Université catholique de Louvain

LINMA2472 : Algorithms in data science

Homework 2 - part II : Random Fourier features (RFF)

Students :	Vincent Cammarano - 5391 21 00
	Louis Parys - 7256 17 00
	Mattias Van Eetvelt - 1660 18 00
Group :	6
Professor :	Jean-Charles Delvenne
	Gautier Krings
	Estelle Massart
	Rémi Delogne
	Bastien Massion
	Brieuc Pinon

November 2022

1 Introduction

In this part of the second homework, three different classifiers have been trained and tested with the MNIST dataset from the Keras package of the TensorFlow library. The goal is to investigate the classifying computation speeds when using random Fourier features (RFF). In this context, 10.000 training and 60.000 testing instances have been used.

2 Different classifiers methods comparison

The goal of the random Fourier features is to reduce the complexity by *explicitly* approximating the high dimensional feature space $\mathcal{V}$ in a lower dimension $\mathcal{D}$ using a randomized map $\mathbf{z} : \mathbb{R}^D \mapsto \mathbb{R}^R$, where ideally $R \ll N$, for N data points. For the positive definite, shift-invariant kernel k

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle\, \varphi(\boldsymbol{x}), \varphi(\boldsymbol{y})\, \rangle_{\mathcal{V}} \approx \mathbf{z}(\boldsymbol{x})^\top \mathbf{z}(\boldsymbol{y})$$

In this homework, the Gaussian kernel was used therefore yielding the following mapping for $\boldsymbol{x}$

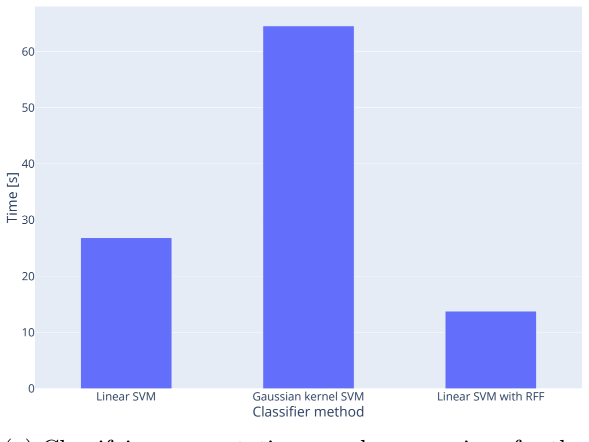
$$\mathbf{z}(\mathbf{x}) = \sqrt{\frac{2}{\mathrm{D}}}\, cos(\boldsymbol{\omega}^\top \boldsymbol{x} + b)$$

where

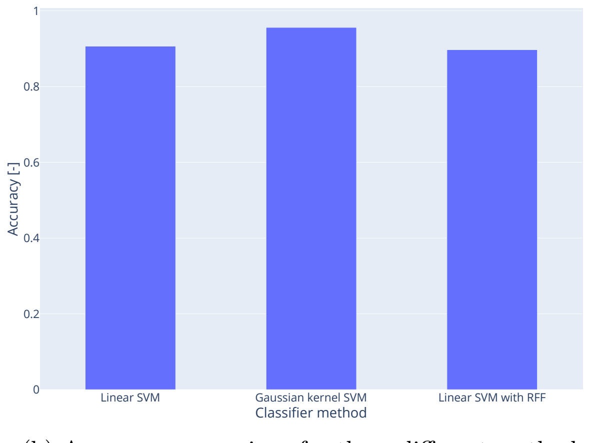
$$\boldsymbol{\omega} \sim N(0,\, 0.01^2) \qquad \text{and} \qquad b \sim Un[0,\, 2\pi]$$

Now that the RFF method is defined, let us compare different classifiers. The various classifiers used are the following : linear SVM, Gaussian kernel SVM and linear SVM with RFF. It is to be noted that the RFF method is used with random features $D = 300$ and a standard deviation $\sigma = 0.01$.

As shown on figure 1b, the accuracy is good for all three methods with the highest value of 0.95 corresponding to the Gaussian kernel SVM. Although the accuracy is consistent through the various methods, the time taken for the classifiers varies a lot as it can be observed on figure 1a. The most accurate method is also the most costful and runs in just over 60 seconds. However, the RFF methods is the faster one with a value of 14 seconds. Thus proving the initial claims of complexity reduction when using RFF.



(a) Classifying computation speeds comparison for three different methods



(b) Accuracy comparison for three different methods

FIGURE 1 – Methods comparison

3 RFF : varying parameters

In this section, the impact of the number of random features D is investigated. A Support Vector Machine (SVM) with RFF has been tested with the following various values of D : 10, 100, 300, 500, 700 and 1000.

It is worth bearing in mind that during the classifier training, the number of iteration was limited to 100.000 in order to avoid excessive computation time. This can be done by using the `max_iter` argument in the `svc.SVM()` function.

3.1 Accuracy of the linear SVM with RFF

Figure 2 clearly shows that the accuracy is a function of the number of random features D. The maximum accuracy is obtained with $D = 1000$ and is equal to 0.942. However it seems like there is a threshold value for the accuracy that can not be exceeded, even if the value of D keeps increasing.

