13

A Software Architecture for Data Mining Environment

Georges Edouard KOUAMOU
National Advanced School of Engineering, Cameroon

1. Introduction

Data Mining also called Knowledge Discovery consists in analyzing a large set of raw data in order to extract hidden predictive information. It is a discipline which is at the confluence of artificial intelligence, data bases, statistics, and machine learning. The questions related to the knowledge discovery present several facets whose principal ones are: classification, clustering and association. Several models of algorithms are used for each aspect: Neural Networks, Lattice, Statistics, Decision trees, Genetic Algorithms (Mephu, 2001).
Technically, data mining is the process of analyzing data from many different dimensions or angles, and summarizing the relationships identified. For example, analysis of retail point of sale transaction data can give information on which products are selling and when. The summary of this information on retail can be analyzed in the perspective of promotional efforts to provide knowledge of consumer buying behavior. Based on the acquired knowledge, a manufacturer or retailer could determine which items are most susceptible to promotional efforts. Although several tools were developed for this purpose, we note the lack of software environments which are able to support the user activities in a coherent way during the process of data mining.
A software environment can be seen as a set of components which are able to work in cooperation. Each component offers services to other components and it can require some services from them. The role of the environment consists in coordinating the execution of the components which it incorporates, in order to complete the tasks which are assigned to him. The required goal is to offer a support to the software process i.e. the scheduling of the various stages which describe the goal to reach. In software Engineering in general, this goal is related to the capacity of production and evolution of software. In this case we talk about Software Environment Engineering (SEE) also known as CASE tools. When an applicative software environment is concerned, it is specific to a given theme like data mining, thus it consists of a set of tools designed to analyze and produce information likely to improve the choice of the decision makers.
In general, software environments must evolve to take into account changes and new requirements. For instance, the addition of new tools which implies new functionalities, the removal or the replacement of an existing one in the system becomes essential. These modifications should not involve a consequent restructuring of the whole system. For that, it is interesting to reason on the structure of the environment in order to determine its possibilities of modularity and adaptation to its context. This capacity of evolution of the

software environments is different to the monolithic vision which consists in statically binding all the elements together during compilation. Several mechanisms are available to implement this approach of construction of the extensible environments with the support of software reuse in particular Design Patterns, software architectures and component based development.
In this chapter, we are interested in the environments of data mining. The purpose of this study is to describe an open software architecture, which is able to be instantiated by the adding of components which implement the various algorithms, in the view to develop environments for knowledge discovery. This architecture must face many constraints:

- to take into account the various models of algorithms,
- to provide the mechanisms of interoperability among the models,
- to be integrated into the existing Information System since it makes up the main source of data.

Being given the interest that express the business community and the researchers for the acquisition and/or the development of these types of environment, it is necessary to find the means of reducing the efforts necessary to this effect in particular in a context where powerful human resources are missing. Through the content of this chapter, we will explore the design of such a generic architecture by detailing the possible solutions to resolve the associated constraints.

2. The problem

The activities in the domain of data mining were formerly carried out by the researchers. With the maturity of the algorithms and the relevance of the problem which are addressed, the industrial are interested from now on to this discipline. In this section, we explore what is done by other researchers and the industrialists.

2.1 Context

In the research domain, the interest relates to the development of algorithms and the improvement of their complexity. The tools which result from the implementation of these algorithms are used separately on sets of targeted and formatted data quite simply because, the researchers remain in the situations of test on academic cases.
There are efforts to gather these various tools in a common environment. In this regard, the experience of the University of WAIKATO gave place to a homogeneous environment in term of programming language called WEKA (Witten and Frank, 2005). However the algorithms must be written in java programming language to be integrated into this platform. Also the mechanisms of interoperability or exchange the models among tools are not provided.
Concerning the industrialists, they are trying to implement the data mining algorithms on top of DBMS (Hauke et al., 2003), or to integrate them with ERP (Enterprise Resource Planning) in order to increase the capabilities of their CRM (Customer Resources Management) (Abdullah et al., 2009). Indeed, the algorithms are mature and the tools which result from them are efficient and consistent so that they outperform the old statistical methods. In addition the ability to store large databases is critical to data mining.
The major problem with existing Data mining systems is that they are based on non-extensible frameworks. In fact tools are independent even in an integrated data mining

framework; no models sharing, nor of interaction exchange between tools. Consequently the mining environments are non-uniform because the user interface is totally different across implementations of different data mining tools and techniques. Also a model obtained from one tool is not accessible to another tool.
Thus the needs of an overall framework that can support the entire data mining process is essential, in other words the framework must accommodate and integrate all data mining phases. Also the corresponding environment should provide a consistent, uniform and flexible interaction mechanism that supports the user by placing the user at the center of the entire data mining process. The key issue of this observation will be the construction of an open architecture, allowing extensions and integration of different algorithms implemented by third parties in possibly any language.
Conscious of this disappointment, some studies are undertaken on the subject, but while approaching the question more under the angle of presentation (Khimani, 2005; Poulet, 2001). The aim of their studies consists of developing an effective user-centered graphical environment dedicated to data mining; improving comprehensibility of both the data and the models resulting of data mining algorithms, improving interactivity and the use of various algorithms.

2.2 Contribution

It is a necessity to consider such environments in its globality. The plan consists in reconciling data mining discipline and software engineering with a view to improve the structure of these environments. It is a question of studying the structure of a modular and extensible environment whose design would profit from the current best practices from software engineering which let to implement the advanced techniques to manage the heterogeneity, the distribution and the interoperability of the various components, and especially the reuse and structuring approaches.
As being mentioned early, some reflections are carried out to provide a uniform view on data mining processes. In this way this study is trying to bring together the various approaches to integration (Wasserman, 1990; Ian and Nejmah, 1992) in data mining process. One approach deals with the presentation dimension to integration. The aim is to help users navigate through enormous search spaces, help them gain a better insight into multi-dimensional data, understand intermediate results, and interpret the discovered patterns (Han and Cercone, 2000; Khimani et al., 2004). Another approach is closed to the data dimension to integration. The goal of this approach is to avoid manual data manipulation, and manual procedures to exchange data and knowledge models between different data mining systems (Kurgan and Musilek, 2006).

