Implementing privacy-preserving filters in the MOA stream mining framework

David Martínez Rodríguez

Arquitectura del Software
Facultat d'Informàtica de Barcelona, UPC

October, 2014

Contents

1 Introduction

The present work is the final report for the Project Management module of the Degree's Final Project. This project will be carried out at the Barcelona School of Informatics and will be directed and supervised by Jordi Nin Guerrero, from the Computer Architecture department.

1.1 Context summary

Although a more thorough definition of the project's context will be given in section 3, we will layout now the basics, in order to understand the scope and goals of the project.

1.1.1 Data mining

Today's information society produces vast amounts of data all over the world. This data comes from innumerable sources and in diverse formats, and has been stored for years in data warehouses, waiting to be processed. Nowadays, all progress made in both hardware and software fields allows us to exploit this stored data and distill knowledge from it, through a series of techniques known as *data mining*.

This is indeed a holistic process, where many different disciplines are involved, from data acquisition and storage, through its selection, filtering and analysis up to information extraction, visualization and knowledge discovery.

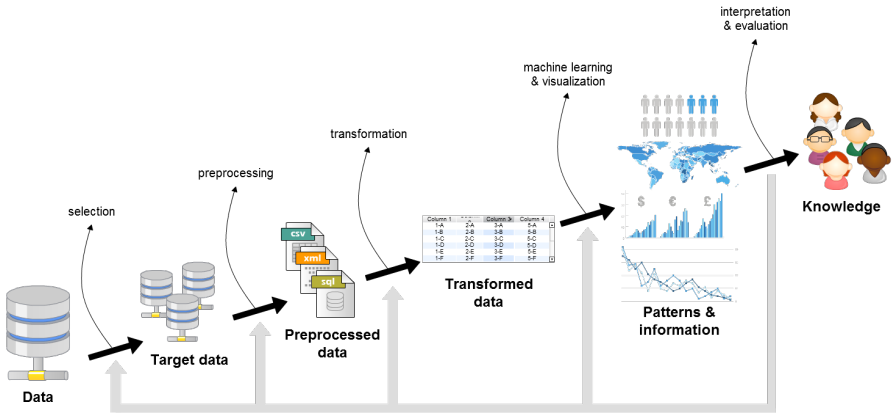

Figure 1: Data mining as a process, from data acquisition to knowledge discovery. Source: adapted from *From Data Mining to Knowledge Discovery in Databases* [4]

Data mining enables a better understanding of human or natural processes and provides us with means to identify trends, predict future events or discover useful patterns. Its uses range from scientific and medical applications to social sciences or business administration [4].

Despite lots of effort is put into enhancing different data mining processes, there still are many cases where these techniques fail to perform correctly; mainly, it is a matter of scale. On one hand, traditional data mining workflows cannot cope with the really massive data sets that are available nowadays, if performed on a common infrastructure. To solve this issue, clusters of hundreds or thousands of computers are used to run such analysis. It is costly and complex but, doing so, we can mine data that we couldn't some time ago.

On the other hand, we face another type of scaling problem. In some situations, data acquisition throughput is so high that it can't be stored anyway, so another approach is needed to avoid the loss of information that it could deliver us. Moreover, it could be that we didn't want to store it, even when we could, but yet we wanted to analyze it to extract knowledge from it, as soon as we received it. Both scenarios are addressed with a series of techniques known as *stream mining*.

1.1.2 Stream mining

Stream mining or data stream mining is a process that allows us to still discover knowledge and patterns in data, even when it comes in the form of a continuous stream, or many of them [6]. Instead of processing all statically stored data, like traditional data mining does, a relatively small portion of it is kept during the analysis, and it is updated when needed - either because more resources are available to the system or because new data is acquired.

MOA, initials for Massive Online Analysis, is an open source framework for data stream mining [7], originally developed at the University of Waikato, New Zealand. It includes several machine learning algorithms[1], to perform the analysis, tools to evaluate the quality of the results and also deals with a problem known as *concept drift*[2]. It is related to the Weka[3] package, but it is built to perform at a greater scale for more demanding problems.

1.1.3 Privacy & data mining

Privacy has become a hot topic in debates nowadays, concerning what information is collected from individuals, who owns it and with which purposes. It is a matter of great importance and certainly worth to be examined carefully. Information technology brings us many benefits at many levels - safer streets, cheaper communications, better health systems, more convenient shopping - but at the high cost of losing our privacy.

