AN ARCHITECTURE FOR COLLABORATIVE DATA MINING

Francisco Correia, Rui Camacho
LIAAD & DEI, Faculdade de Engenharia da Universidade do Porto, Portugal
Francisco.Correia@fe.up.pt, rcamacho@fe.up.pt

João Correia Lopes
INESC-Porto & Faculdade de Engenharia da Universidade do Porto, Portugal
jlopes@fe.up.pt

Keywords: Collaborative Data Mining, Web Services

Abstract: Collaborative Data Mining (CDM) develops techniques to solve complex problems of data analysis requiring sets of experts in different domains that may be geographically separate. An important issue in CDM is the sharing of experience among the different experts. In this paper we report on a framework that enables users with different expertise to perform data analysis activities and profit, in a collaborative fashion, from expertise and results of other researchers. The collaborative process is supported by web services that seek for relevant knowledge available among the collaborative web sites.
We have successfully designed and deployed a prototype for collaborative Data Mining in domains of Molecular Biology and Chemoinformatics.

1 INTRODUCTION

Multi-Relational Data Mining (MRDM) (Dzeroski, 2001) is a very active research field that strives to construct complex models for data. One flavour of MRDM is Inductive Logic Programming (ILP) (Muggleton and De Raedt, 1994). ILP systems can construct complex models, represented in a First Order Logic language, using relevant background knowledge provided by domain experts.

The success of MRDM/ILP applications depends often on the collaboration between domain (e.g Molecular Biology) and ILP experts. The former provides the background knowledge whereas the later knows how to encode the provided background knowledge and how to use the algorithms to construct the models. Also a key point in Data Mining (and ILP) applications in similar domains is that there may profit from sharing information. Sharing information may speed up new DM (ILP) tasks in similar domains.

As in the CRISP-DM methodology (CRISP-DM, 2007), where pre-processing the data takes a significant percentage of the whole DM process, deploying the background knowledge is a significant part of an ILP-based Relational Data Mining process. Sharing components (predicates) of the background knowledge may provide a considerable speedup in the deployment of an ILP application an reduce the dependency on the ILP expert.

There have been several successful CDM experiences as reported in (Lavrac et al., 2004; Blockeel and Moyle, 2002; Moyle et al., 2003).

In this paper we report on a Service Oriented Architecture (SOA), implemented as SOAP Web Services (Papazoglou and Georgakopoulos, 2003), that provides a framework for Collaborative Data Mining. The Web services provide a mechanism that makes completely transparent for users the sharing of information (data sets, background knowledge predicates and papers) among the participant sites.

The rest of the paper has the following structure. In Section 2 we describe the data analysis method we use and explain its advantages and also the advantages of using a collaborative approach. In Section 3 we describe the web application we have developed. The use of Web services in the Service Oriented Architecture is described in Section 4. A case study in domains of Molecular Biology and Chemoinformatics is described in Section 5. We finally present our conclusion and point out future work in Section 6.

2 ILP-BASED MRDM

In order to understand our application and the usefulness of the SOA architecture proposed we explain briefly how an ILP system is used in a MRDM task. An ILP system requires two main ingredients: a set of examples (the data to be analysed) and; a set of predicates called the background knowledge that encode knowledge that the domain expert thinks is relevant for the analysis process. The result of the data analysis process is a model encoded as a set of rules (clauses). Each of these rules (clauses) use the predicates in the background knowledge as their basic blocks for the conditional part of the rule (literals in the body of a clause).

To solve a problem with ILP we require a domain expert or team of domain experts to define the problem and provide the data (examples) and we need an ILP expert to "frame" the domain problem into an ILP problem and run the data analysis experiments. We also need a tight collaboration between domain and ILP experts in the development of the background knowledge. Development of background knowledge is a very time consuming stage of the whole MRDM process. We have speed up the whole process of background knowledge development in three ways: i) reusing existing predicates available at other web sites; avoiding experiments that lead to bad results and; ii) starting off with an existing (in another web site) background knowledge and try to improve it.

3 A WEB-BASED APPLICATION FOR MULTI-RELATIONAL DATA MINING

The application reported in this paper is deployed as a set of collaborative sites supporting the proposed SOA architecture shown in Figure 1 (a).

Each site is devoted to the active collaborative work of a group of researchers, that may be geographically distributed. At each site the different kinds of used/produced items of information are classified as private or public. Public information are accessible to groups in other sites via Web services (passive collaborative work). We now describe in detail each of the sites architecture and the functionalities provided for its users.

At each web site we have adopted the architecture shown in Figure 1 (b). It uses a standard n-tire model: user interface, business logic, data access. The implementation details will be provided in Section 4.

