Towards a Pervasive Data Mining Engine - Architecture overview

Rui Peixoto[1], Filipe Portela[1,2], Manuel F. Santos

Algoritmi Research Centre, University of Minho, Portugal [2]ESEIG, Porto Polytechnic, Portugal
ruidfpeixoto@gmail.com; {cfp, mfs}@dsi.uminho.pt;

Abstract. Current data mining engines are difficult to use, requiring optimizations by data mining experts in order to provide optimal results. To solve this problem a new concept was devised, by maintaining the functionality of current data mining tools and adding pervasive characteristics such as invisibility and ubiquity which focus on their users, providing better ease of use and usefulness, by providing autonomous and intelligent data mining processes. This article introduces an architecture to implement a data mining engine, composed by four major components: database; Middleware (control); Middleware (processing); and interface. These components are interlinked but provide independent scaling, allowing for a system that adapts to the user's needs. A prototype has been developed in order to test the architecture. The results are very promising and showed their functionality and the need for further improvements.

Keywords. Data Mining, Pervasive computing, Data mining Engine

1 Introduction

Nowadays, there are many data mining engines. However these engines are difficult to use and optimizing the results takes a large effort. For this reason a new data mining concept was devised. This concept joins the general characteristics of data mining engines with the characteristics of pervasive computing. By bringing the technology into the "background" it is possible to improve the perceived usefulness and ease of use of data mining tools. The Data Mining Engine (DME) architecture proposed is divided into four major components: Database, Middleware (Control and Processing) and Interface. In fact, it provides, at least, the same services as any other data mining engine with extra features. It also provides fully automatic configuration and autonomous data mining services in any place, and in any device available to all users.

Providing Data Mining functionalities and their results (probabilities, dashboards and alerts) automatically and in real-time to anyone, anywhere and anytime is the main goal of this project. This new solution offers an Intelligent Mining and Knowledge discovery to anyone who wants to make previsions without the need to learn how it works. To assess the concept viability and architecture functionality, a case study was performed using the developed prototype. The achieved results are motivating. A

complete and autonomous data mining process was executed using real data collected from Intensive Care Unit (ICU) of Centro Hospitalar do Porto (CHP), Porto.

This article contains five sections beyond this section. The second section provides the state of the art on pervasive computing, data mining and data mining engines. The third section provides an overview of why and how the system was conceived and intrinsic features. The major architectural components are described in the section four. In the section five, a case study is conducted making use of a prototype of the system. The article ends with the conclusion and future work.

2 Background

2.1 Pervasive Computing

Pervasive computing (PC) focus on taking the technology from center stage to the "background" [1], abstracting the user from its complexities. In order to bring the technology to the background a characteristic named "invisibility" is necessary. This concept means that technology is used unconsciously, removing the need for adaptation or understanding of how to use it. This implies the capability to identify and adapt the solution to the environment and its users [2]. To make these decisions there are two different approaches possible, creation of specific models for each environment, or dynamic changing models that detect the environment [3] and then adapt to it. Another key characteristic is ubiquity. This means that technology must be everywhere without the necessity of bringing any specific device anywhere we go [1] so that the user is not aware of its presence. We must understand that this problem is more than a technology problem. Yes its technologically complex, it requires a solution which is a distributed and mobile system, among other concerns, but it has a human component mainly in the way its users perceive it. It may also automatically notify the user about pre-defined requests using the best available device or method: situated devices, email, phone application, etc., depending on the user location.

Satyanarayanan [4] considers pervasive computing as an evolution of distributed systems and mobile computing. Most of the challenges are addressed and a direct solution can be implemented into PC. Challenges in distributed systems [5] such as heterogeneity, openness, security, scalability, failure handling, concurrency and transparency must be addressed and resolved. In mobile computing wireless networks, mobility and portability are the challenges to consider and address [6]. As determined earlier pervasive computing bring new challenges [7]: Localized scalability, physical spaces Heterogeneity, Integration, Invisibility, Context awareness and management.

