Implementing privacy-preserving filters in the MOA stream mining framework

Degree Final Project, major in Computer Science

Project Management module report

Author:
David Martínez Rodríguez

Supervisor:
Jordi Nin Guerrero

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya

October 12, 2014

Contents

Contents 2

1 Introduction 3
- 1.1 Context summary 3
  - 1.1.1 Data mining 3
  - 1.1.2 Stream mining 4
  - 1.1.3 Privacy & data mining 4
- 1.2 The project in a nutshell 5

2 Project management 6
- 2.1 Scope 6
  - 2.1.1 Requirement analysis 6
  - 2.1.2 Scope risks analysis 7
  - 2.1.3 Methodology 7
- 2.2 Schedule 8
  - 2.2.1 Overall duration 8
  - 2.2.2 Schedule slack 8
  - 2.2.3 Schedule monitoring & changes 8
  - 2.2.4 Project phases 9
  - 2.2.5 Detailed schedule: Gantt chart 10
- 2.3 Resources 13
- 2.4 Budget 13
  - 2.4.1 Budget estimation 13
  - 2.4.2 Total budget estimation 15
  - 2.4.3 Budget control mechanisms 16

3 Context 17
- 3.1 Data mining 17
- 3.2 Privacy 17
  - 3.2.1 Privacy leaks consequences 17
  - 3.2.2 Legal framework 18
  - 3.2.3 Privacy and data mining 18
- 3.3 Involved stakeholders 19
  - 3.3.1 Users: organizations and developers 19
  - 3.3.2 End users: data providers 19
- 3.4 State of the art 20
  - 3.4.1 Stream mining 20
  - 3.4.2 Privacy preserving stream mining 21
- 3.5 Environmental impact 23
- 3.6 Social impact 23

References 24

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.

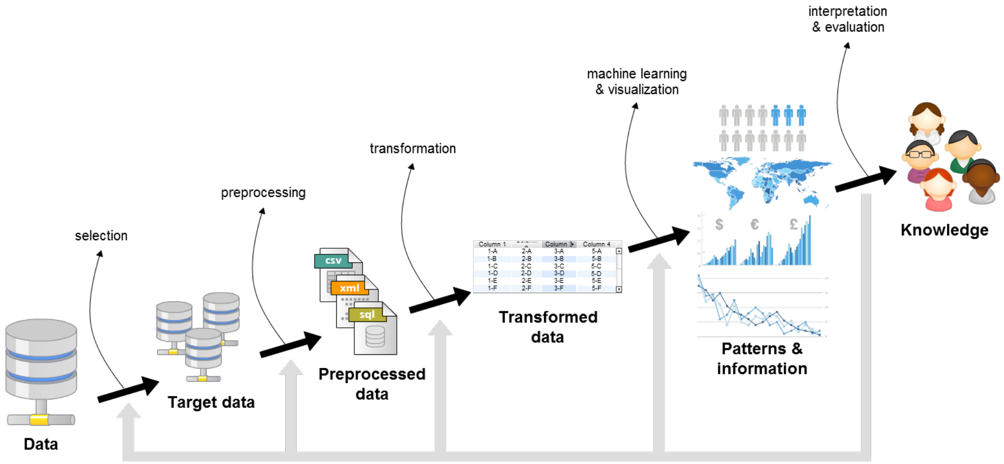


Figure 1: Data mining as a process, from data acquisition to knowledge discovery. Source: adapted from *From Data Mining to Knowledge Discovery in Databases* [7]

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 [7].

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 [18]. 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 [23], 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

[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. [24]

systems, more convenient shopping - but at the high cost of losing our privacy.

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.

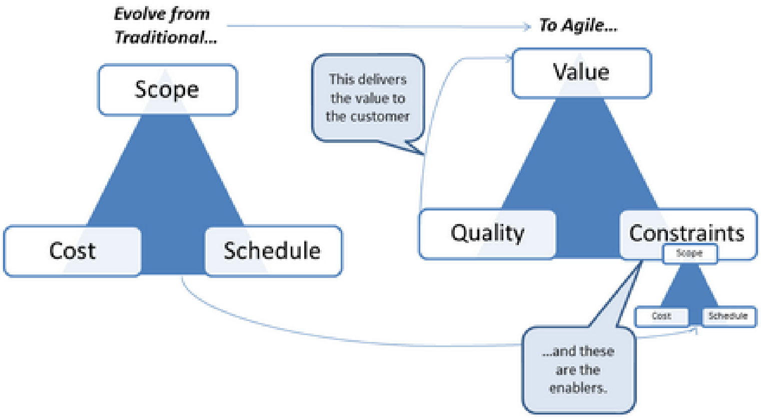

Evolve from Traditional...
To Agile...
Scope
Cost
Schedule
This delivers the value to the customer
Value
Quality
Constraints
Scope
Cost
Schedule
...and these are the enablers.

