid. The stability of the classification cannot be judged as we did not have comparable data, i.e. reliability tests, for the grading in our data set.

8.1.4 Hypothesis 4 – improvement by structure

Hypothesis 4 expressed the expectation that a suitably integrated combination of feature groups with their conceptually adequate dissimilarities yields more stable and better results than the use of a comparable classification with an overall Euclidean distance. We compared the results of a GLVQ using the squared Euclidean distance with the results of the vb-KLVQ on the feature group selections resulting from the first unbalanced reduction step except for the whole feature group set that was used in its unreduced form. In section 7.4 we show the results of these tests.

For the feature group selection that showed a significant increase of generalization ability in the previous reduction step – the reduced computational selection **C2** – the vb-KLVQ in tendency achieved better recognition rates and Cohen's kappa values than the GLVQ. For the unreduced whole feature group selection **N1** an improvement in the test recognition rate was achieved but not in Cohen's kappa. For the other feature group selections the generalization ability was better in the squared Euclidean GLVQ.

In the overall analysis of the results the GLVQ shows higher training recognition rates whereas the generalization ability in tendency is higher for the vb-KLVQ. The hypothesis 4 is tentatively warranted. A series of tests with more complete data has to be conducted in future work to clarify the validity of the hypothesis.

8.1.5 Hypothesis 5 – identification of saliency

We hypothesized that the leave-one-out validation conducted with focus on the recognition rate for every single patient sample is capable of identifying pathologically salient patient samples. In section 7.5 the results of the leave-one-out validation are shown in exemplary form. There were three kinds of patient samples for which no reliable prediction of the follow-up status was achievable by the learned models:

- samples with preparation problems in histopathology,
- samples with known special tumor types and
- samples with no directly comprehensible modeling problem.

The samples of the last type can provide evidence either for necessary model adaptations or necessary pathological research. In the discussion with the pathological experts it turned out that in some cases there is need for a known medical clue to be represented additionally. Further relevance analyses incorporating feature groups representing this medical clue can probably improve the predictability of the follow-up status for the corresponding patient samples.

When the pathological experts cannot associate known complementing pathological factors for improving the reliable prediction of the corresponding patient sample, pathological research based on the relevances and combination of feature groups used in the leave-one-out validation might be necessary. This effort is justifiable if there are identified salient and potentially relevant patient samples. This depends from large, representative data sets.

From the first two kinds of samples that we identified in the conducted leave-one-out validation we gained tentative evidence that hypothesis 5 is valid.

8.2 Conclusion

We developed the methods in this thesis as a framework to approximate relevant contextual feature group combinations for the identification of disease subtypes in pathological research. These feature group combinations are neither known nor could they be understood immediately in detail if a computer system marks them as significant.

In our application the situation is even worse as neither a necessary amount of data is available nor does the data provide a sound statistical basis for the exact definition of a testable research aim. Even the labels of the patient data samples provide no conceptually adequate assessment base. This would be the case if there was an information about the applied adjuvant therapies. Then the various patient data could be matched against the adequacy of the treatment scheme. That does not change the fact that the introduced reduction process provides a suitable methodological extension to the computer supported learning and identification processes. It shows where to carefully check the behavior of the system in detail.

In the prediction of breast cancer follow-up we could show that using the developed learning and evaluation approaches it is possible to identify medically relevant feature group combinations. The hypotheses formulated in the introduction of chapter 7 were tentatively warranted.

8.3 Future prospects

For future research concerning the system we see the following subjects of study:

1. We want to analyze the creation of feature group selections from the isolated relevance analysis by an approach incorporating the evaluation of the *Cohen's or Fleiss' kappa* between the classifications of the single feature groups. Different feature groups that yield good recognition rates with different classifications are assumed to provide a more comprehensive model of the tumor situation in a patient.

2. In the application of the integrative relevance analysis the *normalization of the dissimilarities* is an important issue that influences the interpretability of the dissimilarity parameters as relevance values. A study analyzing the suitability and the influence of different normalizations on the reduction process and the corresponding generalization ability of the system as well as the pathological relevance of the findings has to be conducted to further improve the applicability of the framework as cognitive support system.

3. Another study subject relates to the appropriate determination of the *cut-off for the reduction* of the feature group selections. Different methods have to be analyzed in their influence on the reduction process with respect to the computational and pathological performance.

4. For the clinical data we mentioned in section 7.1.4, that we assume there are more *suitable dissimilarities for the clinical feature groups* that result from a pertinent combination of statistical and human judgment on the dissimilarities. A close long-term interaction with the pathological experts is needed to achieve a common understanding of the domain specific underlying concepts of dissimilarity. Based on this understanding a suitable dissimilarity definition for the clinical data can be found.

5. A *life-long learning ability* that allows the system to learn during pathological routine will enhance the possibilities of a long term interaction with the pathological experts.

6. A valuable enhancement for the real-world application of the system is the ability to cope with *missing features* as the quality of medical documentation of the patient cases is often insufficient.

Summarizing, our central aim for further research is to apply the system in a study with a larger number of patient cases and a more continuous interaction with the experts of the application domain. The underlying concepts of the framework are generally used to support cognitively difficult insight processes. The cognitive support system can be adapted to the specifics of other application domains by introducing other feature groups from pertinent image analysis or biomedical measurements.

Appendix A

Distance measures in Structured Batch Neural Gas

Basis for all following calculations is the equation:

$$\frac{\partial L\left(w,\alpha\right)}{\partial\left[w_n\right]_{[j]}}=\frac{1}{2}\sum_{k=1}^{K}h_\sigma\left(r_{k,n}\right)\cdot\left(\alpha_j^n\right)^2\cdot\frac{\partial d_j\left(\left[v_k\right]_{[j]},\left[w_n\right]_{[j]}\right)}{\partial\left[w_n\right]_{[j]}}\xi$$

This must be zero for every possible ξ. For solving this equation and determining the new $\left[w_n\right]_{[j]}$ it is necessary to know the structure of the $d_{j\star}$ under consideration.

A.1 Squared Euclidean distance

$$d_j\left([v_k]_{[j]},[w_n]_{[j]}\right)=\left([v_k]_{[j]}-[w_n]_{[j]}\right)^\top\left([v_k]_{[j]}-[w_n]_{[j]}\right)$$

$$\frac{\partial d_j^n}{\partial[w_n]_{[j]}}=-2\cdot\left([v_k]_{[j]}-[w_n]_{[j]}\right)\tag{A.1.1}$$

$$0=\sum_{k=1}^{K}h_\sigma\left(r_{k,n}\right)\left(\alpha_j^n\right)^2\left(\left[w_n\right]_{[j]}-\left[v_k\right]_{[j]}\right)$$

and

$$\left[w_n\right]_{[j]}=\frac{\sum_{k=1}^{K}h_\sigma\left(r_{k,n}\right)\left[v_k\right]_{[j]}}{\sum_{k=1}^{K}h_\sigma\quad\left(r_{k,n}\right)}$$

A.2 γ-Divergences

Collection of parameterized divergences (e.g. Cauchy-Schwarz-Divergence for $\gamma=1$, see (Villmann and Haase 2010) for more detail)

$$\begin{aligned} d_{j,\gamma}\left(v_k, w_n\right) = \frac{1}{\gamma+1} \cdot \log &\left[\left(\sum_{m\in[j]} \left([v_k]_m\right)^{\gamma+1} \right)^{\frac{1}{\gamma}} \cdot \left(\sum_{m\in[j]} \left([w_n]_m\right)^{\gamma+1} \right) \right] \\ - \log &\left[\left(\sum_{m\in[j]} [v_k]_m \cdot \left([w_n]_m\right)^{\gamma} \right)^{\frac{1}{\gamma}} \right] \end{aligned}$$

and

$$\frac{\partial d_\gamma\left([v_k]_{[j]}, [w_n]_{[j]}\right)}{\partial [w_n]_{[j]}} = \frac{[w_n]_{[j]}^{\gamma}}{\sum_{m\in[j]}\left([w_n]_m\right)^{\gamma+1}} - \frac{[v_k]_{[j]} \cdot \left([w_n]_{[j]}\right)^{\gamma-1}}{\sum_{m\in[j]}[v_k]_m \cdot [w_n]_m^{\gamma}} \tag{A.2.1}$$

$$0 = \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)\left(\alpha_j^n\right)^2 \cdot \left(\frac{\left([w_n]_{[j]}\right)^{\gamma}}{\sum_{m\in[j]}[w_n]_m^{\gamma+1}} - \frac{[v_k]_{[j]} \cdot \left([w_n]_{[j]}\right)^{\gamma-1}}{\sum_{m\in[j]}[v_k]_m \cdot \left([w_n]_m\right)^{\gamma}} \right)$$