As advantages it allows the technology to be used by all the people, removing the need for adaptation and resistance to change. In more specific situations, pervasive computing allows a greater comfort in life thanks to smart spaces [8], higher productivity with access to information and computation anywhere [1], and automatic devices configuration [9]. Concluding pervasive computing promises a new age of computation, focused on the people with technology in second place, promoting direct, simple and intelligent access to information and services anywhere [10].

2.2 Data Mining

Data Mining (DM) is defined as the application of algorithms to the discovery of patterns in data [11][12], in order to potentially find useful information. There are two different objectives or categories in Data Mining, prediction and description [13]. Predictive modelling produces a model of a system based on initial data, and its objective is to predict a specific attribute, based on others. If the attribute to be predicted is numeric or continuous, then regression [15] is used, if it is discreet, classification is used [14].

Descriptive modelling creates patterns that describe data and its objective is to allow the interpretation of those patters. There are four main approaches: Clustering, Summarization, Dependency [16] and times series. Algorithms are implementations of generic models (classification, regression, others). In DM there are different algorithms (decision trees, neural networks, others), some are specific to a model type, and other encompass several model types [17]. This project only attempts to solve classification and regression problems. In order to determine the quality of a model it must be evaluated. To perform this, the simplest way is to divide the data into training and evaluation sets. Normally 1/3 of the data is used for evaluation and the remaining for training [12]. Dividing the data in this faction may not be optimal. To solve this problem, there is a process called stratification, which guarantees that each class is properly represented in both data sets, but this is not enough to guaranty adequate representation. For this a common statistical technique is normally used, called cross-validation. Several different partitions of the same data are created, called folds. These two methods [12] are the ones available in the prototype although the architecture allows for other methods to be implemented. There are several metrics [12] to evaluate a prediction model depending on the problem objective / model type. Scoring functions quantify the fit quality of the model created, its usefulness is to compare the fit between the models [15]. Without scoring functions it is impossible to determine the best model or even optimize the parameters of the model or to find the probability associated to a target.

2.3 Process

Specifically for this architecture a four stage data mining process is used, composed of Extract Transformation Load (ETL), Modelling, Model Induction and Evaluation. Each stage is composed by several tasks. ETL is composed of data collection, exploration analysis, data transformation and data selection. Modelling is composed of model selection and model configuration. Model Induction has no sub tasks. Evaluation is composed of model evaluation, process result evaluation and scoring. At any time the process can be returned to the any previous task in the process. The process ends when the model evaluation coincides with the process evaluation target.

2.4 Data mining engine and similar engines

Data mining engine is a mechanism that offers a set of data mining services to its clients. It can be seen as if it was a black box, everything done inside is invisible to its users,

only displaying its services as an input and output. The services can be at a process level or very specific tasks, it depends on the data mining engine capabilities.

There are many data mining engines, ranging from specific tools only providing data mining services, to business intelligence packages with data mining functionalities [18].

Engines like R, Weka, Knime, Rapid miner are well known in the data mining world, but they all suffer from the same problem, ease of use. Data mining engines are considered more difficult to use than other information technology [19], being the two most important factors in tools adoption the ease of use and perceived usefulness. This new engine tackles these two factors. Its perceived ease of use is increased by providing a fully automated process, removing the need of a specialized data mining expert and bringing the potential of data mining to everyone. The perceived usefulness of the data mining engine is increased by autonomous optimization and notification system.

