TOWARDS MKDA: A DATA MINING SEMANTIC WEB SERVICE

Vincenzo Cannella, Giuseppe Russo and Roberto Pirrone
Universita' degli Studi Palermo, Dipartimento Ingegneria Informatica - DINFO
Viale delle Scienze ed. 6 p.3, 90128 Palermo, Italy

Keywords: Data Mining, Knowledge Discovery in Databases, Semantic Web Service, Medical Knowledge Discovery Assistant, Knowledge Discovery Process.

Abstract: Nowadays a huge amount of raw medical data is generated. These data, analyzed with data mining techniques, could be used to produce new knowledge. Unluckily such tasks need skilled data analysts, and not so many researchers in medical field are also data mining experts. In this paper we present a web based system for knowledge discovery assistance in Medicine able to advice a medical researcher in this kind of tasks. The experiment specifications are expressed in a formal language we have defined. The system GUI helps the user in the their composition. The system plans a Knowledge Discovery Process (KDP). The KDP is designed on the basis of rules in a knowledge base. Finally the system executes the KDP and produces a model as result. The system works through the co-operation of different web services specialized in different tasks. The choice of web services is based on the semantic of their functionalities, according to a common OWL ontology. The system is still under development.

1 INTRODUCTION

In recent years the availability of huge medical data collections has sometimes dramatically brought to light the (in)ability to analyze them. Medical centers have huge databases containing therapies, diagnoses and personal data of their patients. Moreover, the automatic devices of relevant data acquisition, such as MRI and PET, extract more and more accurate medical images of patients. Medical images are produced in such a number that they can only be analyzed with the help of complex systems. All these raw data could be usefully investigated by medical researchers to find new knowledge. In this view a very important application field is the Knowledge Discovery in Databases (KDD) that, according to (Fayyad et al., 1996), is defined as the *"non-trivial process of identifying valid novel, potentially useful and ultimately understandable patterns in data"* . This task is brain-intensive. It is usually designed by a human expert. Unluckily the KDD techniques needs a specific skill, and usually doctors are not data mining experts. In this work we propose Medical Knowledge Discover Assistant (MKDA). It is a new tool that we are still developing, to help knowledge discovery process in medical field for non expert users in data mining techniques, as doctors usually are. The system receives the formal specification of the medical experiment research, including goals and the inputs characteristics. The user should say "what" she wants, and not "how" to get it. The system must plan and execute a suitable Knowledge Discovery Process (KDP), designed according to the user's needs and the application domain. Finally, it returns the results. The interaction between the user and the system must be designed carefully. A not expert user must be free from the too technical aspects of the process, and she must be guided through hits and helps.

The rest of the paper is arranged as follows. The next paragraph describes the state of the art for Knowledge Discovery Assistants. The third paragraph introduces a new language to describe the specifications of a medical experimental research. The forth one presents the knowledge base that helps in construction of experiments. Then a new knowledge discovery workflow model is presented in the fifth paragraph. The sixth one describes the functionalities of MKDA system followed by the system architecture in the seventh paragraph. Finally, conclusions and future works are reported.

Cannella V., Russo G. and Pirrone R. (2008).
TOWARDS MKDA: A DATA MINING SEMANTIC WEB SERVICE.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 129-134
DOI: 10.5220/0001521801290134

2 THE STATE OF THE ART

In recent years many researches have been carried out in KDD, with the aim of developing a tool able to perform an autonomous data analysis. The involved field is essentially a combination of some aspects of many research areas such as knowledge based systems, machine learning and statistics. In Mlt-Consultant (Sleeman et al., 1995) the selection of a machine learning method is made with the support of a knowledge-based system. Mlt-Consultant chooses the learning methods on the basis of the syntactic properties of their inputs and outputs according to a set of rules. Another approach is the meta-learning approach. NOEMON (Kalousis and Hilario, 2001) relies on a mapping between dataset characteristics and inducer performance to propose inducers for specific dataset steps. The most appropriate classifier for a dataset is suggested on the basis of the similarity of the dataset with existing ones and on the performance of the classifier for the latter.

The DM Assistant System(Charest et al., 2006) used the case-based reasoning. It has a collection of pre-defined cases. Every time, the system compares a new case with this collection of cases, and establishes the most similar one. This system has been inspired to the CRISP-DM model (cri, 2000).