$$\begin{aligned} \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{\left([w_n]_{[j]}\right)^{\gamma}}{\sum\limits_{m\in[j]}\left([w_n]_m\right)^{\gamma+1}} &= \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{[v_k]_{[j]} \cdot \left([w_n]_{[j]}\right)^{\gamma-1}}{\sum\limits_{m\in[j]}[v_k]_m \cdot \left([w_n]_m\right)^{\gamma}} \\ [w_n]_{[j]}^{\gamma} \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{1}{\sum\limits_{m\in[j]}\left([w_n]_m\right)^{\gamma+1}} &= [w_n]_{[j]}^{\gamma-1} \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{[v_k]_{[j]}}{\sum\limits_{m\in[j]}[v_k]_m \cdot \left([w_n]_m\right)^{\gamma}} \\ [w_n]_{[j]} \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{1}{\sum\limits_{m\in[j]}\left([w_n]_m\right)^{\gamma+1}} &= \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{[v_k]_{[j]}}{\sum\limits_{m\in[j]}[v_k]_m \cdot \left([w_n]_m\right)^{\gamma}} \end{aligned}$$

$$[w_n]_{[j]} = \frac{\sum\limits_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{1}{\sum\limits_{m\in[j]}\left([w_n]_m\right)^{\gamma+1}}}{\sum\limits_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{[v_k]_{[j]}}{\sum\limits_{m\in[j]}[v_k]_m \cdot \left([w_n]_m\right)^{\gamma}}} \tag{A.2.2}$$

A.3 Kullback-Leibler-Divergence for Gaussians

$$d_j\left(v_k, w_n\right) = \frac{1}{2}\left[\frac{\left(\mu_{v_k}^j - \mu_{w_n}^j\right)^2}{\left(\sigma_{w_n}^j\right)^2} + \frac{\left(\sigma_{v_k}^j\right)^2}{\left(\sigma_{w_n}^j\right)^2} - \log\frac{\left(\sigma_{v_k}^j\right)^2}{\left(\sigma_{w_n}^j\right)^2} - 1 \right]$$

$$\frac{\partial d_j}{\partial \mu^j_{w_n}} = -\frac{\left(\mu^j_{v_k} - \mu^j_{w_n}\right)}{\left(\sigma^j_{w_n}\right)^2} \tag{A.3.1}$$

$$\frac{\partial d_j}{\partial \sigma^j_{w_n}} = \frac{1}{\left(\sigma^j_{w_n}\right)^3} \left(-\left(\mu^j_{v_k} - \mu^j_{w_n}\right)^2 - \left(\sigma^j_{v_k}\right)^2 + \left(\sigma^j_{w_n}\right)^2\right) \tag{A.3.2}$$

Considering $\mu^j_{w_n}$:

$$\begin{aligned}
0 &= \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)\left(\alpha^n_j\right)^2 \frac{1}{\left(\sigma^j_{w_n}\right)^2} \cdot \left(\mu^j_{w_n} - \mu^j_{v_k}\right) \\
\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{1}{\left(\sigma^j_{w_n}\right)^2} \mu^j_{w_n} &= \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{1}{\left(\sigma^j_{w_n}\right)^2} \mu^j_{v_k} \\
\mu^j_{w_n} &= \frac{\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)\mu^j_{v_k}}{\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)} \qquad \text{(A.3.3)}
\end{aligned}$$

Considering $\sigma^j_{w_n}$:

$$\begin{aligned}
0 &= \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) \frac{1}{\left(\sigma^j_{w_n}\right)^3}\left[-\left(\mu^j_{v_k} - \mu^j_{w_n}\right)^2 - \left(\sigma^j_{v_k}\right)^2 + \left(\sigma^j_{w_n}\right)^2\right] \\
\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)\left(\sigma^j_{w_n}\right)^2 &= \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)\left(\left(\mu^j_{v_k} - \mu^j_{w_n}\right)^2 + \left(\sigma^j_{v_k}\right)^2\right) \\
\left(\sigma^j_{w_n}\right)^2 \cdot \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right) &= \sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)\left(\left(\mu^j_{v_k} - \mu^j_{w_n}\right)^2 + \left(\sigma^j_{v_k}\right)^2\right) \\
\left(\sigma^j_{w_n}\right)^2 &= \frac{\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)\left(\left(\mu^j_{v_k} - \mu^j_{w_n}\right)^2 + \left(\sigma^j_{v_k}\right)^2\right)}{\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)} \\
\sigma^j_{w_n} &= \sqrt{\frac{\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)\left(\left(\mu^j_{v_k} - \mu^j_{w_n}\right)^2 + \left(\sigma^j_{v_k}\right)^2\right)}{\sum_{k=1}^{K} h_\sigma\left(r_{k,n}\right)}} \qquad \text{(A.3.4)}
\end{aligned}$$

Appendix B

Distance measures and their derivatives

B.1 Squared Euclidean distance

$$d_j\left([v_k]_{[j]},[w_n]_{[j]}\right)=\left([v_k]_{[j]}-[w_n]_{[j]}\right)^{\top}\left([v_k]_{[j]}-[w_n]_{[j]}\right)$$
$$\frac{\partial d_j^-}{\partial[w_-]_{[j]}}=-2\cdot\left([v_k]_{[j]}-[w_-]_{[j]}\right)$$

B.2 γ-Divergences

Collection of parameterized divergences (e.g. Cauchy-Schwarz-Divergence for $\gamma=1$, see (Villmann and Haase 2010) for more detail)

$$\begin{aligned}d_\gamma\left(v,w\right)=\frac{1}{\gamma+1}\cdot\log&\left[\left(\int v^{\gamma+1}\left(x\right)dx\right)^{\frac{1}{\gamma}}\cdot\left(\int w^{\gamma+1}\left(x\right)dx\right)\right]\\&-\log\left[\left(\int v\left(x\right)\cdot w\left(x\right)^{\gamma}dx\right)^{\frac{1}{\gamma}}\right]\end{aligned}$$

with integrals becoming sums (because of discrete data) :

$$\begin{aligned}d_\gamma\left(v,w\right)=\frac{1}{\gamma+1}\cdot\log&\left[\left(\sum_x v_x^{\gamma+1}\right)^{\frac{1}{\gamma}}\cdot\left(\sum_x w_x^{\gamma+1}\right)\right]\\&-\log\left[\left(\sum_x v_x\cdot w_x^{\gamma}\right)^{\frac{1}{\gamma}}\right]\end{aligned}$$

and thus

$$\frac{\partial d_\gamma\left([v]_{[j]},[w_+]_{[j]}\right)}{\partial[w_+]_{[j]}}=\frac{[w_+]_{[j]}^{\gamma}}{\sum_{x\in[j]}[w_+]_x^{\gamma+1}}-\frac{[v]_{[j]}\cdot[w_+]_{[j]}^{\gamma-1}}{\sum_{x\in[j]}[v]_x\cdot[w_+]_x^{\gamma}}$$

B.3 Kullback-Leibler-Divergence for Gaussians

$$d_j = d_j\left(\left[\begin{smallmatrix}\mu_v\\ \sigma_v\end{smallmatrix}\right], \left[\begin{smallmatrix}\mu_w\\ \sigma_w\end{smallmatrix}\right]\right)$$

$$= \begin{cases} \frac{1}{2}\Big[\log\frac{|\Sigma_w|}{|\Sigma_v|} + \mathrm{Tr}\left[\Sigma_w^{-1}\Sigma_v\right] - d \\ \quad + \left(\mu_v - \mu_w\right)^T \Sigma_w \left(\mu_v - \mu_w\right)\Big] & \text{multidimensional case} \\ \\ \frac{1}{2}\left[\frac{(\mu_v-\mu_w)^2}{\sigma_w^2} + \frac{\sigma_v^2}{\sigma_w^2} - \log\frac{\sigma_v^2}{\sigma_w^2} - 1\right] & \text{onedimensional case} \end{cases}$$

$$\frac{\partial d_j}{\partial \mu_w} = \frac{\partial \frac{1}{2}\left(\frac{(\mu_v-\mu_w)^2}{\sigma_w^2} + \frac{\sigma_v^2}{\sigma_w^2} - \log\frac{\sigma_v^2}{\sigma_w^2} - 1\right)}{\partial \mu_w} = -\frac{(\mu_v - \mu_w)}{\sigma_w^2}$$

$$\begin{aligned} \frac{\partial d_j}{\partial \sigma_w} &= \frac{\partial \frac{1}{2}\left(\frac{(\mu_v-\mu_w)^2}{\sigma_w^2} + \frac{\sigma_v^2}{\sigma_w^2} - \log\frac{\sigma_v^2}{\sigma_w^2} - 1\right)}{\partial \sigma_w} \\ &= -\frac{\left(\mu_v - \mu_w\right)^2}{\sigma_w^3} - \frac{\sigma_v^2}{\sigma_w^3} + \frac{1}{\sigma_w} \\ &= \frac{1}{\sigma_w^3}\left(-\left(\mu_v - \mu_w\right)^2 - \sigma_v^2 + \sigma_w^2\right) \end{aligned}$$