The Data Base underlying each web site stores

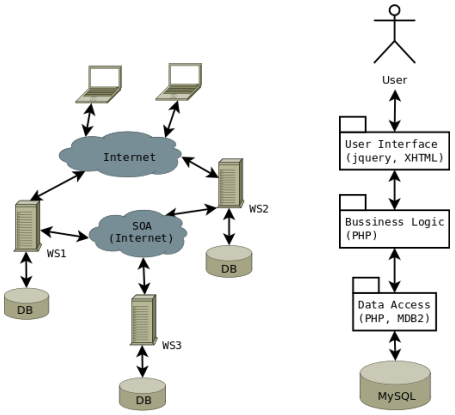

Figure 1: (a) SOA Architecture for CDM.
(b) Implementation Architecture for each Site.

three types of information: data sets upon which data analysis experiments are performed; libraries of predicates for the background knowledge; papers related to the stored data sets; and lots of relevant information concerning the traces of ongoing and past experiments.

The data analysis task is performed by running an ILP system (Aleph (Srinivasan, 2003)) on the working data set. The User Interface (UI) allows any user to run several data analysis tasks. At any time the user may access the status of his tasks using the UI to the *tasks server* that manages a set of machines available in a campus LAN.

Each site includes usage scenarios for different users: the administrator, the ILP expert and the Domain expert.

The implemented administrative tasks, available after authentication by an administrator, include the following: to inspect the details about user actions on web site by checking the log file of actions done; to manage user accounts and access control; to setup the general configuration of the application (e.g. managing the available ILP algorithms; and to manage the list of available web sites (supporting the architecture) that are used transparently to look for information by the Business Logic of the site.

The roll of the ILP expert is mainly to encode the required predicates that are part of the background knowledge. He also is able to manage the hierarchy of categories of such predicate libraries. The ILP expert may perform data analysis tasks with direct control of the ILP system. That is, he may directly control the values of the ILP system parameters.

A domain expert can upload new data sets and papers. He can also manage the meta data: hierarchi-

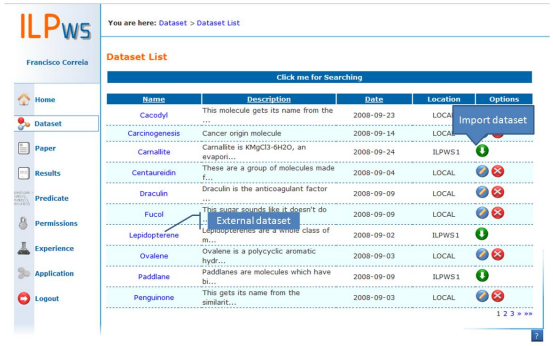

Figure 2: Interface to compose the background knowledge for a data set. Library of predicates may also be searched for externally using web services.

cal categories of papers, of data sets, and of existing predicates. The domain expert may also perform data analysis tasks on personal data.

Access to the Web services (Papazoglou and Georgakopoulos, 2003) is processed at the Web application Business Logic level. According to each user information need request, the business logic layer sends the request to the set of participant sites to get and collect the requested information. Web services are not used to change any information stored in the web sites, they are only used to retrieve information.

4 IMPLEMENTATION DETAILS

In Section 3 the functional and non-functional requirements of the proposed SOA architecture were presented. This section will give implementation details regarding the User Interface (UI), Business Logic (BL) and Database.

User Interface

The interface in Figure 2, shows an example of the provided User Interface. In this example a list of data sets is provided and there are some external data sets (located in the ILPWS1 machine). The options are also different for a local and an external data set. The simplicity and usability are the main points in this application.

The user may choose to access external data sets. If that is the case then the Business Layer will transparently use the registered Web Services to retrieve related information. The user will state his information needs in a special purpose UI and the Business Logic layer will retrieve local information from the database and calls the appropriate Web services to retrieve information from the other participant sites. All that information, coming from different sources, will then be presented to the user in a consistent way.

Business Logic

When the request for information arrives from the XHTML UI in the Browser, the BL layer will query its own database and uses the Web services implemented to retrieve information from other participant sites.

The Web services are used to share/access information (data sets, papers, background knowledge) publicly available at the different web sites that implement the collaborative framework. Communication with the Web service and the answer result is implemented through eXtensible Markup Language (XML) (Moller and Schwartzbach, 2006) and uses the standard Simple Object Access Protocol (SOAP) (Mitra and Lafon, 2007). All possible operations are described in a Web Services Description (WSDL) (Booth and Liu, 2007), where all services provided by the application are listed.

