Association Rules Discovery using Grid Services
Mitica Craus and Cristian Aflori

"Gh. Asachi“ Technical University, Department of Computer Science and Engineering, Iasi, ROMANIA

craus@cs.tuiasi.ro
caflori@cs.tuiasi.ro

Abstract: In this paper we present some aspects about architectures, algorithms and implementations of two emerging fields: Data Mining and Grid technologies. When large data repositories are coupled with geographic distribution of data, users and systems, it is necessary to combine different technologies for implementing high-performance distributed knowledge discovery systems. On the other hand, computational grid is emerging as a very promising infrastructure for high-performance distributed computing. We are considering the implementation of an association rules discovery data-mining task using Grid technologies. For the mining task we are using the Apriori algorithm on the top of the Globus toolkit. The case study is presenting the design and the integration of the data-mining algorithm with the Globus services. We are comparing our version with the related work in the field and we outline the conclusions and the future work.

Key words: Apriori algorithm, Association Rules Discovery, Grid services, Globus toolkit, Open Grid Service Architecture

1 Data-Mining and Grid Technologies

Data and information stored in computers is growing at a very fast rate. Computer based data sources contain a huge amount of information that it makes hard to deal with. Often it is very complex to understand what the important and useful information in data is. To sift large data sources, computer scientists are designing software techniques and tools that can analyze data to find useful patterns. These techniques contribute to define the so-called knowledge discovery in databases (KDD) process.

Data Mining (DM) or Knowledge Discovery in Databases (KDD) (G. Piatetsky-Shaprio, 1991) is an interdisciplinary field with major impact in scientific and commercial environment. Data Mining is the iterative and interactive process of discovering valid, novel, useful, and understandable patterns or models in massive databases. Data Mining means searching for valuable information in large volumes of data, using exploration and analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns and rules. Data Mining is the confluence of several different areas in information technology: machine learning, statistics, databases and data warehousing, high performance computing, visualization.

The major data mining tasks (Fayyad et al, 1996) are prediction and description. Prediction methods use some variables to predict unknown or future values of other variables: these include classification, regression, and deviation detection. Description methods find human-interpretable patterns that describe the data: these include clustering, association rules discovery and sequential pattern discovery. Knowledge Discovery in Databases consists of an iterative sequence of the following steps: data selection, data cleaning, data transformation, pattern generation, validation, and visualization.

Data mining is spreading throughout many organizations because of the substantial contribution it can make. It can be used to control costs as well as contribute to revenue increases.

For example, many organizations are using data mining to help manage customer relationships. Database marketing is another area in which data mining is being used intensively. Data mining offers value across a broad spectrum of industries. Telecommunications and credit card companies are two of the leaders in applying data mining to detect fraudulent use of their services. Insurance companies and stock exchanges are also interested in applying this technology to reduce fraud. Medical applications are another fruitful area; data mining can be used to predict the effectiveness of surgical procedures, medical tests or medications. Companies active in the financial markets use data mining to determine market and industry characteristics as well as to predict individual company and stock performance.
Recently, several KDD systems have been implemented on parallel computing platforms to achieve high performance in the analysis of large data sets that are stored in a single site. However, KDD systems must be able to handle and analyze multi-site data repositories. The combination of large data set size, geographic distribution of data, users and resources, and computationally intensive analysis demand for a parallel and distributed data management and analysis infrastructure for parallel and distributed knowledge discovery (PDKD) (B. Grossman et al, 1999).

Advances in networking technology and computational infrastructure made it possible to construct large-scale high-performance distributed computing environments, or computational grids that provide dependable, consistent, and pervasive access to high-end computational resources. The term computational grid refers to an emerging infrastructure that enables the integrated use of remote high-end computers, databases, scientific instruments, networks, and other resources (I. Foster et al, 1999).