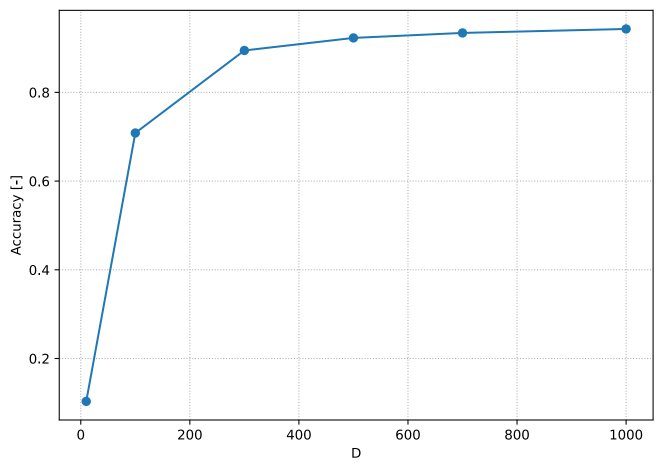


FIGURE 2 – Accuracy of a SVM with RFF for different values of D

3.2 Time taken to create the RFF

The figure 3 displays the expected results i.e. there is a linear relationship between the time needed to create the random Fourier features and the number of random features D. As the value of D grows, so does the number of W and b that have to be sampled from their respective distribution. Then, the mapping $\boldsymbol{z}(.)$ has to be applied for increasing size of matrices as well.

3.3 Time taken to train the RFF

On figure 4, a minimum can be observed for a value of $D = 300$. For values of $D < 300$, the training times are far more important and for values of $D > 300$, the training times seem to increase linearly with the parameter D. This observation could be explained by the fact that for values of $D < 300$, the classifying is more complex as there isn't enough random features, thus increasing the training time, and compromises have to be made yielding poor accuracy as observed on figure 2 for $D < 300$. However for $D > 300$, better classifying can be achieved but more computation time is needed. This yields better results as it can again be observed on figure 2 for $D > 300$. Therefore, $D = 300$ seems to be the optimal number of random features.

3.4 Time taken for classifying the instances

As observed on the previous figure, there is a (local) minimum on figure 5 as well. There seems to be a linear relationship between the classifying time and D, except for $D = 300$ where the time taken decrease

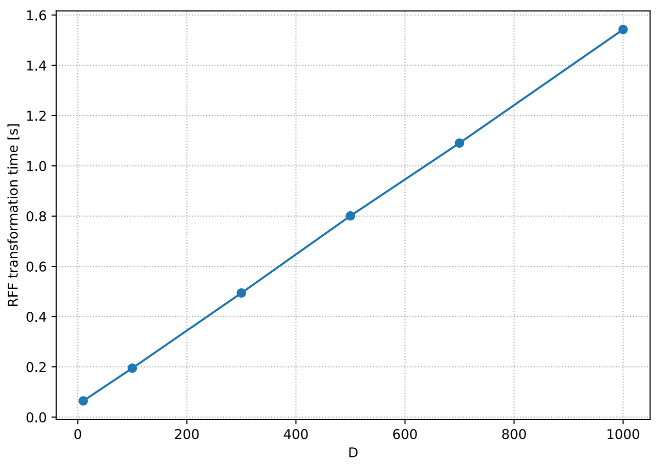


FIGURE 3 – Transformation time needed to create the RFF for different values of D

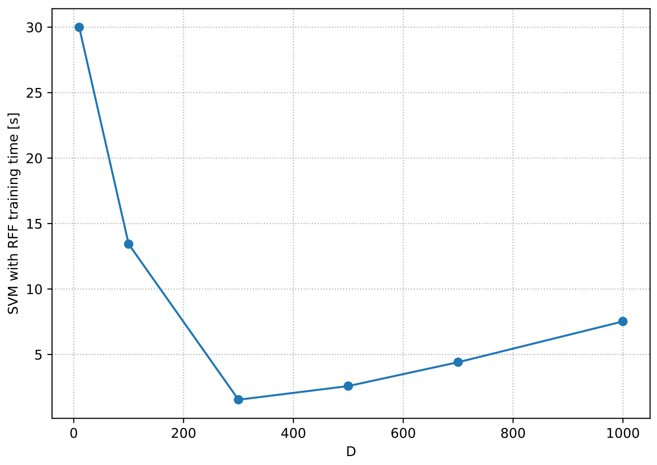


FIGURE 4 – Training time for a SVM with RFF classifier in regard to different values of D

strongly. This confirms the fact that $D = 300$ is indeed an optimal value for the number of random features, in this particular case.

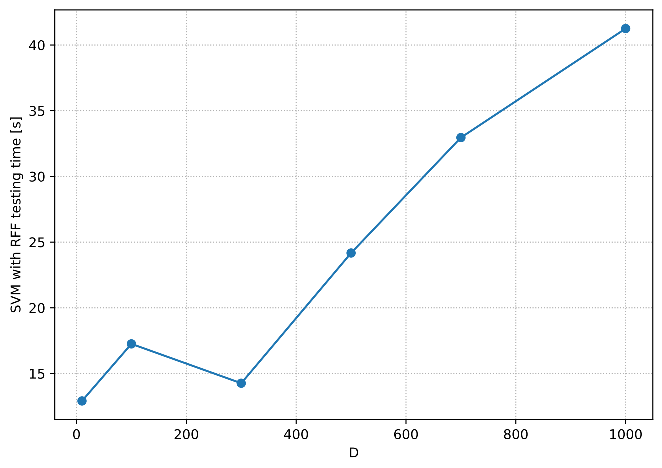


Figure 5 – Testing time for a SVM with RFF classifier in regard to different values of D