Another approach is related to the possibility to build the entire process needed to achieve the goal. As an example the IDEA System (Bernstein et al., 2005) starts from characteristics of the data and of the desired mining result. Then it uses an ontology to search for and enumerate the data mining processes that are valid for producing the desired result from the given data. Each search operator corresponds to the inclusion in the DM process of a different data mining technique. In this field some commercial tools as IBM DB Intelligent have also been built. Miner (Han et al., 1996) integrates a relational database system, a Sybase SQL server, with a concept hierarchy module, and a set of knowledge discovery modules. Another commercial tool is Clementine (Engels, 1996) (Wirth et al., 1997). In this system the user-guidance module uses a task/method decomposition to guide the user through a stepwise refinement of a high-level data mining process. Important issues in this field are open source. Yale (Eliassi-Rad et al., 2006) is an environment for machine learning experiments and data mining. It supports the paradigm of rapid prototyping. Yale provides a rich variety of methods which allows rapid prototyping for new applications. Yale can be used mainly by users skilled in KDD. Yale uses Weka (Witten and Frank, 1999), a collection of Java implementations of machine learning algorithms. The preparation of data is supported in Yale by numerous feature selection and construction operators. However, Yale is applied to a single input data table. The Mining Mart software (Euler, 2005) can be used to combine data from several tables, or to prepare large data sets inside a relational database instead of main memory as in Yale. Mining Mart also provides operators that ease the integration with Yale.

3 MEDICAL EXPERIMENTAL RESEARCH SPECIFICATION

The medical researcher has to define the experiment formally, setting its specification. She has to list the collection of features of her research, as data, goals, metrics and models.

We have defined an XML-compliant experiment specification language, a suitable formal language to describe the inputs. Formally the defined language ESL (Experiment Specification Language) is used to represent: the set of data useful to problem definition, the set of user goals, the set of metrics used to evaluate the process results and the representation used for results. Composing such a description could be too hard for a not expert user. To solve this problem, we have designed and implemented a very simple GUI interface. The user can express her needs graphically, and the GUI composes automatically the correspondent description of the specification. The definition of possible goals drives the entire process of data discovery. The goal definition makes possible the fitting of user choices with system capabilities. Some of the most important goals in data mining are represented in the following list.

- *Association Analysis*: it defines the process of finding frequent and relevant patterns in terms of composition rules;
- *Correlation Analysis*: it is used to define the degree of relation in the association analysis. The correlation analysis gives a measure of the correctness degree of the association;
- *Classification*: given a certain number of attributes useful to identify a class, the classification goal is used to find a model describing the situation;
- *Prediction*: the goal is the same of the classification but inputs are continuous;
- *Relevance Analysis*: it is used to define which the relevant patterns are to describe a certain model useful to aggregate data

- *Cluster Analysis*: it is used to classify data not previously classified into clusters;
- *Outlier Analysis*: it follows the cluster analysis and is used to estimate which are the characteristics of not included data in the clusterization process;
- *Evolution Analysis*: defines the time or space data evolution in terms of a model that represents changes.

Due to complexity of processes many possible metrics to evaluate the system have to be used. It's possible to distinguish them in terms of computation load, usefulness of new founded patterns, novelty. Measures are mostly related to particular data mining algorithms or tasks. In fact, they are in direct relation to goals that user wants to obtain. The same considerations are also valid for the set of possible task-dependent representations. The input data are of different types: numerical data, categorical data, complex symbolic descriptions, rules. A deeper differentiation of data is in relation to data composition. Three different input classes have been defined. The first is the object class: data matrix, dissimilarity matrix, single values, graphs are some possible examples. The second is the special input class, as, for instance, numerics, dates, therapy, diagnoses, diseases, patients, counts, IDs, binary data that are used for images or videos, texts and documents. The third class is the variable type class like internal variables, symmetric or asymmetric binary variables, discrete variables, continue variables, scaled variables. The type of input data is a first factor to discriminate the possible choices in the design of the workflow.

4 THE KNOWLEDGE BASE

The Knowledge Base of the system supports the generation of a complete experiment starting from user requests expressed in a formal language. The knowledge base is built in a modular way and is organized in two main levels. In the highest level there is the definition of concepts that are used for the experiments. The concepts have been grouped in relation to the roles they have in the KDP. The knowledge base is a composition of different aspects. Some key terms involved in the process have to be defined. An *experiment* is the composition of a *workflow*, a *model*, a set of *evaluation*s about the model fitting with initial problem and a *representation* of the obtained model to obtain a possible goal. A model is a formal and well definition specification of the result obtained from an experiment over some particular data. A model is obtained through a sequence of steps in a workflow. The workflow steps are essentially grouped in three aggregates: data pre-processing, data mining process and data post-processing. Also the composition rules have been added inside the knowledge bases. Rules define the workflow steps sequence and the choice method to select operators for a particular step.