A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities. A Grid is built from multipurpose protocols and interfaces that address such fundamental issues as authentication, authorization, resource discovery, and resource access. It is important that these protocols and interfaces be standard and open. Otherwise, we are dealing with an application specific system. A Grid allows its constituent resources to be used in a coordinated fashion to deliver various qualities of service, relating for example to response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet complex user demands, so that the utility of the combined system is significantly greater than that of the sum of its parts (I. Foster et al, 2001).

Grid applications often involve large amounts of computing and/or data. For these reasons, we think grids can offer an effective support to the implementation and use of parallel and distributed data mining systems.

The following sections present the related work in the field, the process of the distributed association mining using Grid services, a case study, conclusions and future work.

2 Related work

There are several systems proposed in the field of the high-performance data mining. Most of them do not use the computational grid infrastructure for the implementation of the basic services of authentication, data access, communication and security. Those systems operate on clusters of computers or over the Internet. The most known systems for distributed data mining are presented.

Kensington Enterprise data mining is a PDKD system based on a three-tier client/server architecture in which the three tiers include: client, application server and third-tier servers (RDBMS and parallel data mining service) (Köler M, 1999). The Kensington system has been implemented in Java and uses the Enterprise JavaBeans component architecture. JAM is an agent-based distributed data mining system that has been developed to mine data stored in different sites for building so called meta-models as a combination of several models learned at the different sites where data are stored. JAM uses Java applets to move data mining agents to remote sites (Stolfo et al, 1997). BODHI is another agent-based distributed data mining system implemented in Java (Kargupta et al, 1999).

Papyrus is a distributed data mining system developed for clusters and superclusters of workstations as composed four software layers: data management, data mining, predictive modeling, and agent (Grossman et al, 1998). Another interesting distributed data mining suite based on Java is PaDDMAS, a component-based tool set that integrates predeveloped or custom packages (that can be sequential or parallel) using a dataflow approach (Rana et al, 2000). Alongside with these research works on distributed data mining, several research groups are working in the computational grid area developing algorithms, components, and services that can be exploited in the implementation of distributed data mining systems. Thus, this work could be useful integrated with work on parallel and distributed data mining to obtain world-wide grid based PDKD systems for the analysis of large data collections in scientific and commercial areas.

The most known effort for integrating the computational grid and data mining techniques is the Knowledge Grid (Cannataro et al, 2001). The Knowledge Grid can be used to perform data mining on very large data sets available over grids to make scientific discoveries, improve industrial processes and organization models, and uncover business valuable information. Knowledge Grid offers global services based on the cooperation and combination of local services. The system architecture is more specialized for data mining tools that are compatible with lower-level Grid mechanisms and also with the Data Grid services. This approach benefits from "standard" Grid services, which are more and more utilized, and offers an open parallel and distributed knowledge discovery architecture that can be configured on top of Grid middleware in a simple way.

The Knowledge Grid services are organized in two hierarchic levels: the Core K-Grid layer and the High level K-Grid layer. The Core K-Grid Layer offers the basic services for the definition, composition and execution of a distributed knowledge discovery computation over the Grid. The High-level K-Grid layer includes services used to compose, validate, and execute a parallel and distributed knowledge discovery computation.
3. Distributed association mining using Grid services

In this section we are presenting our design of the association rules discovery service in the framework of the Open Grid Service Architecture. The implementation issues and further details are discussed in next section.

Parallel and distributed data mining is the application of the knowledge discovery in database techniques to distributed, large, possibly heterogeneous, volumes of data that are residing over computing nodes distributed on a geographic area. Several parallel algorithms for single data mining tasks such as classification, clustering and rules association have been designed in the past years.

Parallel and distributed knowledge discovery is based on the use of high bandwidth communication networks and high-performance parallel computers for the mining of data in a distributed and parallel fashion. This technology is particularly useful for large organizations, environments and enterprises that manage and analyze data that are geographically distributed in different data repositories or warehouses. The Grid has recently emerged as an integrated infrastructure for high-performance distributed computation. Grid applications often involve large amounts of data and/or computing, and are not easily handled by today's Internet and Web infrastructures. Grid middleware targets technical challenges in such areas as communication, scheduling, security, information, data access, and fault detection.