3. Process of data mining

There is a confusion of terms, between Data Mining and Knowledge Discovery, which is recurrent despite Data Mining concerns application, under human control, which in turn are defined as algorithms designed to analyze data, or to extract patterns in specific categories from data; while Knowledge Discovery is a process that seeks new knowledge about an application domain (Klosgen and Zytkow, 1996). This process consists of many steps among with one of them being data mining.
The process is defined as a set of steps to follow to achieve a goal. In software development discipline, the process consists of a set of activities necessary to transform needs and

requirements expressed by the customer into a software product. The description of the process permits to identify the suitable tools and their sequence, since each tool will be associated to an activity to carry out a task. Software engineering adopted Waterfall model (Roy, 1970), Spiral model (Boehm, 1988) and Iterative models (Kroll and Kruchten, 2003; Highsmith, 2002) that became well-known standards in this area.

Concerning knowledge discovery several different process models have been developed both by research and industrial (Kurgan and Musilek, 2006). We notice that all process models are closed to iterative process model in software development because they often include loops and iterations. The major difference between them is the granularity of the processing steps. From these observations, we think that a data mining process model could emphasize main steps, each gathers the steps of different models as sub-steps or elementary task. Thus the data mining environment support a process including: Data Preparation or Pre-processing, Data Mining itself, Evaluation and knowledge Deployment.

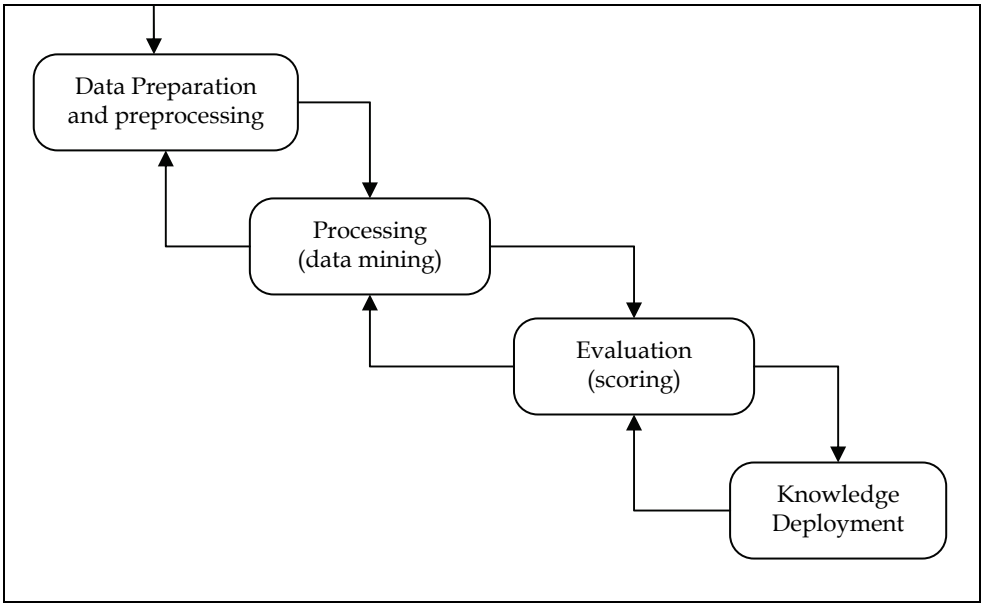


Fig. 1. Description of activities of the data mining process

Preparation and Preprocessing phase: from different sources of heterogeneous data (DB, structured text file, spreadsheet,…) to build a warehouse by combining the various data and removing inconsistency between them. This phase covers all the tasks involved in creating the case table from the data gathered in the warehouse. Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. Tasks include table and attribute selection as well as data cleaning and transformation. For example, you might transform a nominal attribute into binary values or number; you might insert new columns or replace values for example the average in cases where the column is null.

Data mining phase is the phase which consists in processing data to build models. Different methods are applied to extract patterns. The tools used result from the algorithms of

classification, clustering and association rule. The persistent patterns are saved as models in the data warehouse for a later use.

Evaluation also known as scoring is the process of applying a model to new data. Data mining is accomplished by building models. A model uses an algorithm to act on a set of data. The notion of automatic discovery refers to the execution of data mining models. Data mining models can be used to mine the data on which they are built, but most types of models are generalizable to new data.

Knowledge Deployment is the use of data mining within a target environment. In the deployment phase, insight and actionable information can be derived from data. Deployment can involve scoring (the application of models to new data by using either the same tool that produces the given model or another tool), the extraction of model details (for example the rules of a decision tree), or the integration of data mining models within applications, data warehouse infrastructure, or query and reporting tools.

From this description follows the architecture of the environment structured in a layered style including a data warehouse, tools slots and presentation components.

4. Architectural structure

Software architecture is a concept which refers more to the design than with the implementation. It embodies the concepts necessary to describe the system structure and the principles guiding its evolution. Our reflection is based on the following definition: the software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationship among them (Clemens et al., 2003).

From this point of view, we can gather the components according to three layers:

- the components of user interface which ensure the dialog with the user. The actions of the user cover the choice of the tool to be used, the method for the evaluation and the capture of the options (modification of its parameters),
- the business components which ensure the (pre) processing. They consist of four groups of tasks.
- The storage system and the data connectors which get into memory the data contained in the warehouse and arrange in the warehouse the persistent entities to be exploited later.

4.1 Data warehouse

The data warehouse makes it possible to store, organize and manage the persistent data and extracted knowledge. Data can be mined whether it is stored in various format either flat files, spreadsheets, database tables, or some other storage format. The important criterion for the data is not the storage format, but its applicability to the problem to be solved. Although many data mining tools currently operate outside of the warehouse, they are requiring extra steps for extracting, importing, and analyzing the data. Thus connectors must be provided to ensure the interactions between the tools and the warehouse. To guarantee its independence with respect to the processing tools and the management system of the warehouse, it is necessary to provide with the generic interfaces. Furthermore, when new insights require operational implementation, integration with the warehouse simplifies the application of results from data mining.

