Re-using Data Mining Workflows

Stefan Rüping, Dennis Wegener, and Philipp Bremer

Fraunhofer IAIS, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
`http://www.iais.fraunhofer.de`

Abstract. Setting up and reusing data mining processes is a complex task. Based on our experience from a project on the analysis of clinico-genomic data we will make the point that supporting the setup and reuse by setting up large workflow repositories may not be realistic in practice. We describe an approach for automatically collecting workflow information and meta data and introduce data mining patterns as an approach for formally describing the necessary information for workflow reuse.

Keywords: Data Mining, Workflow Reuse, Data Mining Patterns

1 Introduction

Workflow enacting systems are a popular technology in business and e-science alike to flexibly define and enact complex data processing tasks. A workflow is basically a description of the order in which a set of services have to be called with which input in order to solve a given task. Since the construction of a workflow for a specific task can become quite complex, efforts are currently underway to increase the reuse of workflows through the implementation of specialized workflow repositories. Driven by specific applications, a large collection of workflow systems have been prototyped, such as Taverna [1] or Triana [2].

The next generation of workflow systems are marked by workflow repositories such as MyExperiment.org, which tackle the problem of organizing workflows by offering the research community the possibility to publish, exchange and discuss individual workflows.

However, the more powerful these environments become, the more important it is to guide the user in the complex task of constructing appropriate workflows. This is particularly true for the case of workflows which encode a data mining tasks, which are typically much more complex and in a more constant state of frequent change than workflows in business applications.

In this paper, we are particularly interested in the question of reusing successful data mining applications. As the construction of a good data mining process invariably requires to encode a significant amount of domain knowledge, this is a process which cannot be fully automated. By reusing and adapting existing processes that have proven to be successful in practical use, we hope to be able to save much of this manual work in a new application and thereby increase the efficiency of setting up data mining workflows.

We report our experiences in designing a system which is targeted at supporting scientists, in this case bioinformaticians, with a workflow system for the analysis of clinico-genomic data. We will make the case that:

- For practical reasons it is already a difficult task to gather a non-trivial database of workflows which can form the basis of workflow reuse.
- In order to be able to meaningfully reuse data mining workflows, a formal notation is needed that allows to flexibly express both technical information about the implementation of workflows, and high-level semantic information about the purpose and pre-requisites of a workflow.

The paper is structured as follows: In the next section, we introduce the ACGT project, in the context of which our work was developed. Section 3 describes an approach for automatically collecting workflow information and appropriate meta data. Section 4 presents data mining patterns which formally describe all information that is necessary for workflow reuse. Section 5 concludes.

2 The ACGT Project

The work in this paper is based on our experiences in the ACGT project[1], which has the goal of implementing a secure, semantically enhanced end-to-end system in support of large multi-centric clinico-genomic trials, meaning that it strives to integrate all steps from the collection and management of various kinds of data in a trial up to the statistical analysis by the researcher. From the technological point of view, ACGT offers a modular environment based on Grid technologies, in which new data processing and data mining services can be integrated as plug-ins as they become available. ACGT also provides a framework for semantic integration of data sources (e.g., clinical databases) and data mining tools, through the use of a specifically developed ontology and of a semantic mediator. In the current version, the various elements of the data mining environment can be integrated into complex analysis pipelines through the ACGT workflow editor and enactor. With respect to workflow reuse, we made the following experiences in setting up and running an initial version of the ACGT environment:

- The construction of data mining workflows is an inherently complex problem when it is based on input data with complex semantics, as it is the case in clinical and genomic data.
- Because of the complex data dependencies, copy and paste is not an appropriate technique for workflow reuse.
- Standardization and reuse of approaches and algorithms works very well on the level of services, but not on the level of workflows. While it is relatively easy to select the right parameterization of a service, making the right connections and changes to a workflow template is quickly getting quite complex, such that user finds it easier to construct a new workflow from scratch.

[1] http://eu-acgt.org

- Workflow reuse only occurs when the initial creator of a workflow detailedly describes the internal logic of the workflow. However, most workflow creators avoid this effort because they simply want to “solve the task at hand”.

In summary, the situation of having a large repository of workflows to chose the appropriate one from, which is often assumed in existing approaches for workflow recommendation systems, may not be very realistic in practice.