The design of the Knowledge Base has been inspired mainly to the frames. The operators are defined as frame. In general, also on the basis of the paradigm of OWL-S, each operator has four principal characteristics. It can receive an input, produce an output, be activated under certain constraints, and change the general conditions of the process at the end of its execution. The operators are organized according to a taxonomy. If an operator belong to a certain parental line, it inherits the characteristics of its ancestors, but they have been redefined. Operators in the Knowledge Base are classified according to the step of he workflow in which they are involved, too. In particular, there are: data generation operators, I/O operators, pre-processing operators, mining operators, post-processing operators, validation operators, and visualization operators. This classification helps the expert system during the planning phase, reducing the search space. Inputs and outputs have been classified and organized according a taxonomy too. There are many different input-output operators, according to the type of the data source or to the file format of the data file. Many different types of outputs are possible. Some operators manages data. They can change the content or the structure of the data input. Other operators are involved in the generation of the model resulted from the data mining. There are many possible models. They are classified into: Bayesian ones, neural nets, numerical classifier, numerical regression and prevision models, rules, trees. Each of these classes is divided into sub-classes. The expert system can choose an operator on the basis of the model it is able to produce. To define domain structure an OWL-Dl (owl, 2004) ontology has been built. As previously seen, it is possible to split the ontology in different sub-ontologies. The links between the elements in the same sub-ontology are homogeneous and defines structural properties. The links through different sub-ontologies define the relations between different types of elements.

5 OUR KDD WORKFLOW MODEL

Knowledge discovery in database can be planned as a process consisting of a set of steps. The sequence of

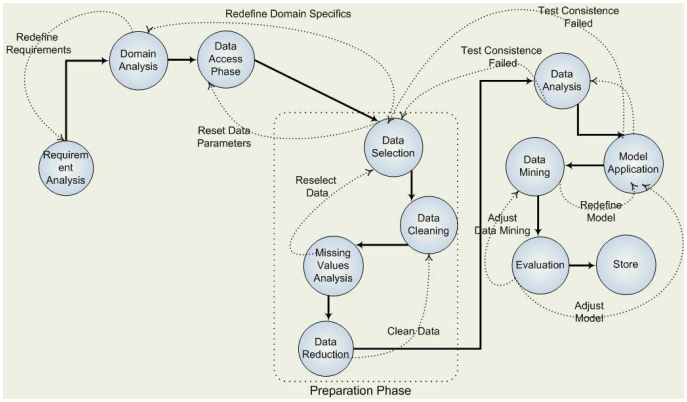

Figure 1: The workflow of a Knowledge Discovery Process.

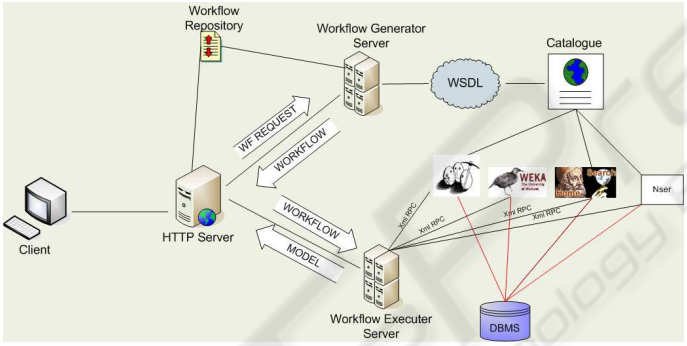

Figure 2: The system architecture and the data flow.

these steps is described with a workflow of the process. We designed a general workflow model as synthesis of different workflows described by literature. Because most of them cover only partially the knowledge discovery process, we tried to redesign a more general workflow (see figure 1).