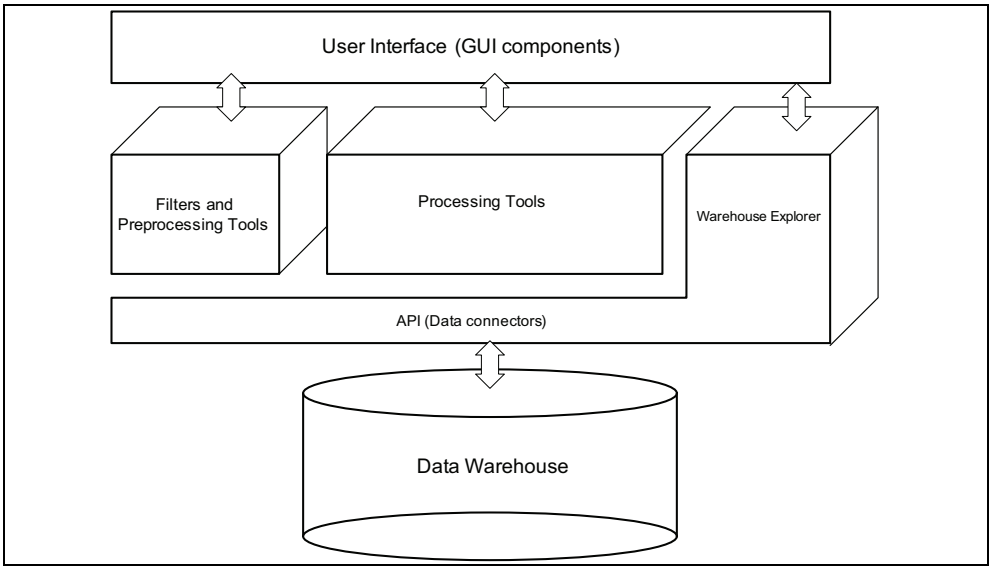


Fig. 2. A layered Architecture for Data Mining environments

4.2 Tools slots

The slots are designed to accommodate the new tools to plug in the environment. These tools are used for raw data preprocessing or data analysis. There are numerous algorithms for business tier, from attributes selection which is a part of data preprocessing to the analysis phase (Han and Kamber, 2006). The commonly used techniques are:

- **Artificial neural networks**: Non-linear predictive models that learn through training and resemble biological neural networks in structure.
- **Decision trees**: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID).
- **Genetic algorithms**: Optimization techniques that use processes such as genetic combination, mutation, and natural selection in a design based on the concepts of evolution.
- **Nearest neighbour method**: A technique that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset (where $k \geq 1$). Sometimes called the k-nearest neighbour technique.
- **Rule induction**: The extraction of useful if-then rules from data based on statistical significance.
- **Lattices**: also known as Galois Lattices are mathematical structure allowing to represent the classes, described from a set of attributes, which are underlying to a set of objects.

The business components of this tier are related to data preparation and data mining itself. The phase of preparation consists of cleaning data in order to ensure that all values in a dataset are consistent and correctly recorded. The data mining analysis tools can be categorized into various methods (Klemettinen et al., 1999):

1. Classification: The process of dividing a dataset into mutually exclusive groups such that the members of each group are as "close" as possible to one another, and different groups are as "far" as possible from one another, where distance is measured with respect to specific variable(s) you are trying to predict.
2. Clustering: The process of dividing a dataset into mutually exclusive groups such that the members of each group are as "close" as possible to one another, and different groups are as "far" as possible from one another, where distance is measured with respect to all available variables.
3. Association rule: Data can be mined to identify associations. In this case data mining model is a set of rules describing which attribute implies another on what confidence and support.
   - **Support:** Support of a rule is a measure of how frequently the items involved in it occur together. Using probability notation, support (A=>B) = P(A, B).
   - **Confidence:** Confidence of a rule is the conditional probability of B given A; confidence (A => B) = P (B | A), which is equal to P(A, B) or P(A).

This categorization helps to define the interfaces independently to the nature of the algorithm and its implementation. That is while multiple implantations could be plugged with respect to the same interface.

4.3 User interface

The user interface ensures the presentation aspect. It permits to explore datawarehouse, to launch a tool and to visualize knowledge in various formats. Insofar as the expert must play a central role during the process, a particular accent must be put on interactive dimension with the user. Visualization must be graphic with possibilities of edition, this to increase interactive dimension with the user. Finally all new tool plugged in the environment will have to adapt to the user interface provided and to reach the warehouse to recover data or to store its results there.

4.4 Documenting the architecture of data mining environment

The architecture focuses on the low layers (business and data), with the hope that the user interface will profit from the results of the existing studies (Khimani and Al, 2004). Since extensibility and reuse are the main features of the environment, the description of architecture is based on the Design Pattern (Gamma and Al, 1996) and UML (Booch et al.,) will be used as language of expression.

The reasoning is based on the families of tools given that they define a common interface under which can be grafted several different implementations. The crucial question consists in finding the appropriate structure of these various interfaces (common attributes and operations).

4.4.2 Classification and clustering

Classification and clustering algorithms proceed in two phases:

- the training which consists in building the classifier who describes a predetermined sets of classes of data; during this phase, the classifier creates the internal model which carries knowledge necessary to classify any object.
- the classification which consists in using the classifier built to assign a class to a new object. At the end of this phase, an evaluation of the model is carried out to produce a confusion matrix.

The confusion matrix shows the dispersion of the examples. It is a matrix of which the columns are equal to the number of initial classes and the rows are equals to the classes determined by the classifier/cluster algorithm. The value of the cell (i,j) represents the examples of the initial class i which has been put in the class j by the classifier. To ensure this, one of the following evaluation methods is considered: holdout, leave-one-out, cross-validation, resubstitution:

- **Holdout**: data are partitioned in two subsets. One set is use for learning and the other set is used for testing the pertinence of hypothesis.
- **Leave-one-out**: each example is used once to test the hypothesis obtained from the others.
- **Cross validation:** the data are partitioned in several subsets of identical size; each one of them is used once to test the hypothesis obtained from the remainder of subsets.
- **Resubstitution:** the set of examples which are used for learning are also used for testing.

The design patterns "Classifieur" and "Cluster" respectively define the skeleton of the algorithms of classification and clustering. However each concrete classifier/Cluster has the responsibility to redefine certain aspects which are specific to the model that it uses. For this layer, we define a succession of abstract entities which make it possible to adapt the concrete components by implementing the appropriate interface.

4.4.2 Association rule

The algorithms of association rule present two facets: the research of the rules of association and the research of the sequences. In the case of the rules of associations, the model consists of all the rules found by the algorithm. The relevance of each rule is evaluated through the metric one used by the algorithm. In general it is confidence. Each rule consists of two whole of items characterized by their support. The first being the antecedent or premise and the second is the conclusion or consequence. One Item indicates an article or a pair attribute/value of the data source. In the case of sequential reasons, the model is rather made up of a set of sequences. A sequence being a continuation of Itemsets ordered in time.
The design pattern « Association » is used to define the interface of the algorithms related to this family of data mining algorithms. As in the case of classification, this interface let the implementation to the concrete association algorithm.

