Using Random Fourier Features with Random Forest

Albert Ribes

December 14, 2018

Contents

1 Context

1.1 General Framework

Machine Learning uses statistical models to make predictions or decisions when there is not known formula of feasible procedure to find the correct answer. In the supervised learning sub-field, it uses a collection of data instances and their corresponding answer to predict the correct outcome for new unseen data. This is known as learning.

The theory behind this process has been developed for many time and is quite old. But it has not been possible to use it in real world problems until the recent years, when the computational power of the machines has grown so much that it is able to perform most of the calculations needed to "learn" with a decent level of accuracy.

Moreover, nowadays it is relatively easy to find or produce huge datasets about almost any field, and that makes it possible for many businesses to learn from this data and to make better decisions.

Nevertheless, the current level of learning is not enough. The error rate is still too high for some applications, and it is needed a lot of computation time to train a model. There is still much work to do in this field.

1.2 Into the specifics

The error made by an algorithm may be caused by inherent limitations of the data (such as random noise, lack of information, mistakes in data collections, etc.) or by limitations in the mathematical model behind the algorithm, such as having a lot of bias or variance. One way to reduce this last problem in to use some kind of ensemble of predictors.

Bagging is an ensemble technique which has shown very good results for many algorithms. It is known to substantially reduce the variance of a model when each of the estimators has very little correlation with the others. It achieves that by using the bootstrap process (a random sampling with replacement), but for most of the models this is not enough. This is why it is rarely used for any model but the Decision Tree. The unstability in this algorithm helps each of the estimators to have low correlation.

There exists a mapping to transform data from a given space to a different one which is called Random Fourier Features. It can approximate any shift-invariant kernel function with a randomized mapping, and has successfully been used to decrease the error rate of a Neural Network.

In this project I try to use this mapping to increase the accuracy that can be achieved by models using the bagging ensemble. It is expected that it will enable other models than Decision Tree to produce uncorrelated estimators and thus reduce the variance of the model.

1.3 State of the Art

It is hard to define the state of the art in Machine Learning since it still does not exists a certain algorithm able to solve any kind of problem. Nevertheless, there are some general methods used to achieve the highest scores.

Deep Learning has proven to give very good results when you feed the network with huge ammounts of data. It has been used to master some difficult games and for image processing, among others.

Support Vector Machine are also very used with the kernel trick .....

But what is really carring Machine Learning to another level is the posibility of building computation mega-clusters to train models in a way that some time ago would have required many years of training.

Nevertheless, we are still not proud of the levels of accuracy we can achieve, and a lot of research is being done trying to increase it.

1.4 Problem to solve

The problem is always the same: we want to achieve higher accuracy while solving our problems. The proposed solutions to try to solve this problem are:

- **Increase the computation time**: either by using very powerful supercomputers or by using more time.
- **Use better datasets**: the larger and complete a dataset is, the better the accuracy will be.
- **Improve current algorithms or create new ones**: by making scientific studies

In this project I try to increase the accuracy with the third method. It has the advantage of being the cheaper in the long term and more affordable.