Figure 2: Traditional to Agile project management evolution. Source: [12]

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* [6].

[5] Within the MOA context, *filters* are procedures applied to data prior to their analysis using machine learning algorithms.

    a) Noise addition [5, p. 54]
    b) Multiplicative noise [5, p. 57]
    c) Microaggregation [5, p. 60]
    d) Rank shuffling [5, p. 73]
    e) Differential privacy [4]

- **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 Project phases

The project is divided in 4 main phases, besides of the undertaking of the Project's Management module. Each phase has an estimated duration and a risk evaluation in terms of schedule deviation. The amount of hours is an approximated calculation from the number of days in each phase: 4 hours a day are estimated to be spent, because I am currently working part-time and also taking some subjects. A more detailed task granularity can be seen in the Gantt chart. Task dependencies are shown in the chart too. Those phases, chronologically ordered are:

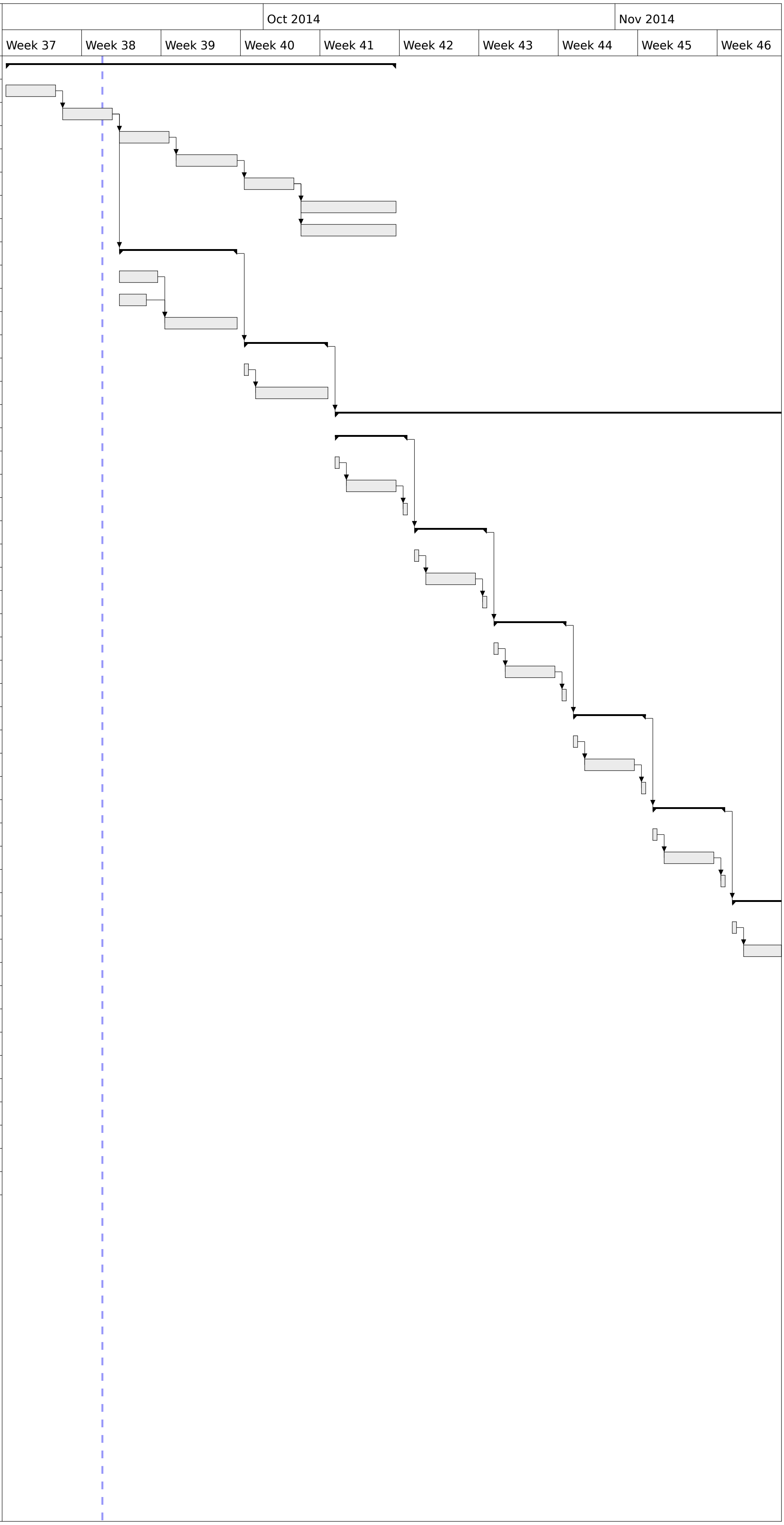
1. **Contextualization**: it is intended to perform a deeper bibliographic research and a study of the main subjects concerning the project, at the theory level - no practical skills or technological research will be done.
    - **Duration estimation:** 11 days (44 hours).
    - **Risk:** this phase has a medium to high risk of being delayed, due to lack of effective time (a wrong estimation), and also because more insight than planned might be needed, consuming more time.
2. **Environment setup**: during this phase, all necessary tools and material resources will be gathered and configured. The concrete developing workflow will be decided, too.
    - **Duration estimation:** 8 days (32 hours).
    - **Risk:** this phase has a low risk of being delayed, because the technology that is to be used is, a priori, well known to us.