4.4.3 Data manipulation and representation

All the algorithms input a context. Let O be a set of objects, and A a set of attributes. A *Context* is a relation I=O×A such as:

- $I(o_i,a_j) = v$ which represents the value of attribute a_j for the object o_i
- $I(o_i,a_j) =$ Null if the value a_j is unknown for the object o_i

Then an example represents a line of this relation. The Figura 3 is an illustration of a relation I using O={1,2,3,4,5} and A={a,b,c}

	a	b	c
1	v_{a1}	v_{b1}	v_{c1}
2	v_{a2}	v_{b2}	v_{c2}
3	v_{a3}	v_{b3}	v_{c3}
4	v_{a4}	v_{b4}	v_{c4}
5	v_{a5}	v_{b5}	v_{c5}

Fig. 3. An illustration of context

The Context is the representation in memory of the data stored on the disc (in the warehouse). The structure of table is adapted for its representation. In object perspective orientation, a context has composite entity formed by examples each being a line of the tabular structure.

4.4.4 Logical structure of the data mining environment

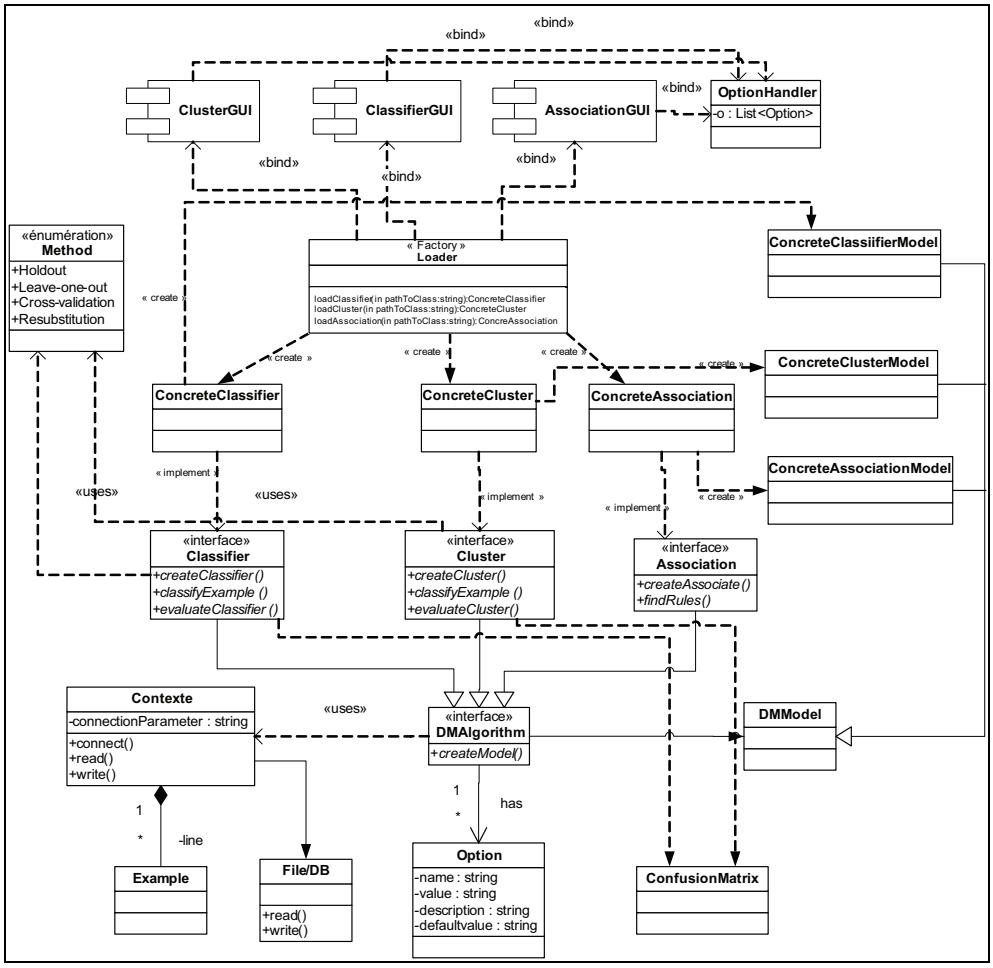


Fig. 4. A logical view of the environment

The environment shall be configured with different families of algorithms. For each family, just their interface is exposed. The loader is responsible for the instantiation of each algorithm in the sense that it implements the operations to create concrete algorithms. The information necessary to this task will be stored in a repository which could be a configuration file in this case. The data mining algorithms manipulate model (or knowledge). The models are created from contexts which can be seeing as proxy since it

provides an image in the memory for data in the warehouse. Classifier (resp. Cluster) algorithms encapsulate the knowledge of which confusion matrices are created. The concrete algorithms redefine the operations to return the appropriate model and the appropriate confusion matrix.
Because of the complexity of GUI components, there are designed as black box subsystems intending that the suitable GUI framework which is more adapted to the needs of the environment will be plug in this structure.

5. Discussion and future trends

As this study is going on, there are several issues we may decide to consider while bearing in mind the state of the technology. The technological capabilities, principally the use of standards, are an important factor that influences the openness and the acceptance of such environments. These issues include:

- Integration : adding new tools in the environment;
- Interoperability over multiple dimensions: vertical (interaction between adjacent layers), horizontal (interaction between components of the same layer) and the interaction with other framework.
- Compatibility between legacy tools while plugging in the environment and its accommodation with the interoperability mechanism which is implemented.

5.1 Integration and interoperability issues

Since this environment is developed to provide a framework for integrating the entire data mining process, the main issue to deal with consists in supporting interoperability between the data mining tools in the same environment or interoperability between the data mining environment and other external software (Cios and Kurgan, 2005). Interoperability refers to the ability of a computer system and/or data to work with other systems or data using common standards or processes (Wileden et al., 1991). From the data mining perspective, the management of interoperability can help to achieve multiple goals:

- to standardize communication between diverse data mining tools and storage systems;
- to provide standard mechanisms for sharing data between data mining tools that work on different software platforms.