Grid technologies are evolving toward an Open Grid Services Architecture (OGSA), in which a Grid provides an extensible set of services that virtual organizations can aggregate in various ways (I. Foster et al., 2002). Building on concepts and technologies from both the Grid and Web services communities, OGSA defines uniform exposed service semantics (the Grid service); defines standard mechanisms for creating, naming, and discovering transient Grid service instances; provides location transparency and multiple protocol bindings for service instances; and supports integration with underlying native platform facilities.

OGSA also defines, in terms of Web Services Description Language (WSDL) interfaces and associated conventions, mechanisms required for creating and composing sophisticated distributed systems, including lifetime management, change management, and notification. Service bindings can support reliable invocation, authentication, authorization, and delegation.

The Open Grid Services Architecture supports the creation, maintenance, and application of the service ensembles that Virtual Organizations maintain. OGSA adopts a common representation for computational and storage resources, networks, programs, databases, and the like. All are treated as services—network-enabled entities that provide some capability through the exchange of messages. Adopting this uniform service-oriented model makes all components of the environment virtual—although the model must be grounded on implementations of physical resources. This service-oriented view partitions the interoperability problem into two subproblems: the definition of service interfaces and the identification of protocols that can invoke a particular interface. A service-oriented view addresses the need for standard interface definition mechanisms, local and remote transparency, adaptation to local Operating System services, and uniform service semantics. A service-oriented view also simplifies virtualization through encapsulation of diverse implementations behind a common interface.

The service architecture supports local and remote transparency with respect to service location and invocation. It also provides multiple protocol bindings to facilitate localized optimization of services invocation when the service is hosted locally with the service requestor. In addition, it enables protocol negotiation for network flows across organizational boundaries to allow choosing between several interGrid protocols, each optimized for a different purpose. Finally, the implementation of a particular Grid service interface can map to native, no distributed, platform functions and capabilities.

OGSA defines a Grid service as a Web service that provides a set of well-defined interfaces and that follows specific conventions. The interfaces address discovery, dynamic service creation, lifetime management, notification, and manageability. The conventions address naming and upgradeability. Grid services also address authorization and concurrency control. This core set of consistent interfaces, from which we implement all Grid services, facilitates the construction of hierarchal, higher-order services that can be treated uniformly across layers of abstraction.

Association rule induction is a powerful method, which aims at finding regularities in the trends of the data (Agrawal et al, 1993). With the induction of association rules one tries to find sets of data instances that frequently appear together. Such information is usually expressed in the form of rules. An association rule expresses an association between (sets of) items. However, not every association rule is useful, only those that are expressive and reliable. Therefore, the standard measures to assess association rules are the support and the confidence of a rule, both of which are computed from the support of certain item sets.

Because of the importance of data mining and grid technologies, it is very useful to develop data mining environments on grid platforms by deploying grid services for the extraction of knowledge from large distributed data repositories.

We are proposing a method of integrating the task of the mining association rules in geographically distributed databases with the Open Grid Services Architecture. In the OGSA context, the association rules discovery task is exposed in the form of Grid services. The mining service has several components specific to a Grid service: service data access, service data element, and service implementation. The association rules discovery service is interacting with the rest of the grid services: service registry, service
creation, authorization, notification, manageability, and concurrency.

The architecture of the association discovery service that we are proposing is depicted in Figure 1:

![Diagram of Association Rules Discovery Service in the Open Grid Service Architecture]

The services can be created and destroyed dynamically and they can be destroyed explicitly. They also can be destroyed or become inaccessible through a system failure such as an operating system crash or a network partition. Interfaces are defined for managing a service's life-time and, in particular, for reclaiming the services and state associated with failed operations. Soft-state protocols let the Grid eventually discard the state established at a remote location unless a stream of subsequent keepalive messages refreshes it. Such protocols have the advantages of being both resilient to failure - a single lost message need not cause irretrievable harm - and simple because they require no reliable discard protocol message. A collection of dynamic, distributed services must be able to notify each other asynchronously of significant changes to their state. OGSA defines common abstractions and service interfaces for subscription to and delivery of such notifications, so that services constructed by the composition of simpler services can deal in standard ways with notifications of, for example, errors. A manageability interface defines relevant operations for monitoring and managing potentially large sets of Grid services instances.

