Adapting the Weka Data Mining Toolkit to a Grid based environment

María S. Pérez , Alberto Sánchez , Pilar Herrero, Víctor Robles, José M. Peña

Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain

Abstract. Data Mining is playing a key role in most enterprises, which have to analyse great amounts of data in order to achieve higher profits. Nevertheless, due to the large datasets involved in this process, the data mining field must face some technological challenges. Grid Computing takes advantage of the low-load periods of all the computers connected to a network, making possible resource and data sharing. Providing Grid services constitute a flexible manner of tackling the data mining needs. This paper shows the adaptation of Weka, a widely used Data Mining tool, to a grid infrastructure.

Keywords: *Data Mining, Data Mining Grid, Grid Services, Weka Tool*

1 Introduction

Data mining is a complex problem, mainly because of the difficulty of its tasks and the huge volume of data involved in it. Roughly, the data mining tasks can be classified as preprocessing, specific data mining tasks (e.g, rule induction) and postprocessing tasks.

Several initiatives have tried to eliminate this complexity. In this sense, one of the most important trend is the development of high performance data mining systems [8].

Nevertheless, traditional and homogeneous distributed systems do not solve the challenging issues related to data mining. Grid computing has emerged as a new technology, whose main challenge is the complete integration of heterogeneous computing systems and data resources with the aim of providing a global computing space [4]. We propose the use of the grid technology as a new framework in which data mining applications can be sucessfully deployed.

On the other hand, Weka [11] is a widely used Data Mining tool, written in Java. This paper describes a generic architecture for making data mining grid-aware services. It also addresses WekaG, an implementation of this architecture, which uses Weka.

This paper is organized as follows. Section 2 shows how the grid technology allows data mining tasks to be performed in a flexible way. Section 3 describes WekaG, an adaptation of Weka to a grid environment. Furthermore, a generic architecture for making data mining libraries grid-aware is shown. This architecture is referred to as DMGA (Data Mining Grid Architecture). Section 4 analyses a sample data mining algorithm, the Apriori algorithm, deployed in a grid environment. Section 5 talks about works related to our proposal. Finally, we conclude with some remarks and the ongoing and future work.

2 Data Mining Grid

Data mining applications demand new alternatives in the field of discovery, data placement, scheduling, resource management, and transactional systems, among others. This is due in part to the following reasons:

- It is required to access to multiple databases and data holders, in general, because no single database is able to hold all data required by an application.
- In a generic scenario, multiple databases do not belong to the same institution and are not situated at the same location, but geographically distributed.
- For increasing the performance of some steps of the data mining process, it is possible to use local copies of the whole dataset or subsets.
- Business databases or datasets may be updated frequently, which implies replication and coherency problems.

Several architectures have been proposed. In [3], Cannataro et al. define the *Knowledge Grid* as an architecture built on top of a computational grid. This architecture extends the basic grid services with services of knowledge discovery on geographically distributed infrastructures.

Another different architecture has been proposed by Giannadakis et al. in [6], named *InfoGrid*. InfoGrid is mainly focused on the data integration. This infrastructure includes a layer of Information Integration Services, which enables heterogeneous information resources to be queried effectively.

In [10], we introduce a generic and vertical architecture (DMGA) based on the main data mining phases: pre-processing, data mining and post-processing. Within this framework, the main phases are deployed by means of grid services. WekaG, explained in the following section, is an implementation of this architecture.

3 WekaG: A Data Mining Tool for Grids

Weka is a collection of machine learning algorithms for data mining tasks developed at the University of Waikato in New Zealand [11]. Weka contains tools for all the data mining phases: data pre-processing, data mining tasks (e.g. classification, regression, clustering, association rules), and data post-processing. One important feature of this toolkit is the flexibility of this tool for developing new machine learning schemes.

WekaG is thought as an extension of Weka for grid environments. WekaG is based on a client/server architecture. The server side is the responsible of the creation of instances of grid services by using a factory pattern. These grid services implement the functionality of the different algorithms and phases of the data mining process.

We have also developed a WekaG client, which is responsible for communicating with the grid service and offering the interface to users. In this way, Weka is not modified. We only add a new input in the Graphical User Interface for providing this new capability.

The main purpose of adapting Weka to a grid environment is to define an infrastructure for data mining that includes at least the following components and features:

– Coupling data sources, which can be dynamically installed and configured. This characteristic makes easier data movement and replication. Data filtering, data replication and use of local datasets help to enhance the efficiency of data mining applications on grid infrastructures.
– Authorized access to data resources, providing a controlled sharing of data within a virtual organization [5]. This implies the use of authentication and access control mechanisms, in the form of access policies, by using GSI (Grid Security Infrastructure).
– Data discovery based on metadata attributes.
– Planning and scheduling resources to the data mining tasks.
– Based on the application, and maybe on other parameters provided by the user, we can identify the available and appropriate resources to use within the grid. This task could be carried out by a broker function. In this case, if we have several equal or different grid services in different locations, we can use a trading protocol for deciding at run time which one can provide the features which fit most to the client requirements.