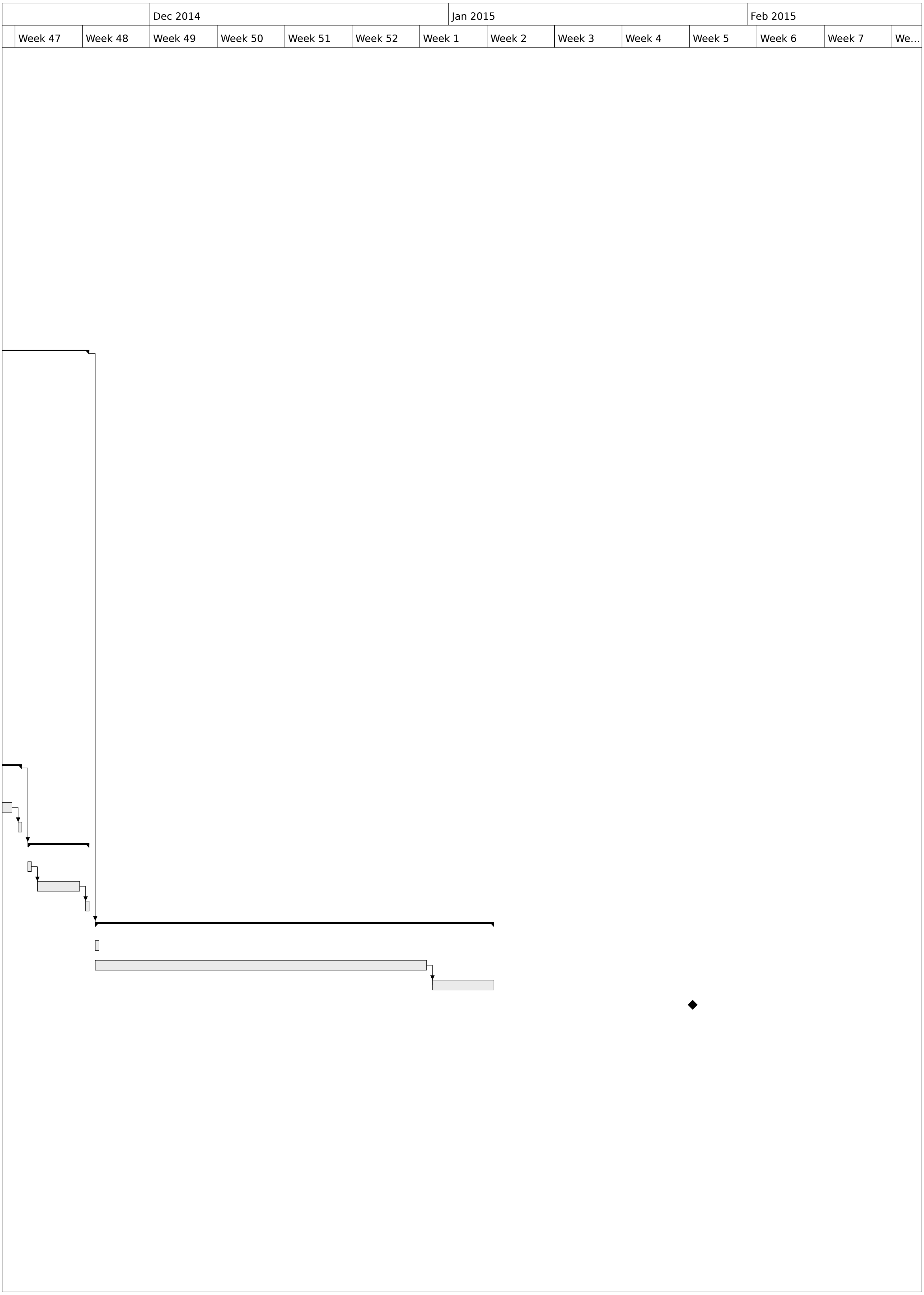
3. **Development**: all of this project coding will be performed during this phase. As said before, a sprint methodology will be used during this phase, being one week each.
    - **Duration estimation:** with an initial planning of 7 sprints, 49 days will be used (196 hours).
    - **Risk:** there is a medium risk of this phase to be delayed. Even with the use of Agile methodologies, if a fundamental feature was needed and there was no more time left, another sprint (or at most a couple of them) could be introduced, to finish the remaining tasks.
4. **Documentation**: the project's report will be written after the development phase, along with any deployment documentation that was required and the final presentation, which will also be rehearsed then.
    - **Duration estimation:** 42 days (168 hours).
    - **Risk:** this phase has a medium risk of being delayed too. Reviews of the report will be made and writing in English might take up more time than expected.