The workflow is the combination of some possible phases. Phases in the workflow must not be necessarily followed linearly. The process may contain loops. The workflow can be divided into main macro-phases, in their turns divided into sub-phases: problem analysis and specifications definition, choice of the tasks and their execution, results analysis. These three phases are recognizable at a very high view of the process. During the first phase, the analyst interacts with the user to understand her needs and expectations to build consequently effective experiments. The analyst must explicit objectives and knowledge discovery goals clearly and correctly, or the entire process design could be wrong. For this reason the first phase is particularly tricky. To encompass problems related with this phase, as just said, we developed a particular problem definition language. We wanted to make the problem definition simpler especially for all not very skilled users. The user must define evaluation criteria too. On the basis of these criteria final results of the workflow can be evaluated to establish the satisfaction of user's needs. The second phase of the workflow is tightly related to the first one. It deals with the definition of relevant knowledge on the application domain. This knowledge is used by the analyst to extract some domain-driven process choices. At the end this phase is documented, and a possible loop is sometimes necessary. In data access phase, the user defines the specific characteristics of the dataset that must be mined. This phase often needs to consult different sources and bring together data in a common format with consistent definitions for fields and keys. Collected data could contain either too much, less or irrelevant information. These problems are solved during the preparation phase, before the application of the modeling and discovery techniques. Data transformation mainly is performed in two ways: horizontally (changing the dimensionality of the data) and vertically (changing the number of data items). The preparation phase is usually the most time and hardware resources consuming one. In return for this computation load, this phase makes pos-

sible saving resources in the next phases of the workflow and getting better results. The preparation phase can be split into four different sub-phases: data selection, data cleaning, missing values handling and data reduction. None of these phases is mandatory, and the execution of the preparation phase can contain many loops.

During the data mining phase data are analyzed through the chosen techniques. The application of data mining techniques requires parameters calibration to optimal values. Therefore, another data preparation and transformation step is often needed. After the model has been produced, it is converted in an autonomous application able to implement the model. The deployment phase deals with this task. Sometimes, the model can be emulated directly through the development tool, which it has been developed with. Other times, it is necessary to implement a new application in a specific programming language. The evaluation phase establishes how the model is suitable for user's needs according to the success and evaluation criteria specified in the first phase. If the system satisfies user's requests, the entire workflow can be recorded into a repository. In this way, it can be employed again in similar tasks.

6 SYSTEM FUNCTIONALITIES

We are developing a Web intelligent system able to analyze data in an experiment, and to design a knowledge discovery process in those data to extract new knowledge from them. The inputs of the system are the data, the preferences of the user and the domain knowledge regarding the problem treated with the experiment. The system has two outputs: the model describing the new knowledge mined from the data, and the workflow applied to get the model. The latter one can be recorded in a repository and re-used in new similar tasks. Initially the system must be able to analyze data evaluating the presence of problems like noise or missing data. The system must resolve these problems to get data that can be used for the construction of the model. It must reach a trade-off between the accuracy and the time cost. The characteristics of the produced model should be chosen by the user. The choices of the system are not mandatory. The user can change some parts of the workflow to get a different result. In other cases, the system can propose different possible workflows and the user chooses what she prefers. In these cases, the user can choose to try many different workflows to find the best one. The user can interact with the system in two different ways. As just said, we have developed a problem definition language. The user can define the experiment through this language. This functionality has been developed for expert users or repetitive processes. On the other hand, the system has a simple GUI which allows the user to define easily the problem and that compose automatically the description. The interaction process has been inspired to programmable interaction with users like in chatbot systems. Unlike such systems, our interaction is graphical. In particular, we refer to ALICE chatbot (ali,), a system that owns a repository composed of question-answer patterns which are called categories. These categories are structured with the Artificial Intelligence Markup Language an XML-compliant language. The dialogue is based on algorithms for automatic detection of patterns in the statements. Our interaction process functionalities are developed in the same manner: the couples question-answer are categorized and through this mechanism is possible to have a tight interaction. This interaction allows the system to collect information about both tasks and needs of the users.

7 SYSTEM ARCHITECTURE

MKDA system has been designed according to the well-known web service architecture in conjunction with the client-server three-tier one (see figure 2).

The core application is executed on an HTTP server. The client connects to this server remotely. Through an intuitive GUI she composes the characteristics of the experiment. The interface of the client has been developed using the AJAX technologies, which stands for Asynchronous JavaScript and XML. The web pages developed through this technology are more flexible, and can easily and quickly be reconfigured on the basis of the content that has to be shown or on the basis of the user interaction. The interface dynamically composes the description of the request of the user in the language described previously. At this point the request is sent to the server. Afterwards, the request is sent to the workflow generator web service (WGWS), which analyzes this request, and constructs the correspondent knowledge discovery workflow. At this aim, it queries the knowledge base module. At the same time, it consults the Experiment Comparator Service, which, on the basis of a case-based reasoning, lists all past experiments in the repository matching the user's requests. Then it consults the data mining tasks catalogue managed by the tasks catalogue web service (TCWS). TCWS advertises the list of tasks that the system can employ. These tasks are the basic ones that make a workflow.