3 Collecting Workflow Information

To obtain information about the human creation of data mining workflows it is necessary to design a system which collects realistic data mining workflows out of the production cycle. For the analysis of these workflows also a new representation of connected workflows is needed which includes all workflow versions of the life cycle of the workflow - from the first try to the final workflow version. We developed a system which collects data mining workflows based on plug-ins which were integrated into the data mining software used for production [3]. In particular we developed plug-ins for Rapidminer, which is an open source data mining software, and Obwious, which is developed by Fraunhofer itself. Every time the user executes a workflow, the workflow definition is send to a repository and stored in a shared abstract representation. The shared abstract representation is mandatory as we want to compare the different formats and types of workflows and to extract the interesting information out of a wide range of workflows to get a high diversity. The plug-ins are designed in a way such that they do not need user interaction at all which allows for not disturbing the user in his workflow creation routines. By this we hope for getting most realistic workflows and for avoiding unwanted side effects if the user feels controlled or monitored.

The definition of a representation of connected workflows as workflow design sequence is needed as we do not only want to observe the final version of a humanly created workflow but the whole chain of workflows which were created in the process of finding and creating this final version. We will call the collection of the connected workflows from the workflow life cycle which solves the same data mining problem on the same data base **workflow design sequence**.

Although almost trivial for the creator of the workflow, it is unknown for the system to which design sequence a workflow belongs to as there is no explicit information from the creator of the workflow that can be used. Therefore we had to find a way to solely compute these dependencies. We have done this by comparing the input data, the parameters of the operator which is loading or generating the input or the data itself. We put those workflows in one design sequence, whose input data is exactly the same or quite similar. We showed that with this method we never put a workflow in a wrong sequence but underestimated the length of the real sequences. The length of the computed sequences is about a quarter of the length of the real ones. But, for the intended purpose of analyzing the human creation of data mining workflows, it is better to underestimate the length of the rows than to put wrong workflows in a sequence. The shared abstract representation of the workflows is solely orientated on CRISP-Phases and its common tasks, as described in [4]. Based on this we created the following six classes: (1) data preparation: select data, (2) data preparation: clean data, (3) data preparation: construct data, (4) data preparation: integrate data, (5) data preparation: format data, (6) modeling, and (7) other.

The operators of the data mining software that was used are classified using these classes and the workflows are transferred to the shared abstract representation. The abstract information itself consists of the information if any operator of the first five classes - the operators which are doing data preparation tasks - is used in the workflow, and if any changes in the operator themselves are done or if any changes in the parameter settings are done in comparison to the predecessor in this sequence. Furthermore in this representation it is noted which learner-operators - operators of the class *Modeling* - are used and if there are any changes in the learner-operators or in the parameter setting of the used operators are done in comparison to the predecessor in this sequence. An example of this representation is shown in Figure 1.

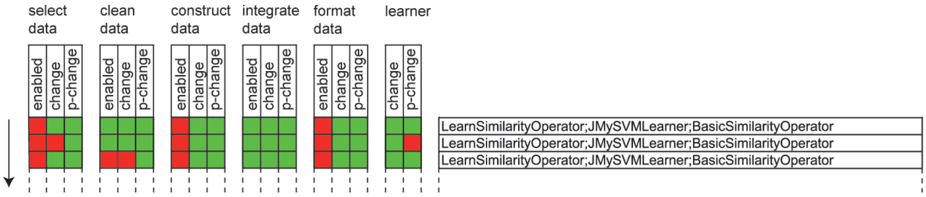
select data
clean data
construct data
integrate data
format data
learner
enabled
change
p-change
LearnSimilarityOperator;JMySVMLearner;BasicSimilarityOperator
LearnSimilarityOperator;JMySVMLearner;BasicSimilarityOperator
LearnSimilarityOperator;JMySVMLearner;BasicSimilarityOperator

Fig. 1. Visualization of a workflow design sequence in the abstract representation

At the end of the first collection phase which lasted 6 months we have collected 2520 workflows in our database which were created by 16 different users. These workflows were processed into 133 workflow design sequences. According to our assumption this would mean that there are about 33 real workflow design sequences in our database. There was an imbalance on the distribution of workflows and workflow design sequences over the two software sources. Because of heavy usage and an early development state of Obwious about 85% of workflows and over 90% of workflow design sequences were created using Rapidminer.

Although there has to be much more time the system collects data there are already some interesting informations in the derived data. In Figure 2 one can see that in the workflow creation process the adjustment and modulation of the data preparation operators is as important as the adjustment and modulation of the learner operators. This is contrarily to common assumptions where the focus is only set on the modeling phase and the learner operators. The average length of a computed workflow design sequence is about 18 workflows. This means, if we reuse our assumption that computed workflow design sequences have a quarter of the workflows that the real workflow design sequences would have, that a human workflow creator produces about 72 workflows until he finds his final version and therefore heavily changes operators and parameters of the CRISP-phases data preparation and modeling.