3 The concept

From the limitations facing current data mining engines a new concept emerged. By joining the characteristics of pervasive computing and data mining a new engine was developed. By devising the system as a distributed systems it can serve multiple users in multiple places/devices. Pervasive computing also goes much further than a concern for user interface, it requires the system to perform tasks without user intervention, by recording the entire process a knowledge base is created for the system and to apply Data Mining algorithms to make better decisions (e.g. first model to use). It will use past processes to make its decisions. This allows the system to mold to its users, even user choices will be reflected in future system choices. To better understand how this was accomplished this new engine is capable of:

- Scaling according to the needs – The system can run on single or multiple server, and can scale each component independently according to the number of processes and users.
- Multiple users – The system is online and designed to service simultaneous users.
- Services anywhere and in any device – As it is a web application it can be accessed from any place from any device.
- Uses other engines for data mining tasks – The system is capable of using other engines to run its data mining tasks, it basically is an abstraction on top of other data mining tasks. The system does not implement new algorithms, it uses code and libraries already deployed to support any data mining service.
- Services to other engines for data mining tasks – Because of its ease of use, online availability and minimal configuration it can be easily used by other programs for easy access to data mining services.

Service to:

- Novices – Because of its automatic process, novice users can start using data mining services without needing to understand the technical necessities. The only minimal knowledge required is a business knowledge in order to understand the significance of the results.

- Experts – Because of the pervasive characteristics, the system must respond to the expectations of its users, providing an environment to expert users, able to perform any task than other DME provides, adding some extra features.

Services that are:

- Automatic/Optimized – The system provides several levels of abstraction and automation. This provides many possible ways to operate the system, combining expert knowledge and system knowledge to suite every user expectation. The system also provides optimization at several steps, such as data selection and model configuration for example.
- Concurrency – Because of its design, it is capable of running multiple models in concurrency, limited only by the physical resources available.

4 Architecture

4.1 Requirements and Features

This project has several requirements derived from the three areas regarding pervasive computing and data mining itself. The requirements are:

- Scalable – In order to meet this requirement a distributed system solution with some component replication was selected. This was done in order to provide not only scalability but also better performance according to the task at hand.
- Available in every device and operating system – For this requirement a web solution was designed, requiring only an html browser application in the device.
- Physical – The physical requirements of the system are as broad as the possible environments. It can run on a single machine, for small data mining projects, to multiple servers providing resources for large companies.

The main features are:

- Pervasive/Ubiquitous – The system is always available anywhere in any device.
- Distributed system – The system operates in several machines, performing its tasks in specific environments, being the access remote to the server.
- Persistence – The entire system is stored in the database, including its decisions.
- Automatic configuration – The system requires none or minimal configurations.
- Expert/Novice – The system provides an interface for varying types of users.
- Optimization – The system has built an optimization process, removing the need for manual optimization of the entire process.
- As a service – The system is designed to support multiple users and working as a service, not as a desktop application.
- Adaptability – Allow easy implementation of new models and data mining engines, allowing the customization even to specific areas of data mining.
- Scalability by component – The system provides scalability by component. Allowing a better suit of the hardware resources and performance concerns.
- Multiuser - The system is designed to work with simultaneous users.
- Privacy – Because it is a multiuser environment the system allows for the encryption of that, for privacy and security reasons.

4.2 Benefits

The principal benefit is the possibility to use DM efficiently without years of training. All tools require making choices that impact significantly the results and without experience it is impossible to know which is best. Also running all manually or making your own tool to automatize the process is time consuming or impossible to some users. It can also be very useful as a learning tool seeing as the system runs automatically the user is able to view every choice it makes. On a more technical note, it supports new tools/algorithms that may arise, as it uses other tools to provide its modelling services, without having to make major changes to the system.

4.3 Description

As mentioned before, this architecture (until now) only solves classification and regression problems. It internally uses some descriptive modeling, joined with prediction to attempt to run the better probabilistic model first. Because of this the architecture is far more complex than the simple process explained earlier. This architecture is composed of four major components: Database – Responsible for the persistence of the system; Middleware (Control) – Composed of the three other components, it manages all the decisions, the servers and the process; Middleware (Processing) – Component where almost all of the processing is performed, it runs the models and the ETL; Interface – Handles all client operations to the system. These four components are in constant communication and require each other to properly function and each component is responsible for its own fault tolerance. Each major components is comprised of several sub-components. Each components is described in the sub-sections below.