Interoperability within the scope of the data mining supposes simply the exchange of the data and knowledge between different software tools. The stake of such an operation is to compare the relevance of results provided by two different algorithms. Conscious of this situation of the reuse of the models from one environment to another, the DMG (Data Mining Group) proposed the Predictive Modelling Markup Language (PMML) (PMML, 2001). The PMML is a specification of the XML (DTD, XML Schema) to represent efficiently the characteristics that allow reusing a model independently of the platform which has built this model. It is currently used by the most important industrials in the domain of data mining environments (IBM, Microsoft, Oracle, SAS, SPSS, etc). However the existing research platforms do not integrate easily this standard because the common work undertaken within this framework relies on the improvement of the complexity and the performance of the algorithms.
The PMML provides specifications (DTD or XML schema) to represent the characteristics of a model independently of the platforms. To reuse the models, the platform consuming the model needs the configuration of the data to use. In general they are the types of the

attributes and their role in the model. Accordingly, the management of interoperability consists in studying the possibility of extracting the generic characteristics which are common to models of the same nature. This abstraction permits to maintain the algorithms without rewriting them. By using PMML, different applications taking from different places can be used across the process in the same environment; one application can generate data models, another application analyzes them, another evaluates them, and finally yet another application is used to visualize the model. In this sense, any subsystem of the environment could be satisfied simply by choosing the suitable software components which have already incorporated the PMML language. For instance on the level of the presentation layer, VizWiz (Wettschereck et al., 2003) can be used for visualization of models, PEAR (Jorge et al., 2002) for exploration and visualization of association rules.
The use of XML language and the related standard to achieve the interoperability in such environment has many advantages unless the compatibility with the legacy systems is assured. XML permits the description and storage of structured or semi-structured data, and to exchange data in a tool-independent way.

5.2 Compatibility issue

Compatibility is a consequence of the integration perspective of the environment. If the new tools can be developed according to the principles of the environment, the integration of the legacy tools is a challenge. We notice that several environments were developed in the commercial world and the world of research for the retrieval of knowledge starting from the data. These environments are heterogeneous on several aspects such as the handled formats of data or the supported platforms, the types of algorithms and built models, the access mode to the data, etc (Goebel & Gruenwald, 1999). In spite of these differences, the objective of very model of dated mining is to store relevant information obtained by data analysis to facilitate decision making by carrying out predictions of the events for example. The problem of incompatibility relates to two aspects: data (to solve the heterogeneity of the data) and interfaces tools (incompatibility of the interfaces).
Since XML is the standard in fact to support interoperability, the solution to face the heterogeneity of the legacy tools, candidate with integration in the environment, consists in conforming them to this standard without however modifying their internal structure. The implementation of the wrappers (Kouamou & Tchuente, 2008) is a technique often used in such circumstances. The wrapper will be charged to transform the data to represent them in a format conforms to that awaited by the legacy tool.

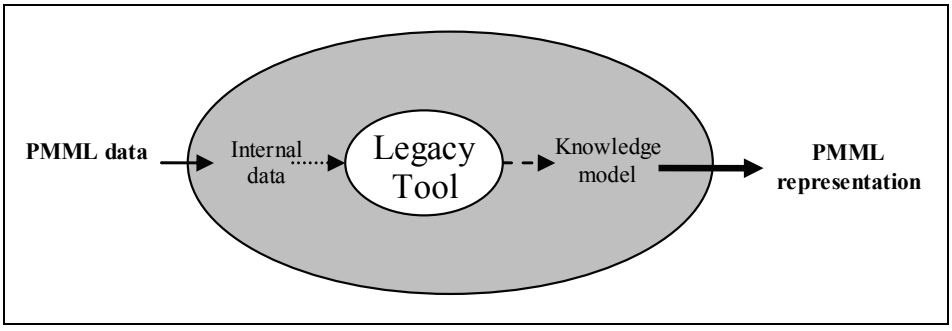


Fig. 5. Overview of the wrapper

The other function of the wrapper is to convert the interface of a legacy tool into one of the main interfaces the loader class expects. Thus the wrapper lets classes work together that couldn't otherwise because of incompatible interfaces.

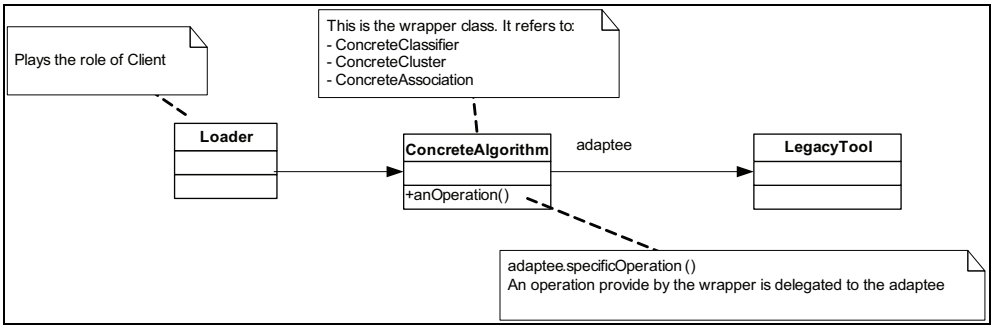


Fig. 6. The structure of the wrapper

The wrapper can be structured in two different ways (Gamma et al., 1995): (i) by interface inheritance or (ii) by delegation. Considering the fact that legacy tool may be heterogeneous in term of programming languages, the second approach is more appropriate in this circumstance.

5.3 Implementation issue

Since the implementation of a data mining environment as described here requires enormous human resources for its construction, to gain in productivity and time, it is important to gradually try out the various aspects developed within the framework of this study. Another issue concern the reuse of the available material principally the software components which are able to realize a part of it. That is while we choose WEKA (Witten & Frank, 2005) as the backbone of experiment because it is a freeware and it has gained considerable popularity in both academia and research communities.

WEKA was developed at the University of WAIKATO. Written in JAVA, it groups a diversity of data mining algorithms based on bayesian network, neural network, Decision Trees and Statistics. Briefly it is more diversified than the concurrent frameworks. However our intention is to evaluate the performance of this framework according to the principles which have been described previously. This evaluation consist to integrate some new types of algorithms, then to appreciate how PMML can be introduced in this framework to assure pattern exchange with other frameworks.