Associated with each interface is a potentially dynamic set of service data elements - named and typed XML elements encapsulated in a standard container format. Service data elements provide a standard representation for information about Grid service instances. This important aspect of the OGSA model provides the basis for discovery and management of potentially dynamic Grid service properties. The Grid service specification defines for each interface a set of zero or more service data elements that must be supported by any Grid service instance that supports that interface.

The Grid service specification defines for each interface a set of zero or more service data elements that must be supported by any Grid service instance that supports that interface. Associated with the GridService interface, and thus obligatory for any Grid service instance, are a set of elements containing basic information about a Grid service instance, such as its GSH, GSR, primary key, and home handleMap.

At the implementation level we are defining the specific association rules discovery task details: algorithm libraries and metadata structure. The algorithm libraries implement the sequential and the parallel versions of the association-mining task. The metadata contains the format of the data sources to be mined, the data locations and the structure and location of the knowledge database (stores results of the mining task).

4. Case Study

In the present time, the organizations have different branches located in various geographical places, and each branch own a local database to store information about their own business.

If the top-level management needs to mine novel information in the process of decision-making, there are two options. The first one, not practical, is to transfer data to a single database and mine it on that database. The second option is to implement a virtual organization based on Grid technologies and to integrate mining services for exploring and analyzing.
the data.

The following figure presents a possible infrastructure of a virtual organization implemented using Grid technologies:

![Virtual Organization infrastructure using Grid technologies](image)

**Fig. 2.** Virtual Organization infrastructure using Grid technologies.

The company has a central branch and several local branches (LB). Each branch is composed by a number of Grid nodes (GN) interconnected in a Grid infrastructure. The implementation of the Grid infrastructure is based on the Globus toolkit.

The Globus toolkit is a community-based, open architecture, open source set of services and software libraries that support Grids and Grid applications (I. Foster et al, 2001). The toolkit addresses issues of security, information discovery, resource management, data management, communication, and portability. Globus toolkit mechanisms are in use at hundreds of sites and by dozens of major Grid projects worldwide.

The Globus Project is evolving the Globus toolkit code base to exploit Open Grid Services Architecture capabilities. The result of this evolution is Globus toolkit version 3 (GT3). Figure 3 illustrates GT3’s structure:

![Globus Toolkit architecture](image)

**Fig. 3.** Globus Toolkit architecture.

The Globus toolkit has the following layers:
- The GT3 core, which implements the Grid service interfaces and behaviors;
- GT3 base services, which exploit the GT3 core to implement both existing Globus Toolkit capabilities (resource management, data transfer, information services).
- Higher-level services that may target both GT3 core and GT3 base services such as data management, workload management, and diagnostics.

On the top of these layers we are implementing the association rules discovery services for our case study.

The formal statement of the problem of mining association rules is presented:

\[ I = \{ i_1, i_2, ..., i_m \} \] - set of items;
\[ D = \text{set of transactions, each transaction } T \text{ included in } I; \]
\[ X \text{ in } I, Y \text{ in } I, X \cup Y = 0. \]

Confidence of rule \( X \rightarrow Y \) is \( c \), where \( c \% \) of transactions in \( D \) that contain \( X \) also contains \( Y \). Support of rule \( X \rightarrow Y \) is \( s \), where \( s \% \) of transactions in \( D \) contains \( X \cup Y \). Given a set of transactions \( D \), generate all association rules (or a specified number) that have support and confidence greater than the user-specified minimum support and minimum confidence.