2.2.5 Detailed schedule: Gantt chart

The following chart was generated with the Project management free software package, available online on the Ubuntu 12.04 Software Center. Please note that there is no way the chart could fit in a single page (not even if it was landscape).

Name	Work
GEP module	**44d**
Scope definition	5d
Project scheduling	5d
Budget management & feasibility	5d
Preliminar presentation	6d
State of the art & literature	5d
Specifications	9d
Final presentation & document	9d
Contextualization	**14d**
IT & Privacy	4d
Data mining	3d
PPDM	7d
Environment setup	**8d**
Workflow design	1d
MOA framework	7d
Development	**49d**
Sprint 1	**7d**
Planning	1d
Implementation	5d
Integration	1d
Sprint 2	**7d**
Planning	1d
Implementation	5d
Integration	1d
Sprint 3	**7d**
Planning	1d
Implementation	5d
Integration	1d
Sprint 4	**7d**
Planning	1d
Implementation	5d
Integration	1d
Sprint 5	**7d**
Planning	1d
Implementation	5d
Integration	1d
Sprint 6	**7d**
Planning	1d
Implementation	5d
Integration	1d
Sprint 7	**7d**
Planning	1d
Implementation	5d
Integration	1d
Documentation	**43d**
Project deployment & finalization	1d
Project report	35d
Presentation & rehearsal	7d
Project defense	




Dec 2014
Jan 2015
Feb 2015
Week 47
Week 48
Week 49
Week 50
Week 51
Week 52
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
We…

2.3 Resources

Resources consumed in this project only fall in one of the following categories: *human resources*, *hardware*, *software* and *other expenses*. For a detailed description of what will be needed in the project, please see the following section, on Budget management. It is important to keep in mind that *all* resources will be consumed equally througout the entire project duration.

2.4 Budget

The project's budget is entirely based on an estimation of human, hardware and software resources costs. No real income is perceived, besides the salary of the project's supervisor, who is a tenure-track lecturer at the Barcelona School of Informatics, and an associate researcher at the Barcelona Supercomputing Center. No third parties are involved in the project - no companies or organizations are providing any funds. Moreover, even though the work is to be integrated into the MOA framework, it is indeed an open-source project, to which we will be contributing, meaning contributions are expected from any kind of source, be it funded or not.

All other associated costs are *externalized*, either by people involved in the project or by the university, where the development of the project will be held.

2.4.1 Budget estimation

The total cost of the project is derived from the sum of the following items:

- **Human resources:** (summarized in table 1)
  All expenses included here are related to people's salaries. Only one developer will be working on this project, but a number of hours involving supervision tasks is also imputed to the project's supervisor, so its corresponding cost is added too. Taxes are included in all of the following items. The price is also an estimation: on the developer's side, it is based on a salaries comparison webpage (*Glassdoor* [13])[6]; on the supervisor side, the price is based on his own estimation.
  - *Developer:* an average of 20 hours a week are estimated, spanning for about 21 weeks, summing up a total of 420 hours.
  - *Supervisor:*
    - Project's take off: 8 hours, between meetings and initial planning.
    - Sprints: 8 hours each sprint, taking into account both face to face meetings and other supervising tasks. There are 7 sprints scheduled so far, making a total of 56 hours.

[6] As of date 12th October, 2014, the average salary for a software engineer in Barcelona is 32000€ per year (including taxes). Considering 12 monthly instalments and an average of 160 hours per month, this yields a total of 16.66€ per hour.

  * Documentation: during the project's final stage, an estimation of 20 hours is taken from the corresponding supervision of the project's report.

Role	**Price (per hour)**	**Working hours**	**Total**
Supervisor	35€	84	2940€
Developer	16.66€	420	6997.2€
		Total	**9937.2€**

Table 1: Human resources associated costs. All taxes are included in the Price per hour column.

- **Hardware:** (summarized in table 2)
  All hardware needed resources are shown in the corresponding table. Their cost is calculated by estimating its amortization, spanned over 5 years (it is a personal laptop). To calculate its amortized cost per hour, we will take into account that this equipment is used throughout the course too, and estimating that 2500 hours of work are carried each year.