Although WekaG constitutes a useful tool for data mining, our main purpose is to extend this functionality for several libraries and new algorithms. In this sense, WekaG is a particular implementation of a more general architecture, whose name is DMGA (Data Mining Grid Architecture). DMGA is shown in Figure 1.

As we can see in this figure, we have chosen the use of Globus Toolkit 3 (GT3 in the figure). This release of Globus is stable.

DMGA can be thought as a flexible way of making services grid aware. Our services offer a client and an user interface, hiding all the problematic related to the management of GT3, which is tackled by the appropriate grid service.

4 AprioriG: A WekaG Case Study

We have built a first prototype, which demonstrates the feasibility of our design. Our first prototype only includes a sample data mining service, but we will intend to create several grid services, which can be composed in order to create dynamic workflows, which improve the Weka functionality. This prototype implements the capabilities of the Apriori algorithm [1] in a grid environment. The Apriori service (`buildAssociation`), which produces association rules from a dataset, is specified by means of WSDL (Web Services Description Language).

For the development of this functionality, we have used the object serialization, in order to store and retrieve the state of the required objects. This feature allows Weka to be extensible to support marshaling and unmarshaling and thus, to access to remote objects. Most of the Weka classes are serializable, and specifically the Apriori class:

```
public class Apriori extends Associator implements
                        OptionHandler, CARuleMiner {
     /* Apriori class extends the abstrac class
        Associator */
}
```

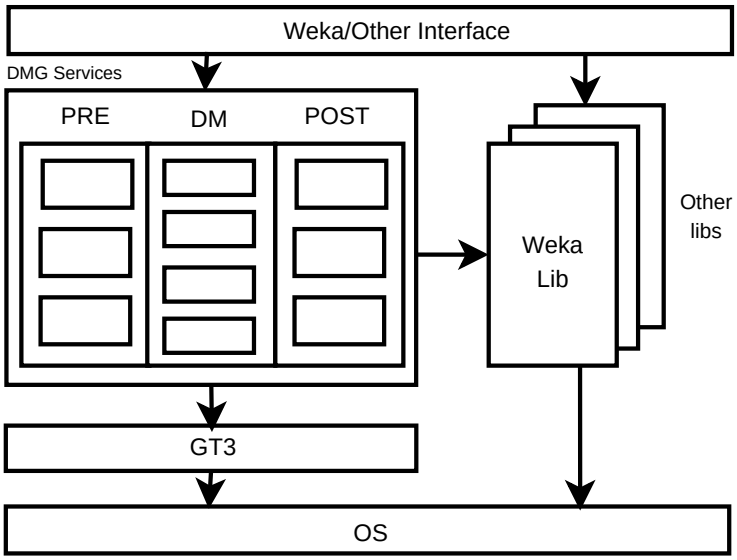

Fig. 1. DMGA Overview

```
public abstract class Associator implements
                      Cloneable, Serializable {
     /* Associator class implements the interface
        Serializable */
}
```

Figure 2 shows an example of execution of the Apriori algorithm over a sample dataset. The algorithm is performed by the AprioriG service, whose results are sent back to the client graphical user interface.

For the deployment of the files to the Grid services nodes, we use GridFTP [2]. The main reason is that GridFTP is integrated within the Grid stack and supports parallel data transfer, which enhances the performance.