Appendix C

Test result tables for isolated relevance evaluation

Feature group selection	Mean test recognition rate	Standard deviation
Grading RH	66.7%	14.0%
SOMnCl3	65.0%	13.1%
Grading R1	66.7%	16.2%
SOMnCl4	60.0%	10.7%
Clinical TN RH	55.8%	6.7%
Clinical LVR RH	57.9%	9.9%
ERRatio	50.8%	3.7%
Grading R2	60.0%	13.1%
CD45co-occurrence	52.1%	5.3%
Hormonereceptors R1	50.8%	4.6%
SOMnCl2	60.0%	13.9%
Hormonereceptors RH	55.0%	9.1%
AreaRegionCluster5	52.9%	7.3%
Clinical LVR R1	57.5%	12.1%
RegionPerimeter	52.9%	7.8%
PRLDQ	55.0%	10.3%
DGDeg2	53.3%	8.7%
DGDeg1	50.0%	5.4%
DistributionPR	58.7%	14.4%
NumberRegions	55.0%	10.9%
AreaRegionsER	51.3%	7.3%
RegionSize	50.4%	7.4%
PRintissue	50.4%	7.4%
ERRCC8	55.0%	12.2%
DGDist1	47.9%	5.3%
RelativeAreaToTumor	50.0%	7.6%

MoAE1AE3nCl4	54.6%	12.2%
AreaRegionCluster4	50.8%	8.5%
SOMnCl7	53.3%	11.6%
Clinical TN R1	51.7%	10.0%
CD45Ratio	52.9%	11.2%
AreaRegionCluster1	50.0%	8.5%
Clinical TN R2	54.2%	13.1%
Clinical LVR R2	55.4%	14.4%
AreaRegionsPR	48.8%	7.8%
DGRandicIndex	50.4%	10.3%
AreaRegionCluster2	49.6%	9.5%
MeanAreaToTumor	47.5%	8.2%
AreaRegionsTum	47.5%	8.2%
MSTDist1	47.1%	8.2%
ERinTissue	51.3%	12.5%
CD45RCC8	49.6%	11.0%
InnerTumorStructure	53.3%	15.2%
CD45inTissue	48.3%	10.3%
Hormonereceptors R2	47.5%	9.8%
PRRatio	46.7%	9.1%
AreaRegionCluster3	47.5%	10.2%
DGCyclomaticNumber	47.1%	9.9%
PRRCC8	47.9%	10.8%
ERLDQ	46.7%	9.9%
DistributionER	49.6%	13.1%
DGDist2	46.7%	10.3%
ERco-occurrence	44.6%	8.7%
NumberOfRegions	47.5%	11.8%
MoAE1AE3nCl3	44.2%	8.6%
DGWeightedDeg	44.6%	9.1%
DistributionCD45	45.4%	9.9%
Age	50.0%	14.8%
DGnNodes	47.5%	13.8%
HRTumorRegions	48.3%	14.7%
AbsoluteArea	46.7%	13.1%
MSTDist2	46.3%	13.9%
MoAE1AE3nCl2	42.1%	11.0%
ClusterRegionNumber	43.8%	12.6%
RelAreaStroma	45.0%	14.4%

CD45LDQ	42.1%	12.8%
PRco-occurrence	41.3%	15.2%

Table C.1: Overview of the generalization results for all feature groups in the isolated relevance analysis

Appendix D

Test results for integral relevance analysis and related tests

D.1 Test results for first test series

The test results for the first test series are given for four different feature group selections. The evaluation measures that we concerned where: training recognition rate, test recognition rate, clinical recognition rate and the test Cohen's κ. Table D.1 shows the results for the unbalanced setting whereas table D.2 gives those of the balanced setting.

D.2 Test results for subsequent reduction of feature group sets

The following tables give the detailed test results for the reduction processes of the four different feature group selections. The training recognition rate, test recognition rate, clinical recognition rate and test Cohen's κ were used as evaluation values. Their average value as well as their standard deviation was considered. Furthermore for the test run in each test series that showed the highest test recognition rate we calculated the corresponding non-euclidean coefficient. It is given in the tables together with the corresponding test recognition rate.

The reduction process for the pathological selection is given in table D.3. For the computational selection the detailed results of the reduction are depicted in table D.4. Table D.5 shows the reduction results for the random selection. For the reduction process of the whole feature group ensemble the detailed results are listed in table D.6.

Feature group selection	Type of evaluation value	average value	standard deviation
No selection	Training RR	62.1%	2.0%
	Test RR	59.6%	4.8%
	Clinical RR	47.5%	3.6%
	Test Cohen's κ	0.405	0.062
Pathological selection	Training RR	64.2%	3.7%
	Test RR	57.9%	8.0%
	Clinical RR	51.5%	3.9%
	Test Cohen's κ	0.410	0.097
Computational selection	Training RR	64.5%	3.8%
	Test RR	56.8%	12.4%
	Clinical RR	49.7%	2.0%
	Test Cohen's κ	0.438	0.135
Random selection	Training RR	66.0%	3.5%
	Test RR	49.3%	6.5%
	Clinical RR	51.1%	4.6%
	Test Cohen's κ	0.427	0.082

Table D.1: *Results for different feature group selections in first test series with unbalanced setting. We abbreviate the recognition rate by RR.*

Feature group selection	Type of evaluation value	average value	standard deviation
No selection	Training RR	56.3%	4.5%
	Test RR	52.5%	8.6%
	Clinical RR	46.7%	9.2%
	Test Cohen's κ	0.386	0.097
Pathological selection	Training RR	62.4%	6.3%
	Test RR	56.3%	12.1%
	Clinical RR	55.8%	20.1%
	Test Cohen's κ	0.459	0.150
Computational selection	Training RR	66.7%	4.2%
	Test RR	54.6%	8.3%
	Clinical RR	49.5%	14.3%
	Test Cohen's κ	0.467	0.098
Random selection	Training RR	68.1%	4.1%
	Test RR	52.5%	12.1%
	Clinical RR	51.9%	17.6%
	Test Cohen's κ	0.477	0.130

Table D.2: *Results for different feature group selections in first test series with balanced test and training data sets. We abbreviate the recognition rate by RR.*

Feature group selection	Setting	Type of evaluation value	average value	standard deviation of value
P1	balanced	Training RR	62.4%	6.3%
		Test RR	56.3%	12.1%
		Clinical RR	55.8%	20.1%
		Test Cohen's κ	0.459	0.150
		Best test RR	75.0%	
		NEC for best test	0.500	
	unbalanced	Training RR	64.2%	3.7%
		Test RR	57.9%	8.0%
		Clinical RR	51.5%	3.9%
		Test Cohen's κ	0.410	0.097
		Best test RR	71.4%	
		NEC for best test	0.500	
P2	balanced	Training RR	59.4%	3.9%
		Test RR	53.8%	11.0%
		Clinical RR	53.0%	18.5%
		Test Cohen's κ	0.438	0.126
		Best test RR	75.0%	
		NEC for best test	0.416	
	unbalanced	Training RR	61.1%	1.7%
		Test RR	58.2%	8.8%
		Clinical RR	58.1%	13.2%
		Test Cohen's κ	0.431	0.116
		Best test RR	71.4%	
		NEC for best test	0.411	
P3	balanced	Training RR	60.0%	2.4%
		Test RR	52.1%	11.7%
		Clinical RR	56.8%	14.7%
		Test Cohen's κ	0.405	0.141
		Best test RR	75.0%	
		NEC for best test	0.421	
	unbalanced	Training RR	58.8%	4.2%
		Test RR	54.2%	11.3%
		Clinical RR	57.4%	16.9%

		Test Cohen's κ	0.442	0.136
		Best test RR	75.0%	
		NEC for best test	0.426	

Table D.3: Results in the reduction process for the pathological selection. The abbreviations for the feature sets correspond to the marks in figure 7.3. Furthermore we abbreviate recognition rate by RR.