CRISP-phase		absolute occurrences	relative occurrences[1]
Data preparation	Change	609	24,17%
	Parameter change	405	16,07%
	Sum of all changes	1014	40,24%
Learner	Change	215	8,53%
	Parameter change	801	31,79%
	Sum of all changes	1016	40,32%

[1] Relative to the absolute count of all workflows of 2520

Fig. 2. Occurrences of changes in CRISP-phases

4 Data Mining Patterns

In the area of data mining there exist a lot of scenarios where existing solutions are reusable, especially when no research on new algorithms is necessary. Lots of examples and ready-to-use algorithms are available as toolkits or services, which only have to be integrated. However, the reuse and integration of existing solutions is not often or only informally done in practice due to a lack of formal support, which leads to a lot of unnecessary repetitive work. In the following we present our approach on the reuse of data mining workflows by formally encoding both technical and and high-level semantics of these workflows.

In [5] we presented a new process model for easy reuse and integration of data mining in different business processes. The aim of this work was to allow for reusing existing data mining processes that have proven to be successful. Thus, we aimed at the development of a formal and concrete definition of the steps that are involved in the data mining process and of the steps that are necessary to reuse it in new business processes. Our approach is based on CRISP [4] and includes the definition of Data Mining Patterns, a definition of a hierarchy of tasks to guide the specialization of abstract patterns to concrete processes, and a meta-process for applying patterns to business processes. The patterns provide a simple formal description for the reuse and integration of data mining.

CRISP is a standard process model for data mining which describes the life cycle of a data mining project in the following 6 phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. The CRISP model includes a four-level breakdown including phases, generic tasks, specialized tasks and process instances for specifying different levels of abstraction. In the end, the data mining patterns match most to the process instance level of CRISP. In our approach we need to take into account that reuse may in some cases only be possible at a general or conceptual level. We allow for the specification of different levels of abstraction by the following hierarchy of tasks: **conceptual** (only textual description is available), **configurable** (code is available but parameters need to be specified), and **executable** (code and parameters are specified).

The idea of our approach is to be able to describe all data mining processes. The description needs to be as detailed as adequate for the given scenario. Thus, we consider the tasks of the CRISP process as the most general data mining pattern. Every concretion of this process for an specific application is also a data mining pattern. The generic CRISP tasks can be transformed to the following components: **Check tasks** in the pattern, e.g. checking if the data quality is acceptable; **Configurable tasks** in the pattern, e.g. setting a certain service parameter by hand; **Executable tasks** or **gateways** in the pattern which can be executed without further specification; **Tasks in the meta process** that are independent of a certain pattern, e.g. checking if the business objective of the original data mining process and the new process are identical; **Empty task** as the task is obsolete due to the pattern approach, e.g. producing a final report.

We defined a data mining pattern as follows: The pattern representing the extended CRISP model is a **Data Mining Pattern**. Each concretion of this according to the presented hierarchy is also a Data Mining Pattern. In summary, the approach of [5] is a first step towards support for an automation of a reuse of successful data mining processes. It is based on CRISP and allows for formally describing the information that is necessary for workflow reuse.

5 Conclusion and Future Work

Setting up and reusing data mining workflows is a complex task. When many dependencies on complex data exist, the situation found in workflow reuse is fundamentally different from the one found in reusing services. In this paper, we have given a short insight into the nature of this problem, based on our experience in a project dealing with the analysis of clinico-genomic data. We have proposed two approaches to improve the possibility for reusing workflows, which are the automated collection of a meta data-rich workflow repository, and the definition of data mining patterns to formally encode both technical and high-level semantic of workflows.

References

1. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M., Li, P., Oinn, T.: Taverna: a tool for building and running workflows of services. Nucleic Acids Research, vol. 34, iss. Web Server issue, pp. 729-732 (2006)
2. Taylor, I., Shields, M., Wang, I., Harrison, A..: The triana workflow environment: Architecture and applications, in: I. Taylor, E. Deelman, D. Gannon, M. Shields (Eds.), Workflows for e-Science, pp. 320-339, Springer, New York, Secaucus, NJ, USA (2007)
3. Bremer, P.: Erstellung einer Datenbasis von Workflowreihen aus realen Anwendungen (in german), Diploma Thesis, University of Bonn (2010)
4. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: CRISP-DM 1.0 Step-by-step data mining guide, The CRISP-DM consortium (2000)
5. Wegener, D., Rüping, S.: On Reusing Data Mining in Business Processes - A Pattern-based Approach. Submitted to the 1st International Workshop on Reuse in Business Process Management (rBPM 2010)