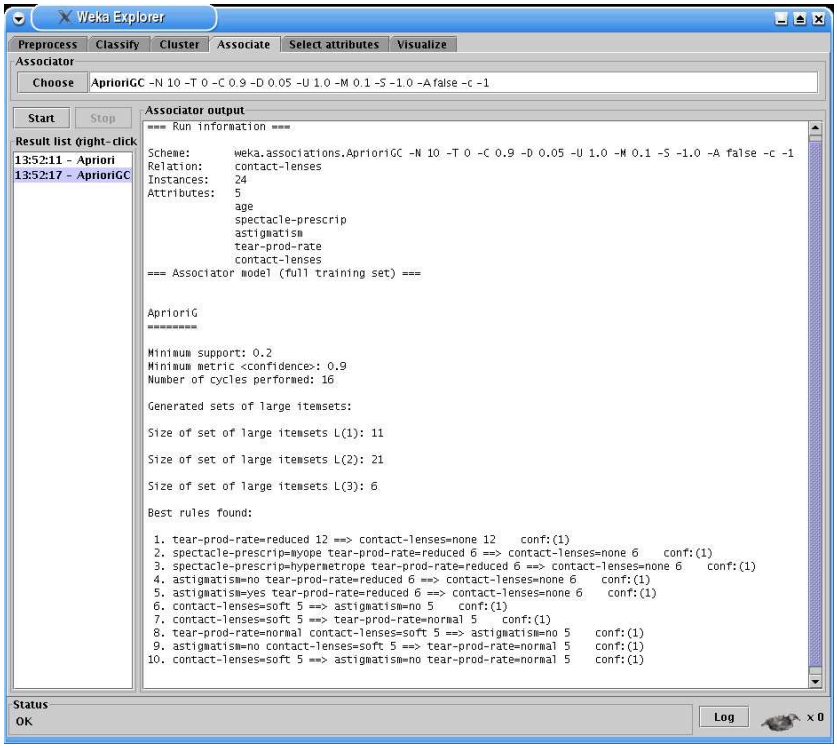

Fig. 2. ApriorioG demonstration

5 Related Work

Grid Weka [7] is being developed in University College Dublin. In this system, the execution of all the tasks are distributed across several computers in an ad-hoc Grid. This proposal does not constitute a real adaptation of Weka to a grid environment, because the framework in which this tool is based is closed. Weka Grid provides load balancing and allow Weka to use idle computing resources. However, it does not provide a "flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources". Moreover, Weka Grid does not use an OGSA-style service, unlike WekaG, whose services are OGSA-compliant. Finally, WekaG is a specific implementation of a more general architecture (DMGA). Our purpose is to extend the functionality of DMGA with different libraries and algorithms, providing a flexible set of services for data mining, which can be used by different virtual organizations, transcending geographical boundaries.

On the other hand, the authors of this paper have developed earlier other tools for performing the Apriori algorithm in a parallel fashion. In [9], an optimization of this algorithm is shown. This work adapts the underlying storage system to this problem through the usage of hints and parallel features. Instead, WekaG and DMGA constitutes

a global optimization to data mining, allowing us to use and reap the advantages of Grid computing in this heavy process.

6 Conclusions and Ongoing Work

This paper describes a generic architecture for making data mining grid-aware services. As an implementation of this architecture, WekaG provides all the functionality of Weka in a grid environment. We have developed a prototype of WekaG, porting the logic of the Apriori algorithm to this new framework.

The advantages of this proposal include the possibility of offering different data mining services in a combined and flexible way, making use of trading and negotiation protocols for selecting the most appropriate service.

As ongoing and future work, we are extending our prototype for building a complete WekaG tool, which includes all the algorithms. Our future work will also include the composition of grid services with the aim of providing suitable data mining services. Additionally, we will evaluate the performance of WekaG in a grid environment composed of different virtual organizations.

References

1. Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *The 1993 ACM SIGMOD International Conference on Management of Data*, 1993.
2. W. Allcock, J. Bester, A. Bresnahan, A. Chervenak, L. Liming, and S. Tuecke. GridFTP: Protocol extensions to FTP for the Grid. *Global Grid Forum Draft*, 2001.
3. Mario Cannataro and Domenico Talia. The knowledge grid. *Commun. ACM*, 46(1):89–93, 2003.
4. I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.
5. Ian Foster. The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150, 2001.
6. N. Giannadakis, A. Rowe, M. Ghanem, and Y. Guo. InfoGrid: providing information integration for knowledge discovery. *Information Sciences. Special Issue: Knowledge Discovery from Distributed Information Sources*, 155(3–4):199– 226, October 2003.
7. Rinat Khoussainov, Xin Zuo, and Nicholas Kushmerick. Grid-enabled Weka: A toolkit for machine learning on the grid. *ERCIM News*, 59, October 2004.
8. William A. Maniatty and Mohammed J. Zaki. A requirements analysis for parallel kdd systems. In Jose Rolim et al., editor, *3rd IPDPS Workshop on High Performance Data Mining*, pages 358–265, May 2000.
9. María S. Pérez, Ramón A. Pons, Félix García, Jesús Carretero, and María L. Córdoba. An optimization of Apriori algorithm through the usage of parallel I/O and hints. *Rough Sets and Current Trends in Computing (LNAI 2475)*, October 2002.
10. Alberto Sánchez, José M. Peña Sánchez, María S. Pérez, Victor Robles, and Pilar Herrero. Improving distributed data mining techniques by means of a grid infrastructure. In Robert Meersman, Zahir Tari, and Angelo Corsaro, editors, *OTM Workshops*, volume 3292 of *Lecture Notes in Computer Science*, pages 111–122. Springer, 2004.
11. H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000.