[1] Algorithms used to perform the actual data mining analysis (the "machine learning & visualization" step on figure 1) belong to the field of machine learning. In MOA, clustering, classification, regression, outlier detection and recommender systems are available.

[2] It is said of statistical properties of a target variable being analyed, when they change over time in unforeseen ways.

[3] Weka is a popular software package including classical data mining algorithms, this is, not stream mining. It is also developed at the University of Waikato. [8]

Data mining is highly related to privacy. These knowledge discovery processes need data to work and, in most of the cases, it is sensitive personal data, which is massively gathered and stored and analyzed without us knowing much about it. Apart from the lack of consent in this data acquisition stage of the process, data mining poses a bigger thread on individuals: information disclosure.

A number of procedures have been developed to avoid information leaks at the individual level, while still being able to get knowledge from aggregated data. Different communities have worked on this area, which is called *statistical disclosure control* by some or *privacy preserving data mining* by others.

1.2 The project in a nutshell

Having presented its overall context, the main purpose of this project is to *implement some privacy preserving data mining procedures within the MOA stream mining framework.* A more detailed description of the project is given in section 2.1.

2 Project management

We will discuss in this section everything concerning the management of the project: *scope*, *schedule* and *budget*. However, we must stress that this classical approach of management analysis is not really suited for our needs. Instead, a more *Agile*[4] methodology will be applied. We cover this on the Methodology section, but there is an important conceptual change to be taken into account: the different driving force of the project. Whereas in classical project management the scope-schedule-budget triad is what must be controlled, in an Agile project management approach it is *value*. Indeed, *quality* must be ensured so maximum value is delivered to the project's stakeholders, thus being scope, cost and schedule constraints to these primary goals.

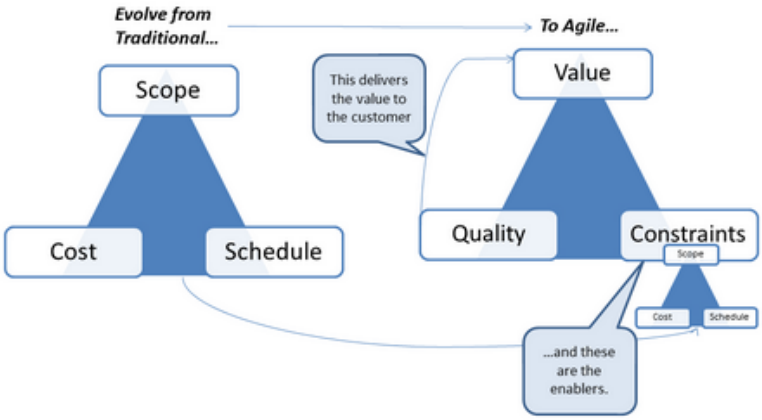

Figure 2: Traditional to Agile project management evolution. Source: [5]

2.1 Scope

One of the first things to do, when beginning any project is delimiting its scope, this is, deciding *what* will be done and *how*, in terms of resources and methodology, for example.

2.1.1 Requirement analysis

We already stated in section 1.2 what the main goal of this project is. A more detailed list of the project's requirements is the following one:

- **Functional requirements:**
  1. Implement privacy preserving stream mining *filters*[5] for the MOA stream mining framework. The suggested algorithms to be implemented are:

[4] Agile software development is based on the *Agile manifesto* [3].
[5] Within the MOA context, *filters* are procedures applied to data prior to their analysis using machine learning algorithms.

a) Noise addition [2, p. 54]

b) Multiplicative noise [2, p. 57]

c) Microaggregation [2, p. 60]

d) Rank shuffling [2, p. 73]

e) Differential privacy [1]

- **Non-functional requirements:**
  1. **Correctness:** privacy protection is at stake in this project, so algorithms must be implemented correctly, from the theoretical point of view, in order to not ease information disclosure when they are used.
  2. **Efficiency:** given that no data mining process can scale well if its algorithms are slow, effort will be put in making them the most efficient we can.
  3. **Test coverage:** measures and tests will be performed to assess the quality of the developed software, as well as its scalability and performance, which is paramount in this project's context.
  4. **Documentation:** MOA is an *open source* data mining framework, which means that its community can assess how is it built and how to improve it. One of the benefits of the open source development model is that software can be safer, more robust and efficient, by receiving contributions from different developers. If people are to continue improving the work done, it has to be well documented.