5.3.1 Extending WEKA

For this purpose, we proceeded to the reverse engineering to rebuild the architectural model of this environment in order to determine the extension points where new algorithms can be plugged in. However, we focused our attention mainly on the classifiers taking into account the implementations available for this stage of our study. Thus we could integrate four new tools without deteriorating the configuration of what exists. These tools are the implementation of the algorithms LEGAL, GALOIS, GRAND and RULEARNER which are based on the lattices as learning model (Mephu & Njiwoua, 2005).

This experiment shows that WEKA provides a set of abstract classes, one for each family of algorithms. A new algorithm must inherit the abstract class relevant to it family, then it

must implement the available abstract methods. The class loader uses a configuration file as tool repository.

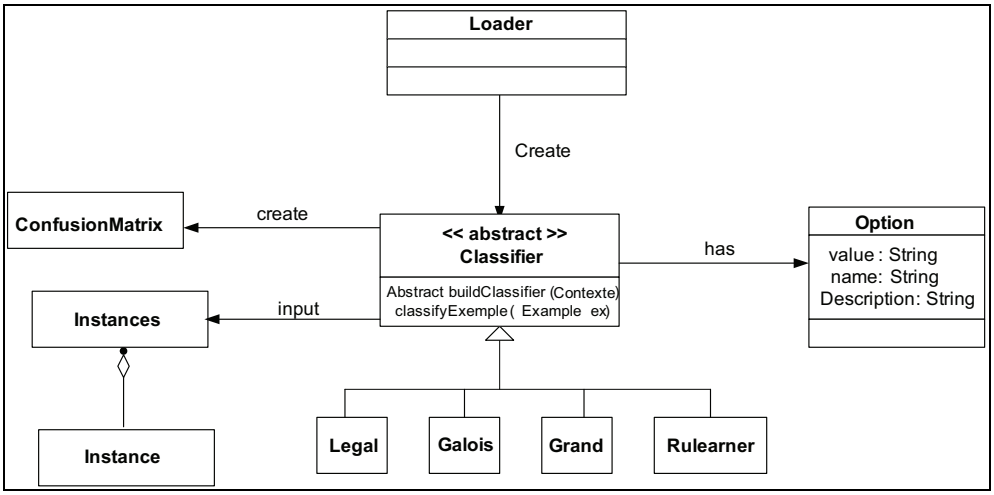


Fig. 7. A partial view of WEKA with the new classification algorithms

In WEKA the extension is done through inheritance. The choice of this approach supposes that algorithms are written in Java. The management of heterogeneity through the reuse of algorithms written in other programming languages relies on the competence of the developer. So the next step will consist to provide the mechanism which could easy the adaptation of heterogeneous algorithms.

5.3.2 Managing interoperability in WEKA platform

The algorithms implements into the WEKA environment (classifier, cluster and association rule) use as in parameter, the data under an owner format (ARFF, Attribute Relation File Format). They produce at output textual unstructured descriptions. The absence of an explicit description of the characteristics of the models thus built does not allow the reuse of extracted knowledge. From this flat structure could be extracted the main characteristics of each type of model. In the case of the association rules, the model consists of all the rules found by the algorithm. The importance of each rule is evaluated through the metric used by the algorithm. In general it is about confidence. Each rule consists of two sets of items (called Itemset) characterized by their support. The first is being the antecedent or premise and the second the conclusion or consequence. One Item indicates an article or a pair attribute/value of the data source. In the case of sequential motifs, the model is rather made up of a set of sequences (called Sequence). A sequence being a chronological series of Itemsets.

Three major concepts permit to characterize the rules independently of the platforms and the algorithms: items, ItemSets and rules. The same reasoning can be applied easily to the other types of model in order to extract their characteristics.

Figura 9 presents the process which has been used to export the models from the algorithm of the associate rule family in WEKA.