Product	**Price**	**Units**	**Amortized price per hour**	**Work time (hours)**	**Total**
Asus k53sv	650€	1	0.052€	420	21.84€
				Total	**21.84€**

Table 2: Hardware amortization costs. All taxes included.

- **Software:**
  All software needed to undertake this project is free and, most of it, is open sourced. Despite this, we will include a list of it here, to show what will be used at a finer grain.
  - *Ubuntu 12.04*: operating system. Available at: `http://www.ubuntu.com/download`.
  - *Trello*: online task management tool. Available at: `https://trello.com/`.
  - *Google Drive*: online, collaborative office software suit, used to create burndown charts (spreadsheets). Available at: `https://drive.google.com`.
  - *Java SDK*: Java language Software Development Kit. Available at: `http://openjdk.java.net`.
  - *Eclipse IDE*: integrated development environment package. Available at: `https://www.eclipse.org/home/index.php`.
  - *Git*: source version control system. Available at: `http://git-scm.com/`. Remote code repositories will be hosted at GitHub (`https://github.com`) for free.

  - *MOA*: Massive Online Analysis, a stream mining framework. Available at: `http://moa.cms.waikato.ac.nz`.
  - LaTeX: document preparation system. Available at: `http://www.latex-project.org`.

- **Other expenses:**
  All expenses not covered in the previous sections are detailed in table 3.

Product	**Price per month**	**Months**	**Total**
Energy	35€	4	140€
Water	25€	4	100€
Heat & air	30€	4	120€
Internet connection	40€	4	160€
		Total	**520€**

Table 3: Uncategorized resources estimated costs. All taxes are included.

Please note that the cost of each item of this section is an estimation. Moreover, even though they are displayed, since no budget is really available, they will be *absorbed* by the university, where most of the work will be carried out.

2.4.2 Total budget estimation

The sum of the subtotals of the previous sections is shown in table 4. Please note that, since taxes are already included in each item appropriately, there is no need to add them here.

Concept	**Total**
Human resources	9937.2€
Hardware	21.84€
Software	0€
Other expenses	520€
Total	**10479.04**

Table 4: Total budget: summation of budget estimations.

All costs are just estimations and are not covered in any way, with the exception of the supervisor's salary. This means that, in fact, there is no possible way this project is feasible. However, given that the developer has no salary at all and that all other extra costs are assumed by the university or the developer, the project can be developed normally.

2.4.3 Budget control mechanisms

Any budget deviations related to material equipment or software purchases will be monitored in the sprint planning meetings at the beginning of each of those phases during the project. These possible extra costs will be assumed by the developer, since no other source of funds is available.

Another source of budget deviations can be found on the project's duration. If the schedule is not fulfilled and the project is delayed, extra cost in terms of human resources, hardware amortizations and other expenses would have to be added. They still would be treated as they are in the present analysis, meaning no significant change would occur.

3 Context

This project is framed in the broad field of *data mining*, of which we already gave a brief introduction in section 1. We will extend that introduction to provide further understanding of the main topics and concepts of the project's environment.

3.1 Data mining

With the continuous increase of computing power, due to the recent advances in software and hardware, the machine learning field, more commonly known as data mining, has arisen. We have already seen that the concept of data mining is broad and covers many aspects of the knowledge discovery process.

We have also reviewed what the scale limits are to classic (batch data processing) data mining techniques. Particularly, it is a matter of the amount of information - the size of the datasets - to be analyzed, but also a matter of the rate of this data acquisition. In this respect, stream data mining is the "subfield" that has grown to give an answer to these technical challenges. A more deeper review of this research area is given in the State of the art section (3.4).

3.2 Privacy

Privacy is a concept that can be defined as the ability of an individual or group to seclude[7] themselves, or information about themselves, and thereby express themselves selectively. It is understood differently depending on the social and cultural background of individuals, but it is, in fact, recognised as one of the most fundamental rights of the human nature. Indeed, the Universal Declaration of Human Rights' 12th article [1] states that:

> No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.

This such right has, however, been violated and breached continuously ever since information exchange and advanced communication technologies have been developed and its use increased. It did not began with the spread of the Internet, but its adoption has greatly magnified both the ability to breach people's privacy and the impact these breaches have.

3.2.1 Privacy leaks consequences

We are fully aware of the dramatic consequences that information leaks have caused these past years: the PlayStation Network outage which left millions of video gamers without

[7] "Seclusion is the act of placing or keeping someone away from other people." Source: Merriam-Webster online dictionary [14]

access to that service [19] or the more recent celebrity intimate photographs leaks [3], just to give a couple of examples. Even though they are referred to as IT security breaches, they are, in fact, privacy rights violations, because sensitive personal information was compromised in either situations.