a) Database

A major component of the system, it provides persistence for the entire system. But it is more than a way to save data. The system runs completely on top of the database (DB). Many events are triggered when a change is detected, some processes are notified at the Middleware level, but no task is started without the confirmation of the database. The system simply does not work without the database. The communication is constant and intensive (requiring high physical resources), this is a conscious and intentional decision. Today the systems are very responsive and reliable, by delegating responsibilities to the database, synchronization efforts at the Middleware level are alleviated by preventing common problems of synchronization at the middleware level. Because of the requirements, the DB is responsible for its own scaling, an independent system. Allowing the administrator to implement a DB that handles the load required is depending on the user's needs. Requiring only that the connecter is changed according to the DB. Implementation is very important on this subject, currently any SQL DB is easy to implement as long as a java connecter exists to that DB system.

b) Middleware (Processing)

Comprised of one or multiple servers. All the ETL, modelling, evaluation and scoring tasks are performed in this layer. As a performance and diversity concern, it allows any other data mining tool functioning on this system. Engines such as R, Weka can be used

to perform any of the tasks, even if a new engine appears, as long as a command line is available it can be used as a component in this system. This functionality allows incredible adaptability, requiring minimal implementation. This can be achieved by creating a connector for each tool that is able to perform datamining task. By defining the entry and end point of each task, as long as these stay the same, the engine is able to perform these tasks. This can be achieved by programing scripts for each task in the control and send it to processing by the connector. Due to the fact of each task being independent to any other task, specific tools can be inserted to perform only specific tasks. For instance the ETL can be performed on R and the modeling on Weka or vice-versa, because each task is recorded on the database, any other tool can continue the process where it was stopped. The only requirement for this system not failing, is for the entry and exit points to be the same. For example, if two models are to be performed, one in R and another in Weka, both models will load same data, and it will record the same information (necessary for the next task), this ensures that implementations are compatible with the current process. The differences are in the task itself, not in the process. All the tasks are available, allowing them to be called from the control/interface, or from other software/devices. The scalability issues are solved by replicating processing servers. Several servers of the same type can be started. The system will scale linearly, as each server processes only one model at a time, if there are ten models to be performed and ten physical servers available they will start one in each server. This example was for modeling process but the same tasks is applied to all other tasks. Several ETL servers can be started depending on the needs of the system. This is not possible in a standard desktop data mining engine (DMEs available in the market at the moment).

c) Middleware (Control)

The core of the system, capable of individual scaling with repetition in each of the sub components except server control, that requires scaling throw partition responsibility.

- Server Control – Responsible for the processing servers, it controls the starting of all the ETL, modelling, scoring and evaluation tasks. It determines the available servers and assigns the priority tasks for each server. Every single task started in Middleware (Processing) has to go through server control.
- Modelling Control – Responsible for generating all the scripts for running and configuration of each individual algorithm implemented in the system. It is a separate system to allow a better division of the code, so that future implementations of new models are easier to perform and comprehend without compromising the integrity of the entire system.
- Process Control – Responsible for the entire DM process it provides individual calls to each DM task, allowing for each individual task to be performed at any time. It also controls the flow of the process, which tasks are performed and in what order. The process is terminated when the target evaluation result is reached, this target can be the default of the system or defined by the user.
- Decision Support System (DSS) – It is where the major decisions are made: transformation, data selection and modelling. To make these decisions, several DM processes are induced. Using the data accumulated from previous processes, a model is created in order to score new data presented to the system. When a decision is requested the DSS always uses the best model

available at that time. The system automatically maintains the model or change when a deviation is detected.

d) Interface

Composed by one or more webservers (depending on number of users). It is able to scaling independently of the other major components. It provides access to data mining services, configuration to the user and administrator, notification medium for the user, and reviewing the results. The notification can also be made by email or message. The interface also provides a different layout (interface) for each type of user.

- Simple – Provides nothing more than dataset loading, define the prediction target, and show the current results. It is designed to be as simple as possible providing only information strictly necessary.
- Advanced – On top of the simple functionalities it provides information on the decisions made and the current stage of the process. It allows the user to define any task manually or a mix of manual and automatic task. For example it allows the automatic selection of attributes and model. It also allows the change of evaluation stopping target.