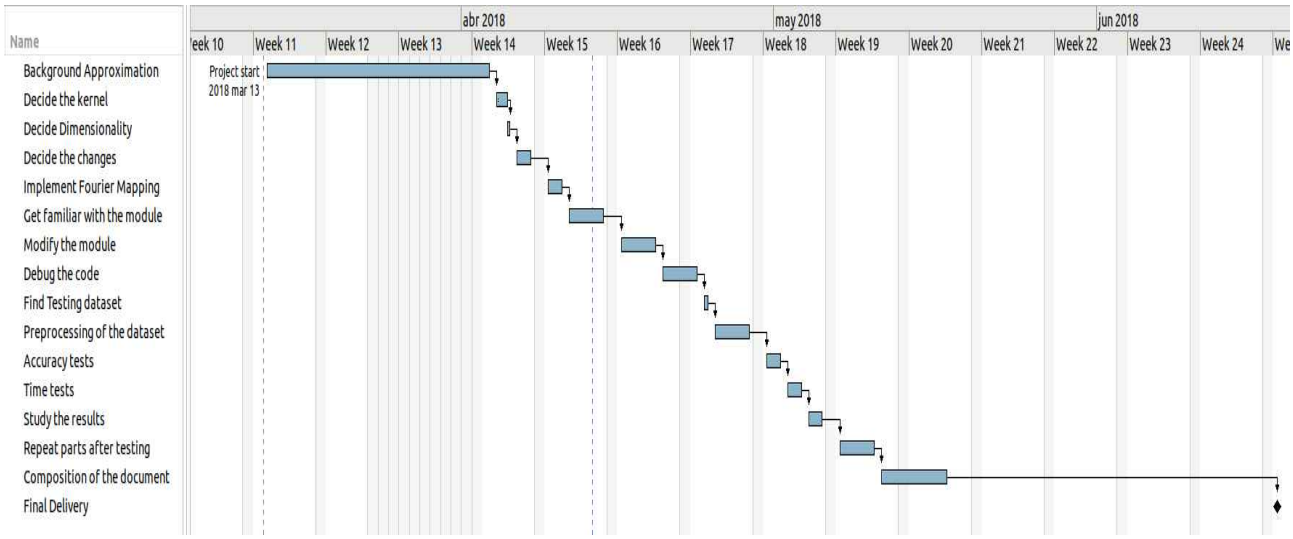
2 Planification

2.1 Original Planification

In the original planification of this project the scope was expected to be limited to the study of Random Fourier Features with the Random Forest algorithm. After checking that accuracy could not be increased to this model, it was changed to cover also other models, such as Logistic Regression and Support Vector Machine with Linear Kernel.

Furthermore, it was expected to deliver it on June 2018, and that was not possible.

The expected planning was:



2.2 Problems encountered with original planification

I encountered many problems with the original planification, both in the setting and in the execution. The problems were:

- **Very few knowledge of the study field**: in the begining of the project I didn't have a clear image of how the whole project would be. I lacked much of the knowledge that I would need in the project, and was not able to write a realistic planification.
- **Naive assumption of success**: I expected to find good results on the first experiments y performed and didn't have a plan for when they failed.
- **Programming time underestimated**
- **Meetings with the project director**: I didn't have a good planning on when to meet the director of the project, and time passed without advancing on the work.
- **Comunication problems**: for some time it was not possible to meet the director of the project physically, and the comunication was too slow.
- **Lack of initiative**: during the whole project I've been blindly following the path proposed by the director, not knowing where we were going. Thus, I needed constant feedback and was not able to advance without his advices.
- **Lack or rogour in following the planification**: the planification was not checked during the project.

Task	Time (h)
Write the project document	50
Test using DT with an ensemble without bootstrap	5
Write a demo with all experiments carried out showing that DT does not benefit from using RFF	15
Solve some problems that Nystroem is giving with some datasets	10
Some testing on why I find different behaviours with different datasets	20
Run experiments on usage of PCA and ordering	10
Debug details of graphical interface	10
Run experiments using Logistic Regression	10
Run experiments using a SVM with a Linear Kernel	10
(Optional) Find and use more datasets	10
Unexpected extras	20
Total	**170**

2.3 Proposed new planification

La fecha de entraga

As the contents of the project have changed from the original planification, It is needed to define a new set of tasks to finish the work. The ones which still need to be done are:

2.4 Current status in the new planification

3 Methodology

3.1 Original Proposed Methodology

The original proposed methodology was to define a set of tasks which should be done by the end of the project and a correct arrangement of them to assure all of them can be done.

3.2 Problems encountered with original Methodology

The problems encountered with the original methodology are similar to the ones in the planification.