Not only rights are breached when privacy is violated. Economical and social consequences are at stake too. For example. identity thefts are performed on a daily basis, thanks to the vast amount of available data about individuals one could easily gather in the web. These theft was estimated to have a cost in the order of billions of dollars, back in 2005 [20]. Privacy is, therefore, a very serious question we have to bear in mind.

3.2.2 Legal framework

Efforts are being carried out to develop legal frameworks to help protect people's privacy, at many levels. One such example is the spanish LOPD[8] [2], a law that aims, among other things, to define different data privacy levels and mandatory proceedings associated to each - no matter the medium used to transfer it or store it.

There are some pitfalls to these legislative efforts, though. Firstly, it is really hard to assess their accomplishment in the IT sector and, thus, it is sometimes a matter of confidence in the developer's good practice. Another important drawback is that online services, such as social networks, can be accessed globally, but, on the other hand, their legislative framework is that of the country to which the backing company offering the service belongs to - jurisdiction definition in the Internet is still a matter of intense debate, nowadays.

3.2.3 Privacy and data mining

Within the field of data mining, sensitive data must be treated accordingly, and that involves not only good IT security practices, but a responsible treatment when research results are published.

Statistical Disclosure Control (SDC) is the name that the statistical community has given to what the data mining community calls Privacy-Preserving Data Mining (PPDM). This field, whatever its preferred name is, deals with controlling the that information about specific individuals is not extracted from statistical summary results. Also, if full datasets are to be released, PPDM practices should be applied to data in order to preserve privacy, whilst maintaining the statistical significance of it, i. e., the amount of information - knowledge - that this data can provide.

Further details of SDC methods and approaches can be found on the State of the art section (section 3.4).

[8] LOPD stands for *Ley Orgánica de Protección de Datos*, a law that was approved by the spanish courts in 1999. It has been modified several times, being the law enforcement regulation approved in 2007.

3.3 Involved stakeholders

A good way to investigate the motivation driving this project is to examine which third parties are (indirectly, perhaps) involved in this project, giving a general overview of their different possible interests in this work.

3.3.1 Users: organizations and developers

The most direct users of the results of this project will be developers - data scientists[9] -, researchers and the companies and organizations that employ them. Using privacy preserving filters, which they don't need to develop themselves, they are allowed to release their results or their data without compromising their user's privacy.

National statistical agencies[10] are often bound to use these methods, by law, but that might not be the case with private organizations. In the first place, information is the key to many companies' success, so they are prone to release it anyway, but there might be many organizations (research centres, academic groups, etc.) that find this possibility interesting or even needful.

Open Data community: the still young and poorly extended (and implemented) concept of *open data* can benefit too from the fact that releasing data is no longer a problem in terms of information security. The idea behind this expression is that data sets should not or need not be stored redundantly - if it was easily accessible (through web services, for example), one could exploit diverse data sources without having to gather it himself. Open data platforms exist, in fact, but their contents are poor, at the knowledge discovery level, and are difficult to integrate into data mining systems due to the lack of standard formats, for example.

If data sets are more easily anonymized using such privacy preserving methods, they are more likely to be published, aiding open data to fulfill its goals.

3.3.2 End users: data providers

When talking about *end users* here, we are referring to everyone that is allowing data be generated from them, gathered, collected, stored and analyzed. Nowadays, this is basically most of the population.

End users get the most important benefit from privacy preserving data mining: their data is secure against information disclosure attacks. Because all kinds of sensitive data are kept and analyzed, this turns to be the main motivation for the project. Even though

[9] The term *data scientist* is used to designate the evolution of data analysts or business analysts job titles in the world of *Big Data* or data mining. They are trained in computer science and applications, modeling, statistics, analytics and math.

[10] The ONS (Office for National Statistics, in the UK) team is using statistical disclosure control methods as part of their process methodology [8].

end users might not be aware of the specific technologies that keep their data safe, they still have the right to demand this protection.

3.4 State of the art

Let's cover now the latest advancements and discoveries in the different related fields that affect this project.

3.4.1 Stream mining

Data stream mining is a relatively new field. Even though its theoretical foundation is based in well-established statistical and computational approaches, it has not been until recent years that this research area has experimented a great growth in interest.

The main problem when dealing with streaming data is the high throughput of data being analyzed, under computational resources constraints. Variable data rates is another problem that has to be addressed too. Once these problems are resolved, efforts are done so the same kind of data mining analysis as in the case of batch data processing are available: classification, regression or clustering tasks, as well as outlier detection and recommendation systems. We will not cover these techniques here, because they are not related to this project, by themselves. Instead, we will have a look at some different stream mining solutions, because their working principles do affect the way the project's algorithms will be implemented.