The task of data mining for association rules can be broken into 2 steps: find all large itemsets that have transaction support > minimum support (Apriori algorithm); for all large itemsets, for each itemset \( L \), find all non-empty subsets of \( L \); for every such subset \( A \), the rule is: \( A \rightarrow L - a \) if \( \text{support}(L) / \text{support}(A) > \text{minimum confidence} \) (Agrawal et al, 1994).

For a general problem of association discovery rule given \( m \) items, there are potentially \( 2^m \) frequent itemsets. Discovering frequent itemsets requires a lot of computation power, memory and I/O. In the Apriory algorithm case, for each iteration the database is scanned to obtain the support for new candidates. If the database doesn’t fit in memory then there is a high I/O overhead for scanning in each iteration.

The most popular algorithm used for association rules discovery is Apriori, algorithm presented in the following figure:

**Initial conditions:**
\[ L_k = \{ \text{set of large } k\text{-itemsets (have min support)} \}; \]
\[ C_k = \{ \text{set of candidate } k\text{-itemsets} \}; \]
\[ D = \{ \text{set of transactions, } t \subset D \} \]

**Algorithm:**
\[ L_1 = \{ \text{frequent 1-itemsets} \}; \]
for \( (k=2; L_{k-1} \neq \emptyset; k++) \) {
\[ C_k = \{ \text{Set of New Candidates} \}; \]
for all transactions \( t \subset D \) {
for all \( k\)-subsets \( s \) of \( t \) {
if \( (s \subset C_k) \cdot s.\text{count}++; \)
\[ L_k = \{ c \subset C_k | c.\text{count} \geq \text{minsupp} \} \}
Set of all frequent itemsets = \( \bigcup_k L_k \);}

![Apriori algorithm for association rules discovery](image)

**Fig. 4.** Apriori algorithm for association rules discovery.
In our case, from the central branch we want to launch the association rules discovery task, mining the data from central and local branches. The hosting environment from the central branch encapsulates computing and storage resources for creating association mining services and storage services for the knowledge collected. The hosting environment from the central branch also implements Virtual Organization registry service for providing information about the location of all mining, data transformation and database services. Each service provider has a local registry services (LR) that provides information about the interface of the implemented services.

![Virtual Organization diagram]

**Fig. 5.** Association discovery rules services implementation on top of the Globus toolkit

The user application queries the Virtual Organization registry for searching the association mining services. Then, the user invokes “create Grid service” requests on the two factories in the different hosting environment: the association discovery service provider and storage service provider. These requests create the Apriori service that will perform the data mining operation on its behalf, and an allocation of temporary storage for use by that computation. Each request involves mutual authentication of the user and the relevant factory (using an authentication mechanism described in the factory’s service description) followed by authorization of the request. Each request is successful and results in the creation of a Grid service instance with some initial lifetime. The new data mining service instance is also provided with delegated proxy credentials that allow it to perform further remote operations on behalf of the user.

The newly created association mining service uses its proxy credentials to start requesting data from the database services, placing intermediate results in local storage. The data mining service also uses notification mechanisms to provide the user application with periodic updates on its status. Meanwhile, the user application generates periodic “keepalive” requests to the two Grid service instances that it has created.

The user application can fail for some reason. The data mining computation continues for now, but as no other party has an interest in its results, no further keepalive messages are generated. Due to the application failure, keepalive messages cease, and so the two Grid service instances eventually time out and are terminated, freeing the storage and computing resources that they were consuming.

The user application receives notifications when an association mining service complete its job and the results can be explored from the knowledge base.

**Conclusion**

The association discovery service that we are proposing is a new one. Our approach for distributed association mining is based on the Open Grid Services and the implementation is realized using the Grid toolkit and the Apriori algorithm.

The future work is focused on the evaluation and the optimization of the association rules discovery system implementation. Also, an important issue is the implementation of the parallel version of the Apriori algorithm using the MPICH 2 standard (Message Passing Interface using Grid services).

**Acknowledgment**

This work is supported partially by a grant from a Romanian programme named Information Society.

**References**


