UNIVERSITAT POLITÈCNICA DE CATALUNYA

FACULTAT D'INFORMÀTICA DE BARCELONA

GEP — Deliverable 2

GEP tutor: Andujar Larios

Analysis of the SVM-RFE algorithm for feature selection

Author:
Robert PLANAS

Director:
Luis A. BELANCHE

Bachelor Degree in Informatics Engineering
Specialization: Computing

March 11, 2021

Contents

Chapter 1

Introduction

This bachelor thesis of the Computer Engineering Degree, specialization in Computing, has been done in the Facultat d'Informàtica de Barcelona of the Universitat Politècnica de Catalunya and directed by Luis Antonio Belanche Muñoz, doctorate in Computer Science.

1.1 Context

In statistics, machine-learning, data-mining, and other related disciplines, it is often the case that there is redundant or irrelevant data in a dataset[1]. Indeed, before we can start working with the data, some form of data analysis and cleaning is required. Data cleaning may include removing duplicated rows or rows with missing values, removing observations that are clearly out-layers, removing irrelevant variables (e.g. name, surname, email address), etc.

With the new era of Big Data, datasets have increased in size, both in number of observations and in dimensions. Applying classical data-mining and machine-learning algorithms to this high-dimensional data rises multiple issues collectively known as "the curse of dimensionality". One such issue is the elevated, usually intractable, cost and memory requirements derived from the non-linear (on number of dimensions) complexities of the algorithms. Another issue has to do with data in a high-dimensional space becoming sparse and negatively affecting the performance of algorithms designed to work in a low-dimensional space. And finally, a third issue is that with a high number of dimensions the algorithms tend to overfit, that is, they don't generalize enough and end up producing models that perform worse with real data that their predicted performance with the training data. (Li et al., 2017)

Simple manual data cleaning is not enough to achieve satisfactory amounts of dimensionality reduction. In this case we can use automatic techniques. We can classify such techniques in two categories, feature extraction, and feature selection.

1.1.1 Feature Extraction

Feature extraction techniques transform the original high-dimensional space into a new low-dimensional space by extracting or deriving information from the original features. The premise is to compress the data in order to pack the same information at the expense of model explainability[2]. Continuing with our data compression analogy, virtually all feature extraction techniques perform lossy compression, that

[1] A table with rows / records / observations and columns / variables / features / dimensions / predictors / attributes.

[2] The ability to explain why certain predictions are made. Also, interpretability.

is, some data is lost which makes the process irreversible. Notice that, if the process was reversible then feature extraction would not decrement explainability.

Some well known feature extraction algorithms include Principal Component Analysis (PCA) and auto-encoders, the first being a linear transformation over the feature-space and the second a neuronal network. PCA may be extended with a kernel method in order to make non-linear transformations. Similarly, we will also use kernels methods to extend SVM-RFE.

1.1.2 Feature Selection

In contrast, feature selection only selects a subset of the existing features, ideally the most relevant or useful. This may imply a greater loss of information compared to feature extraction, but it doesn't reduce explainability. Some problems require feature selection explicitly. In domains such as genetic analysis and text mining, feature selection is not necessarily used to build predictors. For example in micro-array analysis feature selection is used to identify genes (i.e. features) that discriminate between healthy and disease patients.

Feature selection methods may be classified by how they are constructed in three categories:

- **Filter:** A *feature ranking* criteria is used to sort the features in order of relevance, then select the k-most relevant.
- **Wrapper:** They use a learning machine (treated as a black box) to train and test the dataset with different subsets of variables. They rank the subsets based on the performance (score) of the model. A wide range of learning machines and search strategies can be used. Within the greedy strategies we find *forward selection* and *backward elimination*.
- **Embedded:** Like wrapper methods but more efficient. They use information from the trained model itself to make feature selection. Because they don't use the score, they can also skip testing the model.

In both wrapper and embedded methods greedy strategies can be used. **SVM-RFE** is a feature selection algorithm, of the embedded class, that uses Support Vector Machines (SVM) and a greedy strategy called Recursive Feature Elimination (RFE). It takes advantage of the fact that, for linear SVM, the variable with the smallest weight in each iteration is the one that needs to be eliminated. (Guyon and Elisseeff, 2003)

1.2 State of the art

The SVM-RFE algorithm was first proposed in a paper on the topic of cancer classification (Guyon et al., 2002).

Stuff about how SVM-RFE has been used on cancer classification. Stuff about known and natural extensions of SVM-RFE. Maybe other interesting applications of SVM-RFE.