Feature group selection	Setting	Type of evaluation value	average value	standard deviation of value
C1	balanced	Training RR	66.7%	4.2%
		Test RR	54.6%	8.3%
		Clinical RR	49.5%	14.3%
		Test Cohen's κ	0.467	0.098
		Best test RR	75.0%	
		NEC for best test	0.530	
	unbalanced	Training RR	64.5%	3.8%
		Test RR	56.8%	12.4%
		Clinical RR	49.7%	2.0%
		Test Cohen's κ	0.438	0.135
		Best test RR	85.7%	
		NEC for best test	0.501	
C2	balanced	Training RR	67.1%	4.9%
		Test RR	66.7%	7.6%
		Clinical RR	65.8%	7.3%
		Test Cohen's κ	0.609	0.080
		Best test RR	75.0%	
		NEC for best test	0.416	
	unbalanced	Training RR	65.1%	3.9%
		Test RR	57.5%	11.0%
		Clinical RR	56.6%	21.3%
		Test Cohen's κ	0.474	0.120
		Best test RR	78.6%	
		NEC for best test	0.418	

C3	balanced	Training RR	65.8%	4.5%
		Test RR	60.0%	15.2%
		Clinical RR	61.7%	21.2%
		Test Cohen's κ	0.529	0.178
		Best test RR	75.0%	
		NEC for best test	0.500	
	unbalanced	Training RR	64.7%	1.8%
		Test RR	60.7%	8.8%
		Clinical RR	55.0%	14.6%
		Test Cohen's κ	0.518	0.106
		Best test RR	71.4%	
		NEC for best test	0.500	
C4	balanced	Training RR	67.8%	2.9%
		Test RR	57.1%	12.8%
		Clinical RR	57.4%	16.9%
		Test Cohen's κ	0.510	0.160
		Best test RR	83.3%	
		NEC for best test	0.500	
	unbalanced	Training RR	65.8%	2.2%
		Test RR	55.7%	12.2%
		Clinical RR	56.3%	18.3%
		Test Cohen's κ	0.488	0.134
		Best test RR	78.6%	
		NEC for best test	0.386	

Table D.4: Results in the reduction process for the computational selection. The abbreviations for the feature sets correspond to the marks in figure 7.4. Furthermore we abbreviate recognition rate by RR.

Feature group selection	Setting	Type of evaluation value	average value	standard deviation of value
N1	balanced	Training RR	56.3%	4.5%
		Test RR	52.5%	8.6%
		Clinical RR	46.7%	9.2%
		Test Cohen's κ	0.386	0.097

		Best test RR	66.7%	
		NEC for best test	NaN	
	unbalanced	Training RR	62.1%	2.0%
		Test RR	59.6%	4.8%
		Clinical RR	47.5%	3.6%
		Test Cohen's κ	0.405	0.062
		Best test RR	64.3%	
		NEC for best test	NaN	
N2	balanced	Training RR	55.3%	5.1%
		Test RR	49.2%	6.0%
		Clinical RR	46.7%	15.0%
		Test Cohen's κ	0.357	0.082
		Best test RR	66.7%	
		NEC for best test	0.500	
	unbalanced	Training RR	58.3%	0.0%
		Test RR	57.1%	0.0%
		Clinical RR	40.9%	12.7%
		Test Cohen's κ	0.364	0.000
		Best test RR	57.1%	
		NEC for best test	0.500	
N3	balanced	Training RR	57.9%	3.0%
		Test RR	44.6%	11.9%
		Clinical RR	51.5%	24.4%
		Test Cohen's κ	0.356	0.141
		Best test RR	66.7%	
		NEC for best test	0.428	
	unbalanced	Training RR	56.9%	5.6%
		Test RR	50.7%	8.9%
		Clinical RR	53.7%	11.3%
		Test Cohen's κ	0.410	0.089
		Best test RR	64.3%	
		NEC for best test	0.500	
N4	balanced	Training RR	73.6%	5.0%
		Test RR	54.6%	11.3%
		Clinical RR	58.7%	14.6%
		Test Cohen's κ	0.515	0.124
		Best test RR	75.0%	

		NEC for best test	0.431	
	unbalanced	Training RR	70.3%	4.0%
		Test RR	48.9%	11.2%
		Clinical RR	47.6%	23.6%
		Test Cohen's κ	0.447	0.139
		Best test RR	71.4%	
		NEC for best test	0.454	
N5	balanced	Training RR	69.3%	3.4%
		Test RR	51.7%	7.9%
		Clinical RR	52.0%	10.2%
		Test Cohen's κ	0.471	0.100
		Best test RR	66.7%	
		NEC for best test	0.428	
	unbalanced	Training RR	67.8%	2.4%
		Test RR	56.4%	9.5%
		Clinical RR	57.0%	13.0%
		Test Cohen's κ	0.526	0.099
		Best test RR	78.6%	
		NEC for best test	0.419	
N6	balanced	Training RR	66.5%	3.9%
		Test RR	61.2%	11.6%
		Clinical RR	67.5%	9.0%
		Test Cohen's κ	0.554	0.138
		Best test RR	83.3%	
		NEC for best test	0.486	
	unbalanced	Training RR	67.2%	3.4%
		Test RR	55.0%	11.1%
		Clinical RR	56.7%	15.4%
		Test Cohen's κ	0.493	0.130
		Best test RR	78.6%	
		NEC for best test	0.419	
N7	balanced	Training RR	64.1%	4.3%
		Test RR	57.5%	14.5%
		Clinical RR	61.2%	16.5%
		Test Cohen's κ	0.501	0.174
		Best test RR	91.7%	
		NEC for best test	0.500	

	unbalanced	Training RR	61.0%	4.4%
		Test RR	59.6%	13.6%
		Clinical RR	62.0%	17.2%
		Test Cohen's κ	0.520	0.148
		Best test RR	85.7%	
		NEC for best test	0.500	
N8	balanced	Training RR	62.7%	3.0%
		Test RR	65.4%	12.2%
		Clinical RR	69.9%	14.9%
		Test Cohen's κ	0.591	0.141
		Best test RR	91.7%	
		NEC for best test	0.500	
	unbalanced	Training RR	64.0%	2.3%
		Test RR	63.9%	11.7%
		Clinical RR	68.2%	16.8%
		Test Cohen's κ	0.546	0.128
		Best test RR	85.7%	
		NEC for best test	0.500	

Table D.6: Results in the reduction process for the whole feature group ensemble. The abbreviations for the feature sets correspond to the marks in figure 7.6. Furthermore we abbreviate recognition rate by RR.

D.3 Test results for comparison with pure Euclidean LVQ

The following tables allow the comparison of the test results for the vb-KLVQ and the LVQ. The detailed test results for the LVQ runs are given in table D.7. For comparison a choice of evaluation measures is summarized for both vb-KLVQ and LVQ respectively in table D.8.

Feature group selection	Setting	Type of evaluation value	average value	standard deviation of value
R1	balanced	Training RR	68.1%	4.1%
		Test RR	52.5%	12.1%
		Clinical RR	51.9%	17.6%
		Test Cohen's κ	0.477	0.130
		Best test RR	83.3%	
		NEC for best test	0.534	
	unbalanced	Training RR	66.0%	3.5%
		Test RR	49.3%	6.5%
		Clinical RR	51.1%	4.6%
		Test Cohen's κ	0.427	0.082
		Best test RR	64.3%	
		NEC for best test	0.502	
R2	balanced	Training RR	62.4%	3.2%
		Test RR	56.3%	11.7%
		Clinical RR	60.6%	3.6%
		Test Cohen's κ	0.503	0.130
		Best test RR	83.3%	
		NEC for best test	0.391	
	unbalanced	Training RR	63.5%	2.6%
		Test RR	55.0%	8.4%
		Clinical RR	49.6%	16.3%
		Test Cohen's κ	0.475	0.082
		Best test RR	71.4%	
		NEC for best test	0.411	

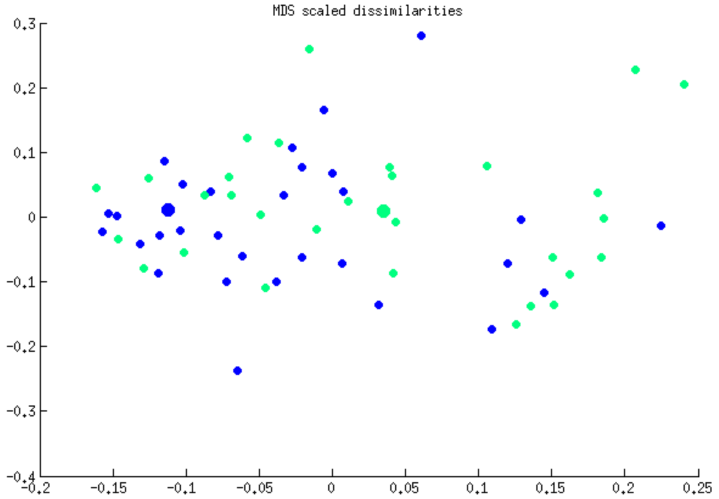
Table D.5: *Results in the reduction process for the random selection. The abbreviations for the feature sets correspond to the marks in figure 7.5. Furthermore we abbreviate recognition rate by RR.*