For the one hand, by the time I defined the set of task I didn't have a clear idea of how the project would go. I lacked a lot of knowledge about the study case and planned the task assuming that I would obtain a results which I never got. Therefore, most of the tasks defined where useless and other other ones should have been there.

By the other hand, there was not a strict monitoring of the tasks that were done and the ones that where missing.

3.3 New methodoogy

I will continue to use the same methodology, but having fixed the problems that I had.

Now I have a clear understanding of the ins and outs of the project, so the new planning is expected to meet the expected requirements. Moreove, I do now follow the planning and meet the director of the project every week.

4 Alternatives Analysis

4.1 Language for development

Pros	Cons
Very fast	Machine Learning algorithms could be more robus
Easy to program with it	Doesn't have a graphical interface by default
Very big community	Modular architecture
I am comfortable with it	Lack of support for a native categorical variable
Open source	Not known by the director of the project
General purpose	

Table 1: Pros and cons of Python

Python

Pros	Cons
Easy to learn	Not intuitive for a programmer
Well integrated with graphical interface	Very slow
Very robust and well tested libraries	Very ugly graphs
Open Source	The majority of the community is from statistics
Well known by the director of the project	Many ways to do the same thing (different modules)
	It is very biased towards statistics
	Hard to use outside of RStudio

Table 2: Pros and cons of R

R

4.1.1 Chosen Option

Python

4.1.2 Reasons to chose that option

4.2 Running environment

Pros	Cons
Very simple	Not very interactive
Can use whatever text editor I want	Graphics integration not enabled by default
Very scalable	

Table 3: Pros and cons of text editor and console

Text editor and console

Pros	Cons
It's nice	It's ugly
It's very nice	It's really ugly

Table 6: Pros and cons of ...

Pros	Cons
Very easy to integrate code, graphs and documentation	A little bit buggy
Easy to show results to the director	The interface for writing code is not confortable
Open Source	Very interactive
Extensions and addons to make a GUI	Not scalleble for large projects
Very agnostic of the code	

Table 4: Pros and cons of Jupyter Notebook

Jupyter Notebook

Pros	Cons
It's nice	It's ugly
It's very nice	It's really ugly

Table 5: Pros and cons of ...

Atom's Hydrogen

Python IDE

4.2.1 Chosen Option

Jupyter

4.2.2 Reasons to chose that option

4.3 Machine Learning Algorithms

Pros	Cons
Very simple	Can't solve non-linear problems
	Designed for regresion
	Requires many hyperparameters

Table 7: Pros and cons of ...

Pros	Cons
Very stable	Also, its stability
Designed for classification	
Capable of non-linear problems	
No need for hyper-parameters	

Table 8: Pros and cons of ...

Linear Regression

Logistic Regression

Naive Bayes

KNN

SVM

Decision Tree

4.4 Chosen Option

Decision Tree, Logistic Regression and Linear SVM.

Pros	Cons
Very simple and fast	Naive assumption
Designed for classification	

Table 9: Pros and cons of ...

Pros	Cons
It's nice	Need to chose the type of distance to use
It's very nice	Need to chose the number of neighbours

Table 10: Pros and cons of KNN

Pros	Cons
It uses kernels, just like RFF	Normal algorithm just discriminates on two classes
Suitable for classification	Training is so hard

Table 11: Pros and cons of SVM

Pros	Cons
Very unstable	Needs a lot of hyperparameters
Suitable to use bagging	It is not well implemented in scikit-learn

Table 12: Pros and cons of Decision Tree

APA
All the Machine Learning theory

Programming
Skills to develop a very large program

Data Mining
Theory on Decision Trees and Random Forest
Theory on PCA
Development tools like Jupyter Notebook

Probability and Statistics
All the probabilistic concepts

4.5 Reasons to chose that option

5 Knowledge Integration

6 Implication and Decision Making

6.1 Meetings with director

6.2 Goals achievement

6.3 Rigour in scientific procedures

6.4 Workflow

7 Laws and regulations

7.1 My responsability

7.2 Others responsability