UNIVERSITAT POLITÈCNICA DE CATALUNYA

FACULTAT D'INFORMÀTICA DE BARCELONA

GEP — Deliverable 1

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 8, 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 several sub-objectives:

Theoretical Part

- Do research in SVM-RFE and in the methods that will be tackled in the project.
- For each method:
  - 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 Software required

The documentation will be written in LaTeX, a document preparation system often used in academia. It has the advantage to work well with version control. A LaTeX template named "`Masters/Doctoral Thesis`" will be used. 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/`.

For the practical part the programming language of choice will be Python 3, which is currently one of the most used for machine learning 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. This documentation may be integrated later, with some modifications, in the final thesis.

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

1.3.3 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.

1.3.4 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.

- **Slow hardware:** Machine learning algorithms in general can be very resource intensive. It could be the case that our hardware can not handle some datasets, but is an obstacle that is likely to be mitigated by using various methods.

- **Lost of data:** A hard drive failure could occur that would end in lost data. To avoid it, all code and documentation is routinely uploaded to the cloud in a GitHub repository.

- **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.

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).