D.4 Test results for reduction process using only samples of grading two

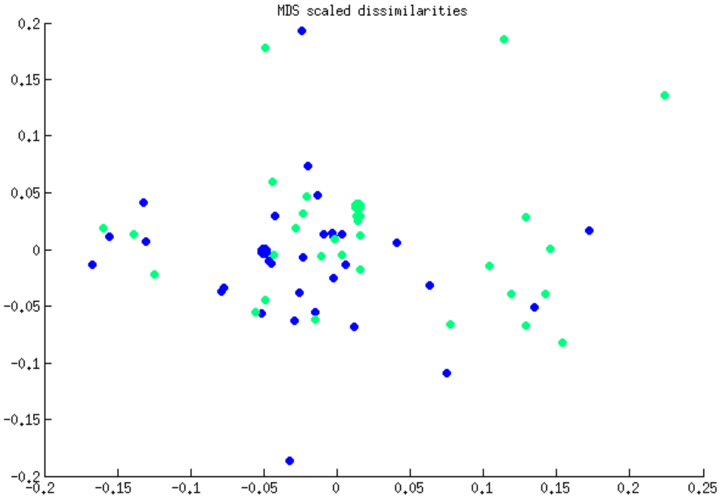
The detailed test results for the reduction process that was conducted using only samples that showed a grading value of two are given in table D.9.

Feature group selection	Type of evaluation value	average value	standard deviation of value
No selection (N1)	Training RR	79.1%	4.7%
	Test RR	50.4%	9.9%
	Clinical RR	56.3%	20.0%
	Test Cohen's κ	0.428	0.113
Pathological (P2)	Training RR	63.5%	3.5%
selection	Test RR	54.6%	11.6%
	Clinical RR	52.7%	18.4%
	Test Cohen's κ	0.457	0.126
Computational (C2)	Training RR	70.2%	3.2%
selection	Test RR	57.9%	12.2%
	Clinical RR	58.7%	17.1%
	Test Cohen's κ	0.517	0.146
Random (R2)	Training RR	61.7%	4.1%
selection	Test RR	58.8%	16.1%
	Clinical RR	57.8%	18.0%
	Test Cohen's κ	0.533	0.173

Table D.7: *Results for pure Euclidean analysis of the reduced feature group selections resulting from the first unbalanced test series. We abbreviate the recognition by RR.*



(a) Multidimensional scaling to two dimensions for lDPP of the training data points using stress function.



(b) Multidimensional scaling to two dimensions for lDPP of the training data points using metric stress function.

Figure D.1: *Labeled DPPs for the best result in integrative relevance analysis over the training data samples.*

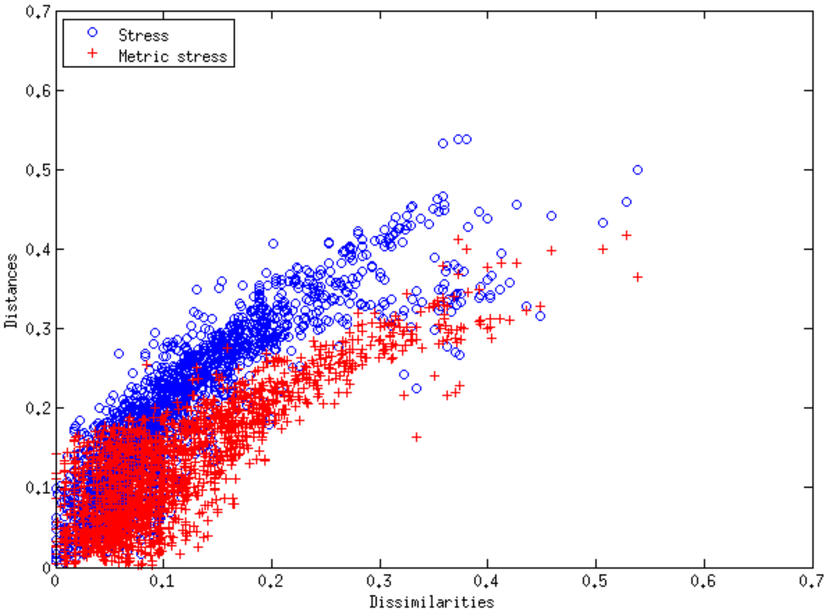


Figure D.2: *Shepard plot for the evaluation of the corresponding mappings in figure D.1.*

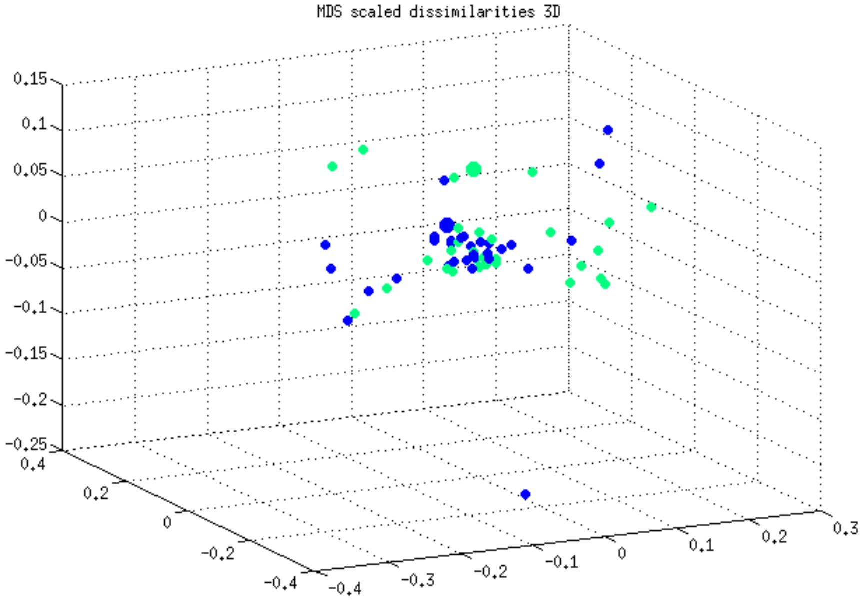


Figure D.3: *Multidimensional scaling to three dimensions for a lDPP of the training data point using classical multidimensional scaling.*

Feature group selection	Type of value to compare	value for GLVQ	value for vb-KLVQ
No selection (N1)	Training RR	79.1%	56.3%
	Test RR	50.4%	52.5%
	Clinical RR	56.3%	46.7%
	Test Cohen's κ	0.428	0.386
Pathological (P2)	Training RR	63.5%	59.4%
selection	Test RR	54.6%	53.8%
	Clinical RR	52.7%	53.0%
	Test Cohen's κ	0.457	0.438
Computational (C2)	Training RR	70.2%	67.1%
selection	Test RR	57.9%	66.7%
	Clinical RR	58.7%	65.8%
	Test Cohen's κ	0.517	0.609
Random (R2)	Training RR	61.7%	62.4%
selection	Test RR	58.8%	56.3%
	Clinical RR	57.8%	60.6%
	Test Cohen's κ	0.533	0.503

Table D.8: *Results comparing pure Euclidean GLVQ with vb-KLVQ for the reduced feature group sets resulting from the first unbalanced test series.*

Feature group selection	Setting	Type of evaluation value	average value	standard deviation of value
G1	balanced	Training RR	66.5%	14.0%
		Test RR	55.0%	23.8%
		Test Cohen's κ	0.456	0.301
	unbalanced	Training RR	55.4%	16.0%
		Test RR	42.9%	21.7%
		Test Cohen's κ	0.336	0.189
G2	balanced	Training RR	79.0%	5.1%
		Test RR	57.5%	24.5%
		Test Cohen's κ	0.525	0.304
G3	balanced	Training RR	72.7%	5.8%
		Test RR	55.0%	22.4%
		Test Cohen's κ	0.485	0.290
G4	balanced	Training RR	73.7%	6.2%
		Test RR	58.8%	24.7%
		Test Cohen's κ	0.537	0.311
G5	balanced	Training RR	76.9%	5.8%
		Test RR	58.8%	20.3%
		Test Cohen's κ	0.558	0.227
G6	balanced	Training RR	62.9%	4.5%
		Test RR	65.0%	23.5%
		Test Cohen's κ	0.599	0.266

Table D.9: *Results in the reduction process for the whole feature group ensemble using only samples with grading two. The abbreviations for the feature sets correspond to the marks in figure 7.14. Furthermore we abbreviate recognition rate by RR.*

Bibliography

Aitchison, J. and Aitken, C. G. G.: 1976, Multivariate binary discrimination by the kernel method, *Biometrika* **63**, 413–420.