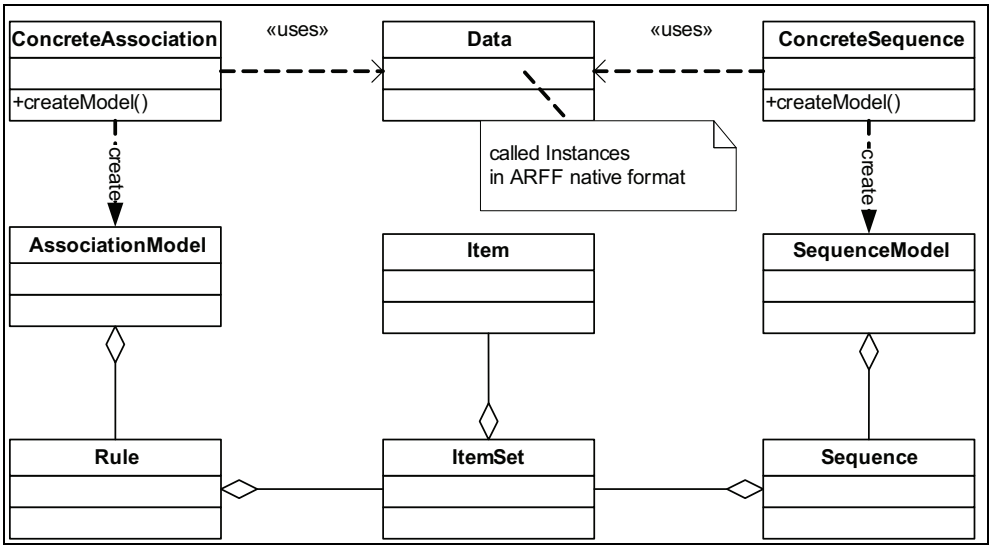


Fig. 8. A view of the association rules model

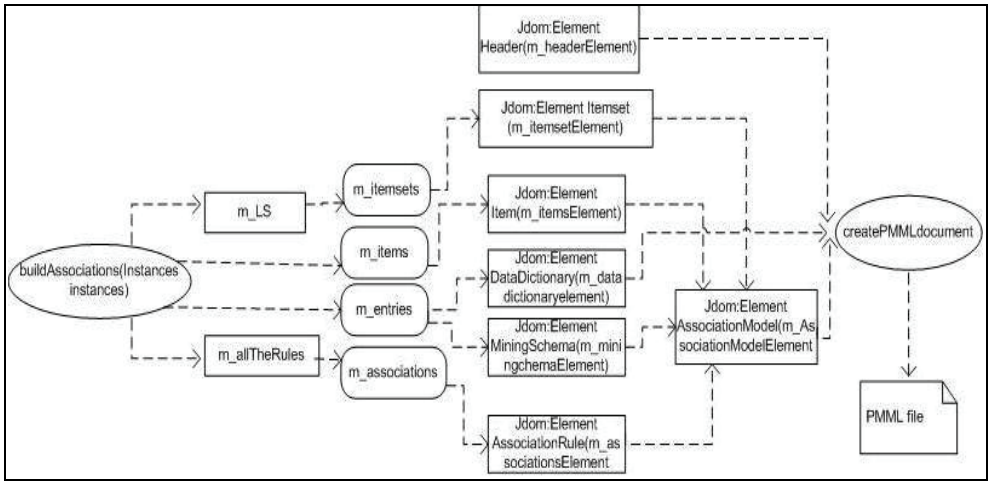


Fig. 9. Exportation of a associate model

From the execution of the association algorithm (buildAssociations operation is executed), the description of the entries of the model (items) is built starting from the instances of ARFF file provides as parameter. The sets of items are built using the structure m_LS which contains the whole frequent items resulting from the execution of the algorithm. The rules are obtained from the structure m_allTheRules. All these parameters make it possible to build the elements of JDOM tree which will be assembled to build the model according to the PMML standard using the method "createPMMLdocument" of the wrapper entity.

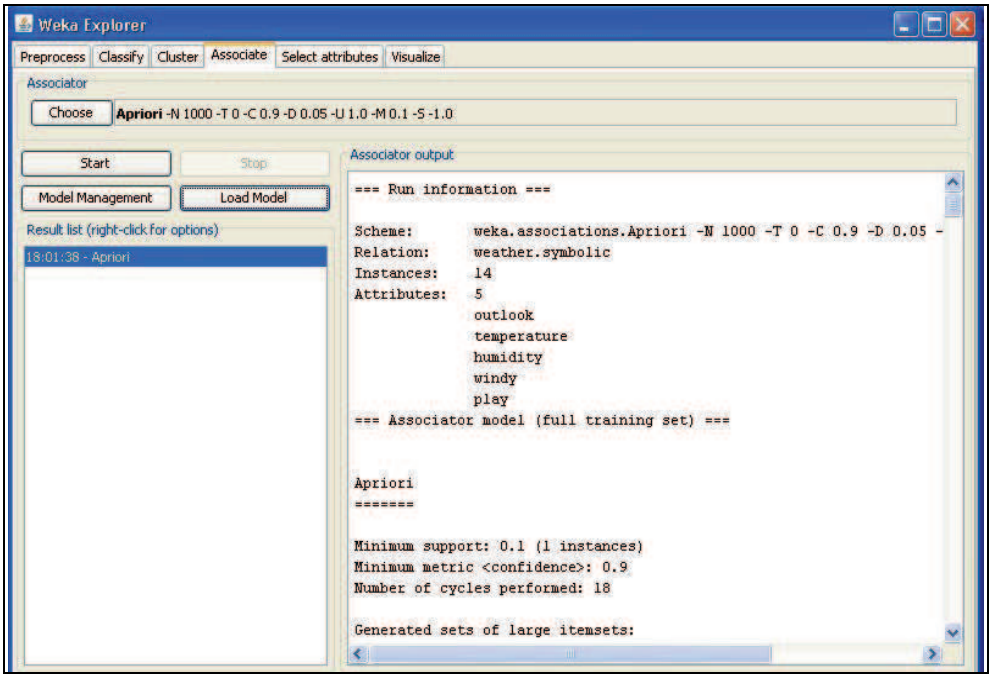


Fig. 10. User Interface with the introduction of the new functionalities

The generated rules and the performance of the algorithms, the number of cycles carried out remain the same. It is proposed the means of handling the model obtained and of being able to export it in PMML format (command button *ModelManagement*) or of importing an external model (button Load Model). A new panel is provided to manage the PMML structure of a model obtained from a WEKA tool or imported from a external framework.

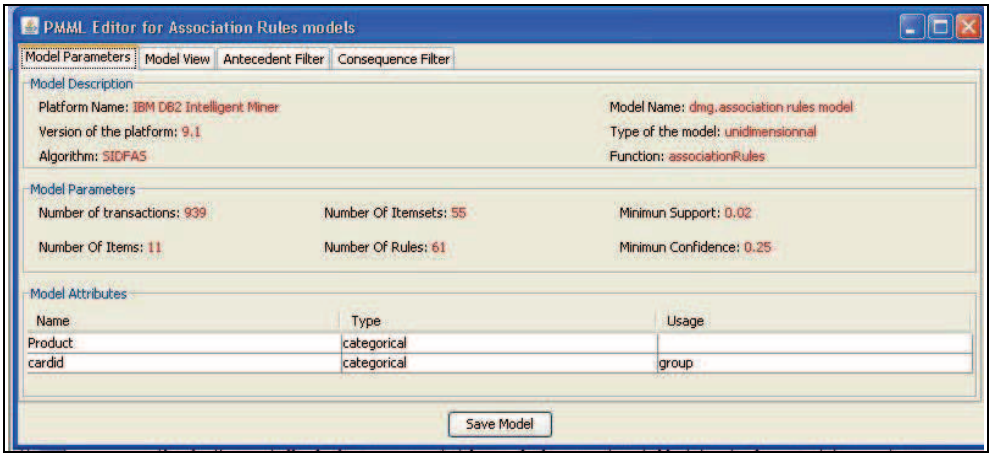


Fig. 11. The panel provided as PMML editor in the WEKA environment

6. Conclusion

Data mining becomes gradually an emergent technology beside the industrial. To favour its use by this target, it is good to be care with the major problems that are facing existing data mining systems: They are based on non-extensible frameworks, they provide a non-uniform mining environment - the user is presented with totally different interface(s) across implementations of different data mining techniques. That is why it is necessary to design a data mining environment which has the following features:

- Open with well defined extension points since new algorithms can be added to the environment,
- Modular because any compatible component is supposed to substitute another,
- Possible integration of different tasks/tools e.g allow to reuse the output of a task by another task
- User flexibility and enablement to process data and knowledge, to drive and to guide the entire data mining process.

This kind of environment is based on a process which is summarized in four steps. Because the components of the environment are made to support the tasks carried out within each step, the process let to the design of the important entities which could be found in such case. This study described the logical structure of data mining environment while insisting on its main issues. In short, integration and interoperability of modern data mining environments may be achieved by application of modern industrial standards, such as XML-base languages. Such synergic application of several well-known standards let the developer gather experiences from other team in the sense that the use of standard favours the reuse of legacy systems (Kurgan & Musilek, 2006).