2.1.2 Scope risks analysis

The methodology approach used in this project will be based on Agile principles. This involves several decisions on how to manage the project and its requirements.

In this particular project, if we are to examine the classic constraints that we talked about at the beginning of this section, we do know that the schedule is fixed (perhaps not the planning, but the final milestone) and this forces us to let the scope opened. This is, we will implement as much features as we can, assessing their quality, but no feature list will be fixed from the beginning of the project.

Because we will be working on the basis of an *open scope*, deviations in this field are likely to happen. These, however, will not pose to be a project failure in any case, because it has been agreed to be developed this way.

2.1.3 Methodology

Agile methods will be applied throughout the development phase of this project. Some of the key concepts and practices in this respect are:

- Short to mid range development **sprints** (phases), in order to keep track of the project's evolution and to be able to react to changes, unforeseen constraints or scope drifts.
- **Constant meetings** with the project's stakeholders, in which the progress and deviations of the project will be assessed. Measures to alleviate them will be taken in these meetings.
- Use of **burndown charts** - graphical representations of work left to do versus time.

2.2 Schedule

2.2.1 Overall duration

Taking a general look at the project's schedule, we can estimate it to have a total duration of about 5 months. Even though it was registered on July, 2014, the project did not begin until September, because August is the only month I can have holidays, due to job restrictions. Considering the next possible project's lecture shifts, we believe that the one taking place in December is too close in time. Thus, the project will endure until January the 26th, 2015. This should give us time enough to develop the project and document it without too much pressure, which is key to fulfill one of the main established goals: high quality results.

2.2.2 Schedule slack

The project schedule we present herein does not fill up the total amount of time available - more than two weeks are left blank, with no assigned tasks. This is intended because of the following reasons:

- The amount of time needed to develop the proposed algorithms is uncertain. It is hard to estimate the time it may take, because I have no previous knowledge on the area. Therefore, we opted for, in one hand, an open scope approach, and, on the other, leaving a considerable time gap between the last planned task and the project's final milestone: its defense. Being conservative, if the development of any proposed method is delayed, we still have some leeway to introduce schedule changes, without risking the project's success.
- We have estimated the project's report confection and the defense presentation rehearsals to be 35 and 7 days, respectively, but depending on how much development is finally carried out, it might not be time enough to write down the report. Extra time for doing it can be then borrowed from the schedule slack time.

2.2.3 Schedule monitoring & changes

For the development phase of the project, the most suitable way to monitor the schedule we have found is applying an Agile approach to the process. We will work in one week

long sprints, meeting every week to assess the quality of the solutions, the proper progress of the project and to plan what will be done during the following sprint.

Sprint planning meetings are where the main goals of the project will be sliced in small tasks, which can be tracked and implemented better, because they are not so complex. Thanks to this constant fine-grained planning process, schedule or scope deviations are detected earlier and can be managed efficiently, reacting before they affect deeper the overall success of the project. Given that no fixed features list is assigned to each sprint of the development phase, if the completion of either of those features is delayed, it can be made to span for some more time.

Within each of the development sprints, burn downcharts will be used to monitor the progress of the sprint. These charts are helpful in identifying patterns of work (sprint-end rushes, for example) and can help developers maintain a constant rate of finished features.

Besides burn down charts and sprint planning meetings, the use of velocity charts will also be helpful to increase the predictability of the following sprint plannings. The more predictable they are, the less deviations will occur and the schedule will be more likely to be fulfilled.

2.2.4 Detailed schedule

2.3 Budget

3 Context

References

[1] Cynthia Dwork. Differential privacy. In *in ICALP*, pages 1–12. Springer, 2006.

[2] Anco Hundepool et. al. *Statistical Disclosure Control*. John Wiley & Sons, Ltd.

[3] Martin Fowler et. al. Agile manifesto for software development, October 2014.

[4] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. Advances in knowledge discovery and data mining. chapter From Data Mining to Knowledge Discovery: An Overview, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[5] Shane Hastie. Agile australia - opening keynotes, October 2010.

[6] Anand Rajaraman and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA, 2011.

[7] New Zealand University of Waikato. Moa - overview, October 2014.

[8] New Zealand University of Waikato. Weka 3 - data mining with open source machine learning software in java, October 2014.