Alhoniemi, E., Himberg, J., Parhankangas, J. and Vesanto, J.: n.d., Som toolbox, http://www.cis.hut.fi/projects/somtoolbox/.

Arnonkijpanich, B., Hasenfuss, A. and Hammer, B.: 2011, Local matrix adaptation in topographic neural maps, *Neurocomputing* **74**(4), 522 – 539.

Arnonkijpanich, B. and Hammer, B.: 2010, Global coordination based on matrix neural gas for dynamic texture synthesis, *Artificial Neural Networks in Pattern Recognition*, Springer-Verlag, Berlin, Heidelberg, pp. 84–95.

Bezdek, J. C.: 1974, Cluster validity with fuzzy sets, *Cybernetics and Systems* .

Bezdek, J. C.: 1981, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA.

Biehl, M., Hammer, B. and Schneider, P.: 2006, Matrix learning in learning vector quantization, *Technical report*, Clausthal University of Technology, IfI-06-14.

Biehl, M. and Schwarze, H.: 1993, Learning drifting concepts with neural networks, *Journal of Physics A: Mathematical and General* **26**, 2651–2665.

Bishop, C. M.: 2007, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. 2006. corr. 2nd printing edn, Springer. http://www.worldcat.org/isbn/0387310738.

Bojer, T., Hammer, B., Schunk, D. and von Toschanowitz, K. T.: 2001, Relevance determination in learning vector quantization, *Proc. of European Symposium on Artificial Neural Networks*.

Bootkrajang, J. and Kabán, A.: 2011, Multi-class classification in the presence of labelling errors, in *ESANN* (*ESANN 2011, 19th European Symposium on Artificial Neural Networks, Bruges, Belgium, 2011, Proceedings* 2011).

Bornemeier, J.: 2011, *Entwicklung von merkmalen zur bestimmung räumlicher ausbreitungsmuster in histopathologischen gewebeschnitten des mammakarzinoms*, Master's thesis, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau.

Bouveyron, C. and C., B.: 2011, Probabilistic fisher discriminant analysis, in *ESANN* (*ESANN 2011, 19th European Symposium on Artificial Neural Networks, Bruges, Belgium, 2011, Proceedings* 2011).

Buja, A., Swayne, D. F., Littman, M. L., Dean, N., Hofmann, H. and Chen, L.: 2008, Data visualization with multidimensional scaling, *Journal Of Computational And Graphical Statistics* **17**(2), 444–472. http://pubs.amstat.org/doi/abs/10.1198/106186008X318440.

Chapelle, O., Schölkopf, B. and Zien, A. (eds): 2006, *Semi-Supervised Learning*, MIT Press, Cambridge, MA. http://www.kyb.tuebingen.mpg.de/ssl-book.

Chawla, N.: 2005, Data mining for imbalanced datasets: An overview, *in* O. Maimon and L. Rokach (eds), *Data Mining and Knowledge Discovery Handbook*, Springer US, pp. 853–867.

Chen, L. and Buja, A.: 2009, Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis, *Journal of the American Statistical Association* **104**(485), 209–219. http://econpapers.repec.org/RePEc:bes:jnlasa:v:104:i:485:y:2009:p:209-219.

Chen, N. and Marques, N. C.: 2010, Extending learning vector quantization for classifying data with categorical values, *in* J. Filipe, A. Fred and B. Sharp (eds), *Agents and Artificial Intelligence*, Vol. 67 of *Communications in Computer and Information Science*, Springer Berlin Heidelberg, pp. 124–136.

Cichocki, A., Zdunek, R., Phan, A. H. and Amari, S.-I.: 2009, *Nonnegative Matrix and Tensor Factorizations*, John Wiley & Sons, Ltd. http://dx.doi.org/10.1002/9780470747278.

Cohen, J.: 1960, A Coefficient of Agreement for Nominal Scales, *Educational and Psychological Measurement* **20**(1), 37–46.

Cohen, J.: 1972, Weighted Chi Square: an Extension of the Kappa Method, *Educational and Psychological Measurement* **32**(1), 61–74.

Collins, F. S. and Barker, A. D.: 2007, Ein atlas des krebsgenoms, *Spektrum der Wissenschaft* **2**(11/07), 40 – 53.

Corsini, P., Lazzerini, B. and Marcelloni, F.: 2006, Combining supervised and unsupervised learning for data clustering, *Neural Computing & Applications* **15**, 289–297. http://dx.doi.org/10.1007/s00521-006-0030-5.

Cottrell, M., Hammer, B., Hasenfuß, A. and Villmann, T.: 2006, Batch and median neural gas, *Neural Netw.* **19**(6), 762–771.

Courrieu, P.: 2002, Straight monotonic embedding of data sets in euclidean spaces., *Neural Networks* pp. 1185–1196.

Crammer, K., Gilad-bachrach, R., Navot, A. and Tishby, N.: 2002, Margin analysis of the lvq algorithm, *In: Advances in Neural Information Processing Systems 2002*, MIT press, pp. 462–469.

Detrano, R.: 1989, International application of a new probability algorithm for the diagnosis of coronary artery disease, *American Journal of Cardiology* **64**, 304–310.

Dou, W., Ren, Y., Wu, Q., Ruan, S., Chen, Y., Bloyet, D. and Constans, J.-M.: 2007, Fuzzy kappa for the agreement measure of fuzzy classifications, *Neurocomputing* **70**(4-6), 726 – 734. Advanced Neurocomputing Theory and Methodology - Selected papers from the International Conference on Intelligent Computing 2005 (ICIC 2005).

Duda, R., Hart, P. and Stork, D.: 2001, *Pattern Classification (2nd Edition)*, 2 edn, Wiley-Interscience. http://www.worldcat.org/isbn/0471056693.

Dunn, J. C.: 1973, A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, *Journal of Cybernetics* **3**(3), 32–57. http://dx.doi.org/10.1080/01969727308546046.

ESANN 2011, 19th European Symposium on Artificial Neural Networks, Bruges, Belgium, 2011, Proceedings: 2011.

Fabbri, R., da F. Costa, L., Torelli, J. C. and Bruno, O. M.: 2008, 2d euclidean distance transform algorithms: A comparative survey, *ACM Computing Surveys* **40**(1), 1–44.

Flatla, D. and Gutwin, C.: 2010, Individual models of color differentiation to improve interpretability of information visualization, *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, Atlanta, Georgia, USA, pp. 2563–2572. Honorable Mention Award.

Fleiss, J. L., Levin, B., Paik, M. C. and Fleiss, J.: 2003, *Statistical Methods for Rates & Proportions*, 3rd edn, Wiley-Interscience.

Fornefett, M., Rohr, K. and Stiehl, H. S.: 1999, Elastic registration of medical images using radial basis functions with compact support, *CVPR*, IEEE Computer Society, pp. 1402–.

Frank, A. and Asuncion, A.: 2010, UCI machine learning repository. http://archive.ics.uci.edu/ml.

Frey, B. J. and Dueck, D.: 2007, Clustering by Passing Messages Between Data Points, *Science* **315**(5814), 972–976. http://dx.doi.org/10.1126/science.1136800.

Fritzke, B.: 1995, A growing neural gas network learns topologies, *Advances in Neural Information Processing Systems 7*, MIT Press, pp. 625–632.

Fukuyama, Y. and Sugeno, M.: 1989, A new method of choosing the number of clusters for fuzzy c-means method, *Proc. 5th Fuzzy Syst. Symp.*, pp. 247 – 250.

Galea, M., Blamey, R., Elston, C. and Ellis, I.: 1992, The nottingham prognostic index in primary breast cancer, *Breast Cancer Research and Treatment* **22**, 207–219. http://dx.doi.org/10.1007/BF01840834.

García-Borroto, M. and Ruiz-Shulcloper, J.: 2005, Selecting prototypes in mixed incomplete data, *in* A. Sanfeliu and M. Cortés (eds), *Progress in Pattern Recognition, Image Analysis and Applications*, Vol. 3773 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 450–459.