Justification, Identify the problem

1.3 Objective

The main objective of this project is to research variants of the SVM-RFE algorithm and try to optimize it. Optimizations may be in the form of improved performance, a

reduction in time utilization or fewer memory required. We've classified the possible optimizations by the following categories:

- Optimizations that try to use the SVM more effectively. This includes exploring non-linear kernels and other values for the regularization parameter.
- Extensions to the RFE strategy. This may include reusing more information from previous iterations, sampling, etc.

1.3.1 Objective break down

To accomplish this objective, the project has been subdivided in two parts:

Theoretical Part

- Do research in SVM-RFE and in the methods that will be tackled in the project.
- For each method:
  - Design the algorithm and write its formalization in pseudocode.
  - Define the expected advantages or disadvantages of this method over the base SVM-RFE.
  - Compute the space and time complexities.

Practical Part

- Program the methods studied and the SVM-RFE extensions.
- For each method:
  - Analyze its behavior for artificial data sets.
  - Analyze its behavior for real-world data sets.
- Compare the results obtained with the ones computed in the theoretical part.
- Draw conclusions about all the results obtained in the project.

1.3.2 Stakeholders

This project is intended to be of use for many involved parties. The most directly involved group, is the tutor and the researcher. Luis Antonio Belanche Muñoz is the tutor of this project. Robert Planas Jimenez would be the researcher. Feature selection algorithms is one of the areas of research of the tutor, and he has wanted to explore extensions to the SVM-RFE algorithm. Thus, he will lead and guide the researcher for the correct development of the project. The researcher is responsible for planning, developing and documenting the project, as well as experimenting, analyzing and drawing conclusions.

The other group of interested parties would be stakeholders that do not interact with the project directly but still benefit from it. In the first place we have researchers on the fields of bioinformatics and data mining, that use machine learning methods (specifically, SVM-RFE) for analyzing micro-array analysis, text analysis, or other of its popular applications. Indirectly companies that make use of their findings will also benefit, and finally the general population may also benefit from better diagnostics and more effective drugs.

1.3.3 Potential obstacles and risks

Some of the obstacles and risks identified that could potentially prevent the correct execution of the project are:

- **Deadline of the project:** There is a deadline for the delivery of the project. This being a research project however, is considerably hard to estimate how much time tasks will take, or even decide whether a task has been finished or not.
- **Bugs on some libraries:** This is considered of low risk, but is still a possibility that errors on the software package used extend to code, making it work incorrectly.
- **Insufficient computational power:** Machine learning algorithms in general can be very resource intensive. It could be the case that our hardware can not handle some datasets.
- **Hardware related issues:** A hard drive failure could occur that would end in lost data, or a failure in a router could disconnect us from the internet.
- **Health related issues:** In addition to health issues that can occur at any time without prior notice, we're in the middle of a pandemic.

1.4 Methodology

1.4.1 Framework

The methodology that I will use for the project is a combination of Waterfall and Kanban methodologies. Waterfall will be used to define the general phases of the project, and Kanban for tracking the individual tasks. In Waterfall tasks can not start until the previous task has been completed, and thus following strict deadlines is important. Phases will be managed like this, one phase will not start until the previous phase ends. Each phase will be composed of multiple tasks, which will then be managed by a different methodology. That will be Kanban.

Kanban is much more flexible than Waterfall, its principal objective is to manage tasks in a general way, by assigning different status to them. Kanban stands out by its simplicity, and we will continue with that simplicity by managing the visual representation of the cards in a simple plain text file, with tabulation separated columns. Each card will be in a row, with the first column defining its name and the other its status. The status I've considered are:

- **To do:** The task has been defined, but I have not started to work on it yet.
- **In progress:** The task has started and some progress has been made.
- **On hold:** The development of the task has been paused for some unforeseen reason.
- **Completed:** The task is finished.

1.4.2 Validation

We will use a GitHub repository as a tool for version control, which will allows us to share code easily and the recover from data lose. The repository will contain both the code for the different experiments, each in one subfolder, and code for the documentation. In order to verify the implemented code, it will be passed through various tests to see whether the code works as expected. In the practical part, hyper-parameters will be selected using some model validation technique, such as the cross-validation. Each experiment will be done 3 times and the average of the results will be the final result. Lastly, face-to-face meetings will be scheduled once every two weeks with the tutor of the project. In these meetings it will be discussed the project status and the tasks to do during the following two weeks, before the next meeting. In case of problems in the project, extraordinary meetings can be arranged.