For data sets, papers and background knowledge each site provides services to list all (e.g. data sets), get information about one resource given the ID, download the resource to the local file system and get and import the resource (e.g. data sets) into the site database.

For data sets the implementation provides the calls to: return the list of data sets that exist in the site; return the stored information about a data set; return the content of a data set; and returns all information related with the dataset in a format that can be used to be inserted in the local site database [1].

Because some information may be very large (e.g. the data sets) we give the option to the user to see if its really what he wants before retrieving the files. The requests to a specific information is done only when it's needed.

The advantages of using a Web service instead of direct access to the database of the remote Business Logic layer, are that Web services implementation works over different platforms. This way each site may be using different Operating Systems and different implementation languages may be used.

5 THE *ILP-WS* FRAMEWORK

We have implemented and deployed a prototype of the framework described in the previous sections. The application domains of this case study

[1] For papers and background knowledge is similar.

are Structure-Activity Relationship (SAR) problems in chemoinformatics and problems in genomic and proteomic in Molecular Biology. Our prototype has two web sites where biologists, biochemists and ILP experts may solve the referred kind of problems. The ILP system available for the experiments is Aleph.

Each site running the application allows domain and ILP experts to implement active Collaborative Data Mining tasks. Domain experts provide problems and data (examples) and the ILP experts develop the background knowledge predicates for those problems. Each site has libraries of available predicates organised in a hierarchical fashion and according to a hierarchical structure defined by domain experts. Each stored predicate has an English description of its function and the detailed implementation is hidden from the domain expert.

We have provided an interface (see Figure 2) where the domain expert may assemble the background knowledge by searching and choosing predicates from this hierarchically organised library of predicates. At this stage Web services are used to search other web sites where the application is deployed, looking for predicates of the required category. This procedure may save time in the development of the background knowledge. An ILP expert is required only when the domain expert decides to use some knowledge that is not encoded as a predicate locally neither available using the Web services.

Before starting the data analysis experiments the user may use the UI to inspect existing results of other experiments on the data set, if publicly available. This will give him an idea of what background knowledge have been tried and what were the correspondent results and therefore avoid repeating useless experiments or avoid choosing predicates that seem to be of no use for the analysis of the data.

The expert may undergo a sequence of experiments where models are constructed and shown to the expert. Each step of the experimental process is recorded so the expert may inspect previously constructed models and in the end he may decide which models to store as final results of the analysis process. He may also decide which information to make public.

6 CONCLUSIONS

In this paper we have described a framework for Collaborative Data Mining. At each site the framework enables the solving of domain problems with the help of ILP experts that develop the background knowledge and use ILP systems. Web services look at other sites for publicly available information that are relevant for the solving of problems.

The use of Web services extended the *traditional* approach to Collaborative Data Mining possibilities implementing a *passive Collaborative Data Mining* that searches web sites for relevant information.

ACKNOWLEDGEMENTS

This study was funded by FCT project "ILP-Web-services" (PTDC/EIA/70841/2006).

REFERENCES

Blockeel, H. and Moyle, S. (2002). Collaborative data mining needs centralised model evaluation. In *Proceedings of the ICML-2002 Workshop on Data Mining Lessons Learned*, pages 21–28.

Booth, D. and Liu, C. K. (2007). Web services description language (WSDL) version 2.0 part 0: Primer. Technical Report Second Edition, W3C Recommendation. `http://www.w3.org/TR/wsdl20-primer`.

CRISP-DM (2007). Cross industry standard process for data mining. `http://www.crisp-dm.org/`.

Dzeroski, S. (2001). *Relational Data Mining*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Lavrac, N., Motoda, H., Fawcett, T., Holte, R., Langley, P., and Adriaans, P. (2004). Introduction: Lessons learned from data mining applications and collaborative problem solving. *Machine Learning*, 57(1-2):13–41.

Mitra, N. and Lafon, Y. (2007). SOAP version 1.2 part 0: Primer. Technical Report Second Edition, W3C Recommendation. `http://www.w3.org/TR/soap12-part0/`.

Moller, A. and Schwartzbach, M. I. (2006). *An Introduction to XML and Web Technologies*. Addison Wesley.

Moyle, S., McKenzie, J., and Jorge, A. M. (2003). Collaboration in a data mining virtual organization. In *Data Mining and Decision Support: Integration and Collaboration*, The International Series in Engineering and Computer Science, chapter 5, pages 49–62. Springer.

Muggleton, S. and De Raedt, L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679.

Papazoglou, M. P. and Georgakopoulos, D. (2003). Service-oriented computing. *Communications of the ACM*, 46(10):2528.

Srinivasan, A. (2003). The Aleph Manual. Available from `http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph`.