Geweniger, T., Kästner, M. and Villmann, T.: 2011, Optimization of parameterized divergences in fuzzy c-means, in *ESANN* (*ESANN 2011, 19th European Symposium on Artificial Neural Networks, Bruges, Belgium, 2011, Proceedings* 2011).

Geweniger, T., Zühlke, D., Hammer, B. and Villmann, T.: 2009, Fuzzy variant of affinity propagation in comparison to median fuzzy c-means, *Proceedings of the 7th International Workshop on Advances in Self-Organizing Maps*, WSOM '09, Springer-Verlag, Berlin, Heidelberg, pp. 72–79. http://dx.doi.org/10.1007/978-3-642-02397-2_9.

Geweniger, T., Zühlke, D., Hammer, B. and Villmann, T.: 2010, Median fuzzy c-means for clustering dissimilarity data, *Neurocomputing* **73**(7-9), 1109 – 1116. Advances in Computational Intelligence and Learning - 17th European Symposium on Artificial Neural Networks 2009, 17th European Symposium on Artificial Neural Networks 2009. http://www.sciencedirect.com/science/article/B6V10-4Y70C4S-1/2/fb8e7dd1d23b33e05d376467492145c6

Globerson, A., Chechik, G., Pereira, F., Tishby, N. and Lafferty, J.: 2005, Euclidean embedding of co-occurrence data, *Advances in Neural Information Processing Systems 17*, MIT Press, pp. 497–504.

Grivennikov, S. I., Greten, F. R. and Karin, M.: 2010, Immunity, inflammation, and cancer., *Cell* **140**(6), 883–899. http://www.ncbi.nlm.nih.gov/pubmed/20303878.

Hadamard, J.: 1902, Sur les problèmes aux dérivés partielles et leur signification physique, *Princeton University Bulletin* **13**, 49–52.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M.: 2001, On clustering validation techniques, *Journal of Intelligent Information Systems* **17**, 107–145. http://dx.doi.org/10.1023/A:1012801612483.

Hammer, B. and Hasenfuss, A.: 2007, Relational neural gas, *KI '07: Proceedings of the 30th annual German conference on Advances in Artificial Intelligence*, Springer-Verlag, Berlin, Heidelberg, pp. 190–204.

Hammer, B. and Villmann, T.: 2002, Generalized relevance learning vector quantization, *Neural Networks* **15**, 1059–1068.

Hammer, B. and Villmann, T.: 2007, How to process uncertainty in machine learning, *ESANN*.

Handl, J., Knowles, J. and Kell, D. B.: 2005, Computational cluster validation in post-genomic data analysis., *Bioinformatics (Oxford, England)* **21**(15), 3201–3212. http://dx.doi.org/10.1093/bioinformatics/bti517.

Haralick, R. M., Shanmugam, K. and Dinstein, I.: 1973, Textural features for image classification, *Systems, Man and Cybernetics, IEEE Transactions on* **3**(6), 610–621.

Hastie, T., Tibshirani, R. and Friedman, J. H.: 2003, *The Elements of Statistical Learning*, Springer. http://www.worldcat.org/isbn/0387952845.

Heskes, T.: 2001, Self-organizing maps, vector quantization, and mixture modeling, *IEEE Transactions on Neural Networks* **12**, 1299–1305.

Hill, D. L. G., Batchelor, P. G., Holden, M. and Hawkes, D. J.: 2001, Medical image registration, *Physics in Medicine and Biology* **46**(3), R1. http://stacks.iop.org/0031-9155/46/i=3/a=201.

Horsfall, D. J., Jarvis, L. R., Grimbaldeston, M. A., Tilley, W. D. and Orell, S. R.: 1989, Immunocytochemical assay for oestrogen receptor in fine needle aspirates of breast cancer by video image analysis., *Br J Cancer* **59**(1), 129–34. http://www.biomedsearch.com/nih/Immunocytochemical-assay-oestrogen-receptor-in/2547412.html.

Huang, Z.: 1998, Extensions to the k-means algorithm for clustering large data sets with categorical values, *Data Min. Knowl. Discov.* **2**, 283–304. http://dx.doi.org/10.1023/A:1009769707641.

humanIT Software GmbH: n.d., Infozoom, http://www.infozoom.com/nc/en/home.html.

Jaccard, P.: 1901, Étude comparative de la distribution florale dans une portion des Alpes et des Jura, *Bulletin del la Société Vaudoise des Sciences Naturelles* **37**, 547–579.

Jahkola, T., Toivonen, T., Virtanen, I., von Smitten, K., Nordling, S., von Boguslawski, K., Haglund, C., Nevanlinna, H. and Blomqvist, C.: 1998, Tenascin-c expression in invasion border of early breast cancer: a predictor of local and distant recurrence., *Br J Cancer* **78**(11), 1507–13.

Jarque, C. M. and Bera, A. K.: 1980, Efficient tests for normality, homoscedasticity and serial independence of regression residuals, *Economics Letters* **6**(3), 255–259. http://ideas.repec.org/a/eee/ecolet/v6y1980i3p255-259.html.

Kato, T.: 1950, On the adiabatic theorem of quantum mechanics, *Journal of the Physical Society of Japan* **5**(6), 435–439. http://jpsj.ipap.jp/link?JPSJ/5/435/.

Khabirova, E.: 2011, *Image processing descriptors for inner tumor growth patterns*, Master's thesis, Bonn-Aachen International Center for Information Technology (B-IT), University of Bonn.

Kim, D.-W., Lee, K. H. and Lee, D.: 2003, Fuzzy cluster validation index based on inter-cluster proximity, *Pattern Recogn. Lett.* **24**, 2561–2574. http://dx.doi.org/10.1016/S0167-8655(03)00101-6.

Klipp, E., Liebermeister, W., Wierling, C., Kowald, A. and Lehrach, H.: 2009, *Systems biology: a textbook*, Wiley-VCH. http://books.google.com/books?id=HMVwy6urHb4C.

Kohonen, T.: 1986, Learning vector quantization for pattern recognition, *Technical Report TKK-F-A601*.

Kohonen, T., Schroeder, M. R. and Huang, T. S. (eds): 2001, *Self-Organizing Maps*, 3rd edn, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Kushner, H. J. and Clark, D. S.: 1978, *Stochastic approximation methods for constrained and unconstrained systems / Harold J. Kushner, Dean S. Clark*, Springer-Verlag, New York.

Lancaster, H. O.: 1949, The combination of probabilities arising from data in discrete distributions, *Biometrika* **36**, 370–382.

Langevin, A. and Riopel, D.: 2005, *Logistics Systems: Design and Optimization*, Springer Science Business Media, New York. http://www.springer.com/business+%26+management/production/book/978-0-387-24971-1.

Lee, J. A. and Verleysen, M.: 2005, M.: Generalization of the lp norm for time series and its application to self-organizing maps, *Proc. of Workshop on Self-Organizing Maps (WSOM) 2005.*, COTTRELL, M., Paris, Sorbonne, pp. 733–740.

Levasseur, C., Mayer, U. F. and Kreutz-delgado, K.: 2009, Classifying non-gaussian and mixed data sets in their natural parameter space, *IEEE International Workshop on Machine Learning for Signal Processing, 2009.*, IEEE, pp. 1–6.

Li, C. and Biswas, G.: 2002, Unsupervised learning with mixed numeric and nominal data, *IEEE Trans. on Knowl. and Data Eng.* **14**, 673–690. http://dx.doi.org/10.1109/TKDE.2002.1019208.

Lincoff, G.: 1981, *The Audubon Society Field Guide to North American Mushrooms*, Alfred A. Knopf, New York.

Luo, H., Kong, F. and Li, Y.: 2006, *Clustering mixed data based on evidence accumulation (Lecture Notes in Computer Science)*, Vol. 4093, Springer-Verlag New York, Inc.

Manning, C. D., Raghavan, P. and Schütze, H.: 2008, *An Introduction to Information Retrieval*, Press, Cambridge U.

Martinetz, T., Berkovich, S. and Schulten, K.: 1993, "Neural-gas" Network for Vector Quantization and its Application to Time-Series Prediction, *IEEE-Transactions on Neural Networks* **4**(4), 558–569.

Martinetz, T. and Schulten, K.: 1991, A "neural-gas" network learns topologies, *Artificial Neural Networks* **I**, 397–402.

Mathworks: n.d.a, Bioinformatics toolbox, http://www.mathworks.de/products/bioinfo/.

Mathworks: n.d.b, Image processing toolbox, http://www.mathworks.de/help/toolbox/images/.