5 Case Study

A first version of the solution was tested using real data collected by INTCare project [20-21] in the ICU of CHP. The current development still does not have a graphical interface but already implements a partial process, from start to finish. The automatic transformations and the intelligent system inside DSS are not implemented. Excluding these features the system is up and running on a server with two R processing threads, a MySQL database and a java Middleware. The system loads the data, selects the model and performs the induction, evaluation and scoring, stopping when the target evaluation is reached. The dataset used has 214709 records, 32 columns with information about patient vital signs. The target is a numeric field with two possibilities (1 – critical, 0 – stable). The goal was to predict if the patient is in critical or stable condition [22].

The process starts with the input of the dataset. This is done manually by the user, as well as selection the column for the target. The system automatically loads the dataset and generates descriptive statistics about the dataset (column data type, number of classes, max, min, and others). After loading is complete, the model is selected. A set of predefined configurations was selected by the system, according to the priority levels defined to each configuration, this information is stored in the database. After the configuration is terminated, the system detects which servers are available and assigns the models for induction. The instructions for processing are sent from the control to the processing, and the model is created, evaluated and all the results are saved directly to the database. When this is finished the control is notified, and the process checks if the evaluation target is reached using the scoring process, otherwise a new model is assigned to be inducted. The system records all the available metrics. In this test the metric used was relative absolute error, defined at the start as 25%. The target evaluation was reached on the 4th model and the system stopped. As mentioned above, it still does not have a graphical interface as such the user notification is not yet

implemented. All the request, even the request for loading initial data is recorded in the database, and the server control is then notified. The processing server access directly to the databases to minimize transactions, when it finishes the request, it notifies the server control and/or process control depending on the task it is performing. After model selection is determined and the configurations are defined in the database, the server control allocates the script to the processing server. When evaluation target is not reached, the connector notifies process control with an unsuccessful status. The process control then determines if new models or parameters are needed.

6 Conclusion and Future Work

In this paper was presented a new data mining engine with pervasive characteristics. Although the system is still in development, the concept is well defined and a working prototype was deployed. Its capabilities for simple and fast data mining services is undeniable. The tested prototype is capable of looking for optimal results, and the architecture works as expected providing the base for future and improved developments. On a more specific note, it is apparent that the information generated by the system provides useful information for all types of users, be either for new users learning how data mining works, at home or in school environments, or by expert users, providing new, unexplored paths to achieve the same or better goals.

At the moment many possible data mining approaches are explored with minimal knowledge of which will produce the best results, improving the system in this subject is the real challenge. We cannot also ignore that the system is designed to learn from itself, requiring more time to produce better results.

The main difference between the DME proposed and the existing one is in the concept novelty and in the architecture. This approach is totally new for the scientific community. From the development made it can be concluded that the characteristics of pervasive computing and data mining can indeed be joined to create a new concept as explained in this article. The architecture presented has already proven it can minimize the technical expertise needed to use a data mining engine. A case study demonstrated that interesting results can be attained by the system. Nevertheless it does seem that the architecture devised is capable of performing and reaching the goals defined in the project, requiring improvements in the decision support system.

Briefly and as main gain with this solution the user only need to load a dataset, choose a target and then click in start. Then the system is responsible by treating the data, induce and evaluate the models. Finally the scoring tasks is performed and all the probabilities are presented to the user.

Future research should be focused in the decision support system and in a statistic module. Two major features will be included in the system. First, a clustering approach as a description and prediction modelling. The clusters allow not only for one more type of data mining but also to explore large datasets. Second another descriptive data mining techniques to provide metadata to the cluster modelling. This way, better and dynamic models can be created. So in the future the models does not need to have predefined classes, the cluster can be used to decide which classes should be considered based in their relation with the target.

Acknowledgment

This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the Project Scope UID/CEC/00319/2013 and the contract PTDC/EEI-SII/1302/2012.