Chapter 2

Project Planning

This thesis is worth 15 ECTS credits, each of which with an estimate cost of 25 to 30 hours. Therefore, the total time allocated for this project, as indicated by the faculty, is of 375 to 450 hours. This time is to be distributed in 100 days, from 03/08/21 to 06/15/21, with an estimated work of 4 to 5 hours a day. The date of the oral defense is planned for the first week of July, this sets the deadline to be on the 06/18/21.

An extra 3 ECTS credits are to be used for project managment, this is roughly 80 hours, which makes the total time estimate for the whole project (Theisis + Project Managment) to be at best 450 hours and at worse 540 hours. In order to make a proper plannification, we have defined the estimated cost to be at 500 hours.

2.1 Task definition

In this section it is presented all the tasks that will be carried out along the project. For each, a description, duration and a list of dependencies with other tasks are given.

Project management is a mandatory group of such tasks, albeit not very useful, considering that: One, this project is done by a single individual, with assistance from a project director; And two, this is a research project, which makes planning of specific tasks difficult, since it is the results from the research that drive the next steps to be done.

- **Context and scope:** We have to indicate the general objective(s) of the project, contextualize it and justify the reason for selecting this subject area.
- **Project planning:** This will help us not lose focus while we're working on the project.
- **Budget and sustainability:** For this specific project, this is irrelevant. The budget required is negligible and already defrayed; and the impact, beyond trivial matters, is likely zero and otherwise unknown.
- **Final project definition:** Review the work done in the project management tasks.
- **Meetings:** Online meetings are scheduled once every two weeks with the tutor of the project. Discussion of the status and next tasks to do will be done.

This project is research focused. Therefore, before starting the practical tasks research on the various topics needs to be done. This will involve collecting and analyzing previous studies that tried new methods and extensions to the SVM-RFE algorithm. We will also have to document ourselves in the SVM and feature selection

areas, as well as the algorithms and the statistical theory used in the studies. Some basic understanding of bioinformatics may also be required considering the use case of most if these studies.

- **Research** previous work on the literature on extensions of the algorithm and create a short **report** with the findings.
- Write the algorithm formalization in pseudocode.
- Define the expected advantages or disadvantages of this method over the base SVM-RFE.
- Compute the space and time complexities.

Once the initial research is done the algorithm must be codified and tested. This group is composed of the following tasks:

- **Program the base SVM-RFE algorithm.** For this we will use a library for the SVM. For the RFE part, since we need to be able to extend the algorithm, we will program it from scratch.
- **Program the extensions of the SVM-RFE algorithm** based on the research done and the pseudocode.
- **Test the new extensions with artificial data.** This requires creating models and testing their performance with artificial data sets.
- **Obtain data sets with real data.** Multiple papers refer to publically available data-sets, we could use those and compare our results.
- **Test the new extensions with real data.** This requires creating models and testing their performance with real data sets.
- **Analyze the results** obtained in the experiments and draw conclusions. A report will be made for reference during the final documentation.

Once finished, the final documenting phase will begin. Firstly, we will collect all the information obtained in the experimental and analysis part, which will be available in the form of the reports that we've done along the tasks. Afterwards, we can start writing the documentation of the project. This will include a fairly extensive review on concepts related to SVM, statistics, feature selection and RFE. Finally, we will have to prepare for the oral defense of the project.

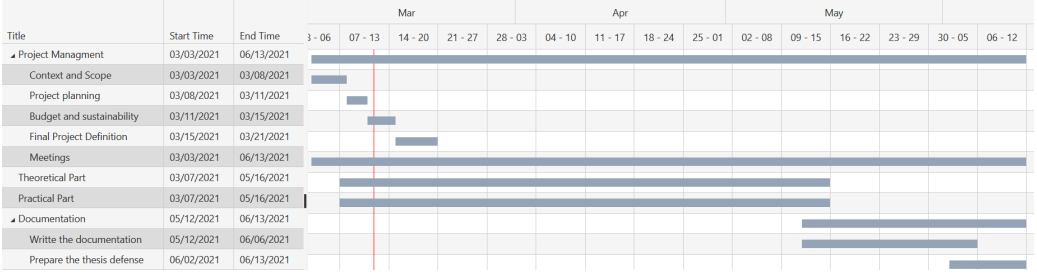

FIGURE 2.1: A summary of the tasks represented with a gantt chart. Notice that all the theoretical and practical tasks are done in parallel.