Mathworks: n.d.c, Statistics toolbox, http://www.mathworks.de/help/toolbox/stats/.

Mcallester, D. A.: 2003, Pac-bayesian stochastic model selection, *Machine Learning*, p. 2003.

Mwebaze, E., Schneider, P., Schleif, F.-M., Aduwo, J., Quinn, J., Haase, S., Villmann, T. and Biehl, M.: 2011, Divergence-based classification in learning vector quantization, *Neurocomputing* **74**(9), 1429 – 1435. Advances in artificial neural networks, machine learning, and computational intelligence - - Selected papers from the 18th European Symposium on Artificial Neural Networks (ESANN 2010).
http://www.sciencedirect.com/science/article/B6V10-5276T2N-6/2/8166687d2cd551558553a1b28c540ebf

Olson, A. H.: 2007, Image analysis using the aperio scanscope$^{\mathrm{TM}}$. http://www.quorumtechnologies.com/pdf_files/AnalysisWhitePaper.pdf.

Otsu, N.: 1979, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man and Cybernetics* **9**(1), 62–66.

Paclík, P. and Duin, R. P. W.: 2003, Dissimilarity-based classification of spectra: computational issues, *Real-Time Imaging* **9**(4), 237 – 244. Special Issue on Spectral Imaging.
http://www.sciencedirect.com/science/article/B6WPR-49VC6XV-2/2/eda0e2fe3b8698d9a3635e39c4cba23b

Pal, N. R., Pal, K., Keller, J. M. and Bezdek, J. C.: 2005, A possibilistic fuzzy c-means clustering algorithm, *IEEE T. Fuzzy Systems* **13**(4), 517–530.

Papari, G., Bunte, K. and Biehl, M.: 2011, Waypoint averaging and step size control in learning by gradient descent, *MIWOCI 2011, 3rd Mittweida Workshop on Computational Intelligence*, Machine Learning Reports, pp. 16–26.

Pękalska, E. and Duin, R. P. W.: 2006, *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*, Machine Perception & Artifical Intelligence, World Scientific Pub Co.

Pękalska, E. and Duin, R. P. W.: 2009, The dissimilarity representation for pattern recognition, a tutorial, Tutorial.

Pearl, J.: 1988, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Pluim, J. P. W., Maintz, J. B. A. and Viergever, M. A.: 2003, Mutual information based registration of medical images: A survey., *IEEE Trans. Med. Imaging* pp. 986–1004.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P.: 2007, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3 edn, Cambridge University Press.

Prim, R. C.: 1957, Shortest connection networks and some generalizations, *Bell System Technology Journal* **36**, 1389–1401.

Principe, J., Xu, D. and Fisher, J.: 2000, Information Theoretic Learning, *in* S. Haykin (ed.), *Unsupervised Adaptive Filtering*, John Wiley & Sons, New York.

Qin, A. and Suganthan, P.: 2004, Growing generalized learning vector quantization with local neighborhood adaptation rule, *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference*, IEEE Press, pp. 524–529.

Qinand, A. K. and Suganthan, P. N.: 2004, A novel kernel prototype-based learning algorithm, *Pattern Recognition, International Conference on* **4**, 621–624.

Randell, D. A., Cui, Z. and Cohn, A. G.: 1992, A spatial logic based on regions and connection, *PROCEEDINGS 3RD INTERNATIONAL CONFERENCE ON KNOWLEDGE REPRESENTATION AND REASONING*.

Rangayyan, R. M., El-Faramawy, N. M., Desautels, J. E. L. and Alim, O. A.: 1997, Measures of acutance and shape for classification of breast tumors., *IEEE Trans. Med. Imaging* pp. 799–810.

Rexhepaj, E., Brennan, D., Holloway, P., Kay, E., McCann, A., Landberg, G., Duffy, M., Jirstrom, K. and Gallagher, W.: 2008, Novel image analysis approach for quantifying expression of nuclear proteins assessed by immunohistochemistry: application to measurement of oestrogen and progesterone receptor levels in breast cancer, *Breast Cancer Research* **10**(5), R89.

Rousseeuw, P.: 1987, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics* **20**(1), 53–65. http://dx.doi.org/10.1016/0377-0427(87)90125-7.

Sachs, L.: 2006, *Angewandte Statistik*, Vol. 12, Springer.

Sato, A. and Yamada, K.: 1996, Generalized learning vector quantization, *Advances in Neural Information Processing Systems 8*, MIT Press, Cambridge, MA, USA, pp. 423–429.

Schleif, F.-M., Villmann, T., Hammer, B., Schneider, P. and Biehl, M.: 2010, Generalized derivative based kernelized learning vector quantization., *IDEAL'10*, pp. 21–28.

Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Müller, K. R., Ratsch, G. and Smola, A. J.: 1999, Input space versus feature space in kernel-based methods, *Neural Networks, IEEE Transactions on* **10**(5), 1000–1017. http://dx.doi.org/10.1109/72.788641.

Schneider, P.: 2010, *Advanced methods for prototype-based classification*, PhD thesis, University of Groningen.

Schneider, P., Biehl, M. and Hammer, B.: 2009, Adaptive relevance matrices in learning vector quantization, *Neural Comput.* **21**, 3532–3561. http://dx.doi.org/10.1162/neco.2009.11-08-908.

Smeulders, A. W. M., Member, S., Worring, M., Santini, S., Gupta, A. and Jain, R.: 2000, Content-based image retrieval at the end of the early years, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 1349–1380.

Strickert, M., Labitzke, B., Kolb, A. and Villmann, T.: 2011, Multispectral image characterization by partial generalized covariance, in *ESANN* (*ESANN 2011, 19th European Symposium on Artificial Neural Networks, Bruges, Belgium, 2011, Proceedings* 2011), pp. 105–110.

Tannock, I. F.: 2001, Tumor physiology and drug resistance, *Cancer and Metastasis Reviews* **20**, 123–132. http://dx.doi.org/10.1023/A:1013125027697.

van der Heijden, F., Duin, R. P. W., de Ridder, D. and Tax, D.: 2004, *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using Matlab*, Wiley, New York. http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470090138.html.

Villmann, T., Geweniger, T., Kästner, M. and Lange, M.: 2011, Theory of fuzzy neural gas for unsupervised vector quantization, *Technical report*, University of Applied Sciences Mittweida, Machine Learning Reports MLR-2011-06.

Villmann, T. and Haase, S.: 2010, Mathematical aspects of divergence based vector quantization using frechet-derivatives, *Machine Learning Reports 01/2007*. http://www.uni-leipzig.de/~compint/mlr/mlr_01_2010.pdf.

Villmann, T. and Haase, S.: 2011, Divergence-based vector quantization, *Neural Computation* **23**(5), 1343–1392.

Villmann, T. and Hammer, B.: 2009, Functional principal component learning using oja's method and sobolev norms, *in* J. Príncipe and R. Miikkulainen (eds), *Advances in Self-Organizing Maps*, Vol. 5629 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 325–333.

von Luxburg, U.: 2007, A tutorial on spectral clustering, *Statistics and Computing* **17**(4), 395 – 416.

Vovk, V.: 2005, *Algorithmic Learning in a Random World*, Springer, Berlin.

Wendland, H.: 1995, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Adv. Comput. Math.* **4**(1), 389–396.

Widmer, G. and Kubat, M.: 1996, Learning in the presence of concept drift and hidden contexts, *Machine Learning*, pp. 69–101.

Witten, I. H. and Frank, E.: 2005, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Series in Data Management Systems, second edn, Morgan Kaufmann. http://www.worldcat.org/isbn/0120884070.

Xie, X. L. and Beni, G.: 1991, A validity measure for fuzzy clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**, 841–847.

Yang, J. Y. and Qu Yang, M.: 2006, Assessing protein function using a combination of supervised and unsupervised learning, *Proceedings of the Sixth IEEE Symposium on BionInformatics and BioEngineering*, BIBE '06, IEEE Computer Society, Washington, DC, USA, pp. 35–44. http://portal.acm.org/citation.cfm?id=1169220.1169380.

Zhang, D. and Lu, G.: 2002, Shape-based image retrieval using generic fourier descriptor, *Signal Processing: Image Communication* **17**(10), 825 – 848. http://www.sciencedirect.com/science/article/pii/S092359650200084X

Zühlke, D., Geweniger, T., Heimann, U. and Villmann, T.: 2009, Fuzzy fleiss-kappa for comparison of fuzzy classifiers, *ESANN*.

Zühlke, D., Schleif, F.-M., Geweniger, T. and Villmann, T.: 2010, Learning vector quantization for heterogeneous structured data, *ESANN*.