Solutions provided in this field can be categorized into *data-based* and *task-based* ones [11], depending on their approach.

- **Data-based stream mining solutions:** The idea behind these solutions is to use a subset of the original dataset to perform the required analyses. Diverse techniques that have been used in this sense can further be split into two more categories:
  - *Sampling methods:* either by randomly picking samples of the data stream or by randomly selecting chunks (subsets) of the stream, sampling methods discard part of the incoming data, while performing the knowledge discovery processes with the sampled data. The main problem with this approach is that is hard to know when to pick a sample or which records should be stored, because there is no previous knowledge of the dataset size or its information structure.
  - *Summarizing methods:* they use aggregated data or calculated statistical measures (that are continuously recalculated) to provide the information needed for the data mining algorithms. In this case, it is the loss of information and accuracy and the inability to control data distribution fluctuations what renders these methods not so usable as it was desired.

- **Task-based stream mining solutions:** The solutions that fall into this category are based not on performing data transformations, but on changing the data mining methods to enable their use on data streams.
    - *Approximation algorithms:* these are a kind of algorithms that are designed to solve computationally hard problems, by giving an approximate result. Instead of computing exact solutions, they just guarantee a certain error bound. The problem with these methods is, again, the high received data throughput, which they cannot cope as well. Additional tooling is therefore needed if one wishes to use them.
    - *Sliding window method:* this method, a common pattern in many online[11] applications, maintains a *sliding window* in which the most recent data is kept. As data is received from the incoming streams, this window "advances" so new observations are kept inside. The data mining analyses are then performed using the data available inside the window and summarized versions of the older records, in the form of statistical measures or aggregated data. This particular method is the one that the MOA package uses - thus its name: Massive **Online** Analysis. This solution scheme enables dealing with concept drift, which would not be possible if just aggregated data was used.
    - *Algorithm output granularity:* this method is a resource-aware data analysis approach that can perform the local analysis on resource constrained devices, by adapting to resource availability and data stream rates - when resources are completely running out, the results are merged and stored.

Data stream mining software: because it is an incipient field, stream mining software packages are quite uncommon. Even though specific applications have been developed [15], MOA remains as one of the few generic[12], free and open sourced systems. In relation to MOA, a new project called SAMOA [25] is being developed too, based on top of MOA itself, and a couple of streaming processing engines: Apache S4 [9] and Apache Storm [10], developed by the Apache Software Foundation.

3.4.2 Privacy preserving stream mining

Many different methods have been developed to help prevent information disclosure when data mining datasets or results are released. These algorithms pursue the generation of results or data that have particular properties concerning privacy preservation.

Privacy preserved data properties: some of the desirable properties of privacy-protected data are:

[11] In computer science, an *online algorithm* is one that can process its input piece-by-piece in a serial fashion, i.e., in the order that the input is fed to the algorithm, without having the entire input available from the start.

[12] MOA is not focused towards any particular application scenario: it is a base tool with which we can build such specific systems.

- First described in 2002, by Latanya Sweeney, a release of data is said to have the *k-anonymity* property if the information for each person contained in the release cannot be distinguished from at least $k - 1$ individuals whose information also appears in the release [21].
- The evolution of the concept of k-anonymity is *l-diversity* and adds further privacy preservation by adding intra-group diversity, so to avoid the flaws of the k-anonymity privacy model [16].
- Further on, the *t-closeness* property definition adds attribute-based privacy enforcement to the l-diversity model: to better preserve privacy, all values (all observations) from a particular attribute must not be too much different - instead, they should be close up to a certain threshold [17]. This is needed to preserve the privacy of those records that are more easily identifiable because their attribute values are more distinguishable.

Privacy preserving algorithms: the algorithms being used nowadays to achieve effective privacy preserving properties to datasets can be categorized into the following groups [5]:

- *Non-perturbative data masking:* these kind of methods do not perform data values transformations. Instead, they are based in partial suppressions of records or reductions of detail of the datasets. Some examples[13] are:
  - Sampling
  - Global recoding
  - Top and bottom coding
  - Local suppression
- *Perturbative data masking:* these methods do release the whole dataset, if required, but it is perturbed, this is, values are changed by adding them noise. This way, records are diffused and reidentifying individuals is harder. Some examples are:
  - Noise masking
  - Micro-aggregation
  - Rank swapping
  - Data shuffling
  - Rounding
  - Re-sampling
  - PRAM
  - MASSC

[13] We can't cover every algorithm in detail, because some of them are not relevant and because those which are to be implemented in this project will be described in detail in the final project report.