The adoption of standards in this discipline already made it possible to develop procedures of data exchange between various platforms. At the same time there are reflections on the standardization of a data mining process model. From these efforts, the challenge for the future is to develop and popularize widely accepted standards in data mining environment that, if adopted, will stimulate major industry growth and interest. This standard will promote development and delivery of solutions that use business language, resulting in performing projects faster, cheaper, more manageably, and more reliably.

7. References

A. Abdullah; S. Al-Mudimigh, B. Farrukh Saleem, C. Zahid Ullah. (2009). Developing an integrated data mining environment in ERP-CRM model – a case study of MADAR. *International Journal Of Education and Information Technologies*, Volume 3, N° 2.

Bass, L.; Clements, P. & Kazman, R. (2003). *Software Architecture in Practice*, Second Edition, Addison-Wesley.

Boehm, B. (1998). A spiral model of software development and enhancement. *IEEE Computer* , Vol 21, N°5, pp 61–72.

Cios, K. and Kurgan, L. (2005). Trends in data mining and knowledge discovery. In Pal, N and Jain, L (eds). *Advanced Techniques in Knowledge Discovery and Data Mining*, Springer, pp. 1–26.

Gamma, E.; Helm, R.; Johnson, R. & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*; Addison Wesley.

Goebel, M. & Gruenwald, L. (1999). A survey of data mining and knowledge discovery software tools. *ACM SIGKDD*.

Han, J. & Cercone, N. (2000). RuleViz: a model for visualizing knowledge discovery process, In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discover and Data Mining*, pp. 244–253.

Han, J. & Kamber, M. (2006). *Classification and Prediction, in Data Mining: Concepts and Techniques*, 2nd Edition, Morgan Kaufmann Publishers.

Hauke, K.; Owoc, M. L.& Pondel, M. (2003). Building Data Mining Models in the Oracle 9i Environment, *Informing Science*, pp 1183-1191.

Highsmith, J. (2002). *Agile Software Development Ecosystems*, Addison Wesley, 448 pages.

Ian, T.& Nejmah B. (1992). Definitions of Tool Integration for Environments'', *IEEE Software*, Vol. 9, N° 3 pp. 29−35.

Jorge, A.; Pocas, J. & Azevedo, P. (2002). Post-processing operators for browsing large sets of association rules. *In Proceedings of the ECML/PKDD'02 workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning*, pp. 53–64.

Kimani, S.; Lodi, S.; Catarci, T.; Santucci, G. & Sartori, C. (2004). VidaMine: a visual data mining environment, *Journal of Visual Languages & Computing*, Volume 15, Issue 1, pp 37-67.

Klemettinen, M.; Mannila, H. & VERKAMO, A. I. (1999). Association rule selection in a data mining environment. *Lecture notes in computer science*, vol. 1704, pp. 372-377.

Klosgen, W and Zytkow, J. (1996). Knowledge discovery in databases terminology. In Fayyad, U, Piatetsky-Shapiro, G, Smyth, P and Uthurusamy, R (eds) *Advances in Knowledge Discovery and Data Mining*. AAAI Press, pp. 573–592.

Kouamou, G., E. & Tchuente, D. (2008). Experience with Model Sharing in Data Mining Environments. In Proceedings of the 4th ICSEA, Malta.

Kroll, P. & Kruchten, P. (2003). *Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*, Addison Wesley, 464 pages

Kurgan, L., A. and Musilek, P. (2006). A survey of Knowledge Discovery and Data Mining process models. *The Knowledge Engineering Review*, Vol. 21:1, 1–24., Cambridge University Press.

Mephu, N., E. & Njiwoua, P. (2005). Treillis de Concepts et Classification supervisée. *Techniques et Sciences Informatiques*, Vol 24(2), Hermes, Paris.

Mephu, N., E. (2001). Extraction de Connaissances basée sur le Treillis de Galois: Méthodes et Applications, HDR thesis, Université d'Artois, 2001.

Perrin, O. & Boudjlida, N. (1993) Towards a Model for persistent data integration, In *Proceedings of CAISE'93, the 5th Conference on Advanced Information Systems Engineering*, Lecture Notes in Computer Science, pp 93-117, Paris, France.

PMML (2001). Second Annual Workshop on the Predictive Model Markup Language, San Francisco, CA.

Roy, W. (1970). Managing the development of large software system concepts and techniques. In *Proceedings of the WESCON. IEEE*, pp. 1–9.

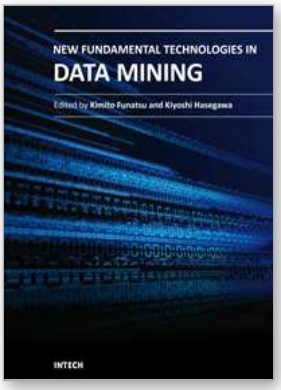
Seifert, J. W. (2004). Data Mining: An Overview, Congressional Research Service, The Library of Congress.

Wasserman, A. I. (1990). Tool Integration in Software Engineering Environments. In Software Engineering Environments. *International Workshop on Environments*, *Lecture Notes in Computer Science* N° 467, pp.137-149. Berlin

Wettschereck, D.; Jorge, A. & Moyle, S. (2003). Visualization and evaluation support of knowledge discovery through the predictive model markup language. In *Proceedings of 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*. Springer, pp. 493–501.

Wileden, J.C.; Wolf, A.L.; Rosenblatt, W.R. and Tan, P.L. (1991) Specification Level Interoperability, *Communications of the ACM* , Volume 34(5), pp. 73−87.

Witten, I. H. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (Second Edition), Morgan Kaufmann, 525 pages, ISBN 0-12-088407-0



New Fundamental Technologies in Data Mining

Edited by Prof. Kimito Funatsu

ISBN 978-953-307-547-1
Hard cover, 584 pages
Publisher InTech
Published online 21, January, 2011
Published in print edition January, 2011

The progress of data mining technology and large public popularity establish a need for a comprehensive text on the subject. The series of books entitled by "Data Mining" address the need by presenting in-depth description of novel mining algorithms and many useful applications. In addition to understanding each section deeply, the two books present useful hints and strategies to solving problems in the following chapters. The contributing authors have highlighted many future research directions that will foster multi-disciplinary collaborations and hence will lead to significant development in the field of data mining.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821