This list is very long. It includes all operators of libraries and systems as, for instance, Weka, Yale and Ptolemy. These three systems have been embedded into web services, which make possible to use remotely their functionalities. The tasks are advertised in WSDL. The data retrieved in this way are inserted into the description of each step of the final workflow. The workflow is generated automatically by the expert system through a planning process. The system must evaluates the possible actions, and plan a sequence of actions able to produce the desired goal. The choices in the planning phase are related to the characteristics of the actions. The planner links actions together, matching the data flowing in the workflow. The system must bind these services with the steps of the workflow. Actually, the binding is syntactical, based on a shared ontology. The system matches the functionalities of each step to the functionalities of the services in the web, and produces a description of how the workflow should be executed. During the execution of the workflow, these data are used to know where each task has to be executed. After the workflow has been generated, it is sent to the workflow executor web service (WEWS). It manages and coordinates the steps of the workflow. Each step is executed resorting to the Yale, Weka and Ptolemy web services. These services can access the database, and return the result of the execution of the workflow as model. The model is returned by the WEWS to the application on the HTTP server and sent to the client as reply to its initial request. The model together with the workflow can be recorded into the repository. They form an experiment. The collection of experiments can be consulted. In this way the user can eventually re-employ a past workflow when she must work with a similar experiment.

8 CONCLUSION AND FUTURE WORKS

In this work we have proposed a new web based system to help knowledge discovery in medical field for non expert users. We have described system architecture and functionalities. The system is able in a very simple manner to collect the characteristic of the treated experiments. Then a knowledge discovery worflow is generate according to the workflow model we have designed. Finally the system is able to execute the workflow and produces a model as result. The user should concentrate on the specification of the problem, while most of the implementation should be delegated to the system. The system is under development yet. The web infrastructure and the workflow model has been realized and future work is focused on its whole development and test.

REFERENCES

Alice. http://www.alicebot.org/.

(2000). Crisp-dm. http://www.crisp-dm.org/.

(2004). Owl web ontology language use cases and requirements.

Bernstein, A., Provost, F. J., and Hill, S. (2005). Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. *IEEE Trans. Knowl. Data Eng.*, 17(4):503–518.

Charest, M., Delisle, S., Cervantes, O., and Shen, Y. (2006). Invited paper: Intelligent data mining assistance via cbr and ontologies. In *DEXA '06: Proceedings of the 17th International Conference on Database and Expert Systems Applications*, pages 593–597, Washington, DC, USA. IEEE Computer Society.

Eliassi-Rad, T., Ungar, L. H., Craven, M., and Gunopulos, D., editors (2006). *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*. ACM.

Engels, R. (1996). Planning tasks for knowledge discovery in databases; performing task-oriented user-guidance. In *KDD*, pages 170–175.

Euler, T. (2005). Publishing operational models of data mining case studies. In *Proc. of the Workshop on Data Mining Case Studies at the 5th IEEE International Conference on Data Mining (ICDM)*, page 99.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors (1996). *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press.

Han, J., Fu, Y., Wang, W., Chiang, J., Gong, W., Koperski, K., Li, D., Lu, Y., Rajan, A., Stefanovic, N., Xia, B., and Zaïane, O. R. (1996). Dbminer: A system for mining knowledge in large relational databases. In *KDD*, pages 250–255.

Kalousis, A. and Hilario, M. (2001). Model selection via meta-learning: A comparative study. *International Journal on Artificial Intelligence Tools*, 10(4):525–554.

Sleeman, D. H., Rissakis, M., Craw, S., Graner, N., and Sharma, S. (1995). Consultant-2: pre- and post-processing of machine learning applications. *Int. J. Hum.-Comput. Stud.*, 43(1):43–63.

Wirth, R., Shearer, C., Grimmer, U., Reinartz, T. P., Schlösser, J., Breitner, C., Engels, R., and Lindner, G. (1997). Towards process-oriented tool support for knowledge discovery in databases. In *PKDD '97: Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 243–253, London, UK. Springer-Verlag.

Witten, I. H. and Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.