Privacy preserving *Stream* Mining: many of the previously listed methods are already implemented in many classical data mining frameworks and software systems; for example, the `sdcMicro` package for the **R** statistical package [22]. However, privacy preserving methods are still not widespread in the stream mining ecosystem - that is another motivation for this project.

3.5 Environmental impact

No relevant direct environmental impact is related to this project, neither tied to its development nor its further deployment. No use of massive resources is done and the results of the work will not, presumably, result in a significant environmental change of any kind.

It is still true, however, that data mining, as a discipline and its broad use, does consume a lot of resources, in terms of technological infrastructure and energy. We cannot forget that collecting, storing and processing data at the industry scale needs entire data centers fully dedicated to the data mining process. Power consumption is a big concern with nowadays information technology, as it is the huge amount of rare materials that electronic devices contain. These are derived or indirect effects of the data mining process. This issue deserves to be examined more closely, in the project's final report.

3.6 Social impact

Concerning social impact, this project's strength is related to users privacy preservation through the implementation of algorithms that help anonymize their data within the data mining process. This topic will be explained more extensively in the project's final report, but it is of paramount importance nowadays. Privacy is being relegated to the background with the advent of new information technologies, devices or platforms such as social networks or banking applications. Data is gathered from everyone and there is an increasing need of methods to protect it. Together with other fields more focused in securing the access to data, privacy-preserving data mining is designed to keep data safe at the analysis stage of the data mining process.

Not only ethical concerns are addressed by protecting the users' privacy, but economical issues too. Industrial-scale information theft has a huge impact on enterprise economies, because of distrust and because disclosed sensitive data can be used to make profit of it.

References

[1] UN General Assembly. Universal declaration of human rights, December 1948.

[2] BOE. Ley orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal, 1999. URL: `https://www.boe.es/boe/dias/1999/12/14/pdfs/A43088-43099.pdf`.

[3] Arthur Charles. Naked celebrity hack: security experts focus on icloud backup theory, September 2014. URL: `http://www.theguardian.com/technology/2014/sep/01/naked-celebrity-hack-icloud-backup-jennifer-lawrence`.

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

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

[6] Martin Fowler et. al. Agile manifesto for software development, October 2014. URL: `http://agilemanifesto.org`.

[7] 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. URL: `http://dl.acm.org/citation.cfm?id=257938.257942`.

[8] Office for National Statistics. Statistical disclosure control, 2014. URL: `http://www.ons.gov.uk/ons/guide-method/method-quality/general-methodology/statistical-disclosure-control/index.html`.

[9] The Apache Software Foundation. S4: Distributed stream computing platform, 2014. URL: `http://incubator.apache.org/s4`.

[10] The Apache Software Foundation. Storm, distributed and fault-tolerant realtime computation, 2014. URL: `https://storm.incubator.apache.org`.

[11] Mohamed Gaber. Mining data streams: a review. *ACM SIGMOD Record*, 34, June 2005.

[12] Shane Hastie. Agile australia - opening keynotes, October 2010. URL: `http://www.infoq.com/news/2010/10/agile-australia-keynotes`.

[13] Glassdoor Inc. Company salaries - glassdoor, October 2014. URL: `http://www.glassdoor.com/Salaries/index.htm`.

[14] Merriam-Webster Inc. Seclusion, October 2014. URL: `http://www.merriam-webster.com/dictionary/seclusion`.

[15] H. Kargupta. Minefleet®: The vehicle data stream mining system for ubiquitous environments. *Ubiquitous Knowledge Discovery*, 2010.

[16] Ashwin Machanavajjhala. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 2007.

[17] Li Ninghui. t-closeness: Privacy beyond k-anonymity and l-diversity. *Proc. of IEEE 23rd Int'l Conf. on Data Engineering*, 2007.

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

[19] Shane Richmond. Millions of internet users hit by massive sony playstation data theft, April 2011. URL: `http://www.telegraph.co.uk/technology/news/8475728/Millions-of-internet-users-hit-by-massive-Sony-PlayStation-data-theft.html`.

[20] Sasha Romanosky. Do data breach disclosure laws reduce identity theft? In *Workshop on the Economics of Information Security*, 2008.

[21] Latanya Sweeney. K-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002.

[22] Matthias Templ. *sdcMicro: Statistical Disclosure Control methods for anonymization of microdata and risk estimation.* CRAN R package repository. URL: `http://cran.r-project.org/web/packages/sdcMicro/index.html`.

[23] New Zealand University of Waikato. Moa - overview, October 2014. URL: `http://moa.cms.waikato.ac.nz/overview`.

[24] New Zealand University of Waikato. Weka 3 - data mining with open source machine learning software in java, October 2014. URL: `http://www.cs.waikato.ac.nz/ml/weka`.

[25] Yahoo. Samoa by yahoo, 2014. URL: `http://samoa-project.net/`.