ID	Description	Hours	Dependencies
T1	Project Managment	80	
T1.1	Context and Scope	20	T2.1
T1.2	Project planning	10	
T1.3	Budget and sustainability	10	T1.2
T1.4	Final project definition	20	T1.1, T1.2, T1.3
T1.5	Meetings	20	
T2	Theoretical Part	160	
T2.1	Research	90	
T2.2	Formalize	20	T2.1
T2.3	Analyze	50	T2.2
T3	Practical Part	160	
T3.1	Program the base SVM-RFE algorithm	10	T2.1
T3.2	Program the extensions	50	T2.1, T3.1
T3.3	Test with artificial data	20	T3.2
T3.4	Test with real data	30	T3.2
T3.5	Analyze the results	50	T3.3, T3.4
T4	Documentation	100	
T4.1	Writte the documentation	80	T2, T3
T4.2	Prepare the thesis defense	20	T4.1

TABLE 2.1: Summary and time estimates of the tasks.

2.2 Resources

Our project needs resources to carry out its correct development. These resources have been divided in 4 different groups: human, hardware, software and material resources.

2.2.1 Human resources

There are three human resources that are directly involved in this project.

- **The researcher:** He is responsible for the development of the project, that is, he will have to plan, analyze, program, experiment and document the project.
- **The director/tutor:** He is responsible for leading and guiding the researcher for the correct development of the project.
- **The GEP tutor:** He is in charge of reviewing the project management tasks done in the initial stage of the project.

2.2.2 Hardware Resources

The most essential resource needed is a computer connected to the internet. In this project a personal computer will be used. Its specializations are 16 GB of RAM and a CPU *AMD Ryzen 7 4800HS*, with a base speed of 2.9 GHz and 8 cores. Hardware required for a connection to the internet (a router, an access point, etc) is also taken into account.

2.2.3 Software Resources

For project management tasks `Google Calendar` will be used. A number of other Google products such as `Google Mail`, `Google Drive` or `Google Meet` will also be used as tools required for communication with the director or storing of information.

The documentation will be written in LaTeX, a document preparation system often used in academia. It has the advantage to integrate well with version control systems. A LaTeX template named "`Masters/Doctoral Thesis`" will be used to facilitate the typesetting and styling of the document. This template was made by Vel and Johannes Böttcher and licensed under `LPPL 1.3`, based on previous work from Steve R. Gun and Sunil Patel and used with minor modifications. The original is available at `http://www.latextemplates.com/`. The document will be written with the `Microsft Visual Studio Code` editor, using the `LaTeX Workshop` extension, and it will be compiled with the `MiKTex` distribuition installed on a `Linux` machine running virtualized within a `WSL` container on top of the actual operating system, a `Microsoft Windows 10`. For browsing the internet the latest available version of the `Firefox` web browser will be used, and references to papers will be kept with the `Zotero` reference manager.

For the practical part, the programming language of choice will be `Python3`. This is currently one of the programming languages with better popularity in machine learning and related applications. Various libraries and software packages will be used for different tasks. For parsing datasets the `pandas` library will be used. For data visualization the `mathplotlib` library will be used. Also, a general tool-set designed for machine learning, the library `sklearn`, will be used. Finally, code will be documented in-place during its development with the `Jupyter Notebook` software.

Other libraries and software packages could also be used if the need arises.

2.3 Risk Management

The potential risks and obstacles have already been introduced in section 1.3.2. In this section we will focus on a contingency plan to mitigate the risks.

- **Deadline of the project:** The flexibility of the Kanban methodology should help us modify our schedule and working hours if required. If it becomes apparent that the deadline will not be meet, an extension of the deadline can be requested.
- **Bugs on some libraries:** Alternative libraries can be used. If no alternative is found, because most of the used libraries are open source, a *bugfix* could be implemented.
- **Insufficient computational power:** This can be mitigated by using a small sample of the data-set. This though, can induce a small performance reduction, and thus make the results not comparable with each other.
- **Hardward related issues:** To avoid it data loss, all code and documentation will be routinely uploaded to the cloud in a `GitHub` repository and `Google Drive` account.

Bibliography

Guyon, Isabelle and André Elisseeff (2003). "An Introduction to Variable and Feature Selection". In: *Journal of Machine Learning Research* 3.Mar, pp. 1157–1182. ISSN: 1533-7928.

Guyon, Isabelle et al. (Jan. 2002). "Gene Selection for Cancer Classification using Support Vector Machines". en. In: *Machine Learning* 46.1, pp. 389–422. ISSN: 1573-0565.

Li, Jundong et al. (Dec. 2017). "Feature Selection: A Data Perspective". In: *ACM Comput. Surv.* 50.6, 94:1–94:45. ISSN: 0360-0300. URL: https://doi.org/10.1145/3136625 (visited on 03/01/2021).