References

1. M. Weiser, “The computer for the 21st century,” Sci. Am., vol. 265, no. 3, pp. 94–104, (1991).
2. M. Weiser, “Some computer science issues in ubiquitous computing,” Commun. ACM, vol. 36, no. 7, pp. 75–84, (1993).
3. K. Lyytinen and Y. Yoo, “Issues and challenges in Ubiquitous computing,” Commun. ACM, vol. 45, no. 12, pp. 63–96, (2002).
4. M. Satyanarayanan, “Pervasive computing: Vision and challenges,” Pers. Commun. IEEE, vol. 8, no. 4, pp. 10–17, (2001).
5. G. F. Coulouris, J. Dollimore, and T. Kindberg, Distributed systems: concepts and design. pearson education, (2005).
6. G. H. Forman and J. Zahorjan, “The challenges of mobile computing,” Computer (Long. Beach. Calif)., vol. 27, no. 4, pp. 38–47, (1994).
7. D. Saha and A. Mukherjee, “Pervasive computing: a paradigm for the 21st century,” Computer (Long. Beach. Calif)., vol. 36, no. 3, pp. 25–31, (2003).
8. W. Mark, “Turning pervasive computing into mediated spaces,” IBM Syst. J., vol. 38, no. 4, pp. 677–692, (1999).
9. G. Banavar, J. Beck, E. Gluzberg, J. Munson, J. Sussman, and D. Zukowski, “Challenges: an application model for pervasive computing,” in Proceedings of the 6th annual ICMCN, 2000, pp. 266–274 (2000).
10. J. Ye, S. Dobson, and P. Nixon, “An overview of pervasive computing systems,” in Ambient Intelligence with Microsystems, Springer, 2008, pp. 3–17 (2008).
11. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and others, “Knowledge Discovery and Data Mining: Towards a Unifying Framework.,” in KDD, 1996, vol. 96, pp. 82–88 (1996).
12. I. H. Witten, E. Frank, and A. Mark, “Data Mining: Practical machine learning tools and techniques.” Morgan Kaufmann, San Francisco. Retrieved, (2011).
13. M. Kantardzic, “Data-Mining Concepts,” Data Min. Concepts, Model. Methods, Algorithms, Second Ed., pp. 1–25, (2011).
14. P. S. Bradley, U. M. Fayyad, and O. L. Mangasarian, “Mathematical programming for data mining: formulations and challenges,” INFORMS J. Comput., vol. 11, no. 3, pp. 217–238, (1999).
15. D. J. Hand, H. Mannila, and P. Smyth, Principles of data mining. MIT press, (2001).
16. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” AI Mag., vol. 17, no. 3, p. 37, (1996).
17. C. Giraud-Carrier and O. Povel, “Characterising data mining software,” Intell. Data Anal., vol. 7, no. 3, pp. 181–192, (2003).
18. R. Mikut and M. Reischl, “Data mining tools,” Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 1, no. 5, pp. 431–443, (2011).
19. T. C.-K. Huang, C.-C. Liu, and D.-C. Chang, “An empirical investigation of factors influencing the adoption of data mining tools,” Int. J. Inf. Manage., vol. 32, no. 3, pp. 257–270, (2012).
20. Portela, F., Santos, M.F., Machado, J., Abelha, A., Silva, Á., Rua, F.: Pervasive and intelligent decision support in Intensive Medicine–the complete picture. Information Technology in Bio-and Medical Informatics, pp. 87-102. Springer (2014)
21. Aguiar, J., Portela, F., Santos, M.F., Machado, J., Abelha, A., Silva, Á., Rua, F., Pinto, F.: Pervasive Information Systems to Intensive Care Medicine. ICEIS 2013 246 (2013).
22. Portela, F., Gago, P., Santos, M.F., Machado, J., Abelha, A., Silva, Á., Rua, F.: Pervasive real-time intelligent system for tracking critical events in intensive care patients. IGI. (2013).