LINM2472 - Project 2

Students : Thonnard Julien - Vermeulen Corentin

1 Part 2: Random Fourier features

1.1 Q1. Testing time with default parameters

The time to classify the 60.000 testing instances:

- SVM with linear kernel on non-transformed data took about 35 seconds for an 0.91 accuracy
- SVM with Gaussian kernel on non-transformed data took about 130 second for an 0.95 accuracy
- SVM with linear kernel on transformed data with random Fourier features took about 17 seconds for an 0.90 accuracy

Accordingly to the course theory, the complexity for computing large kernel matrix is $O(nd)$ and while for linear classification it is about $O(n + d)$

It fits well what we experimentally got since SVM linear on non-transformed data as $d = 784$ and SVM linear on transformed data as $d = 300$ which is about one half of the original dimension. The time is also reduced by a factor two. The SVM with Gaussian kernel is more complex to compute, about four time more complex than SVM linear on non-transformed data and eight times longer than on transformed data.

1.2 Q2. Varying Parameters

We tried different values for D to see how the accuracy, transforming time, training time and testing time evolved as the new dimension changed.
We tested the following values for $D = \{10, 100, 250, 500, 750, 784, 1000\}$. The value for the standard deviation remains the same: $\frac{1}{100}$.

Next you will find plots of the results. We added on the plot the results for the two other techniques (SVM with linear kernel and SVM with Gaussian kernel on non-transformed data). Just be aware that the x-axis for those two series is not relevant since the dimensions of those computation is the original data set dimension: 784 features and does not depend on the value of D.

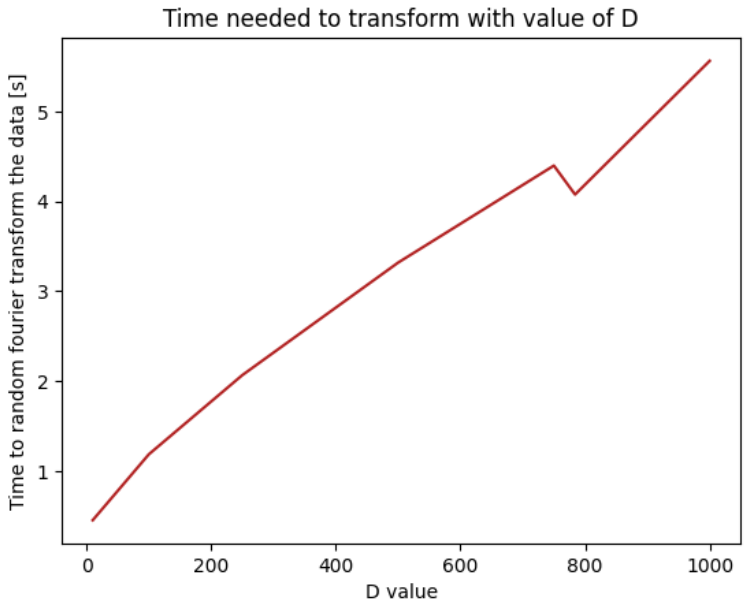

Figure 1: Transformation time depending on D value

It seems logic that the transform time increases linearly with the increase of the D-value because we ask for more dimensions. That linear increases is also related to the complexity of the algorhithm. The little decrease in time with D=784 is supposed to be normal. In this case the number of dimensions is not very different, just 34 more so the time needed must be more or less the same. In this attempt, the time needed for the transformation and for the rest was a bit lower for D=784 than for D=750 but for another the time for D=784 may be higher than for D=750.

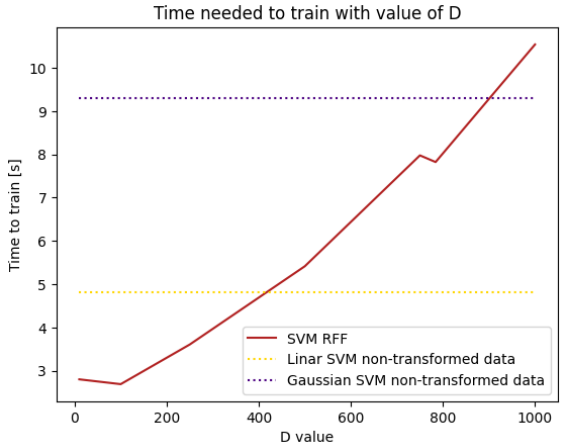

Figure 2: Test time depending on D value

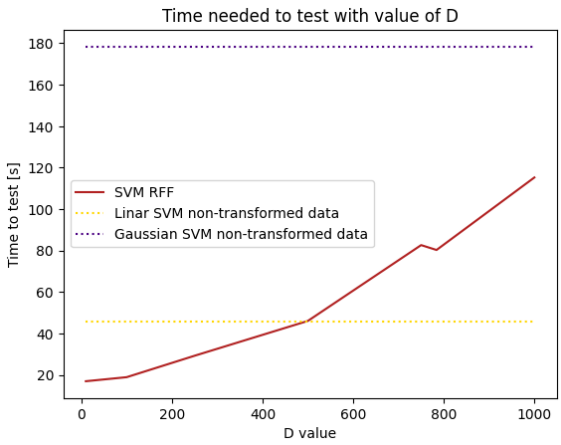

Figure 3: Training time depending on D value

The training and testing time increases linearly, which is also consistent with the algorithm complexity. The linear kernel SVM training and test time is crossed when D is around 400 and 500 meaning that the training and test time is faster on linear kernel SVM since it's dimension is 784.

Compared to the Gaussian kernel SVM, the train time is crossed around D=900 and is not crossed when D <=1000 for the testing time. Therefore the RFF is faster than Gaussian kernel SVM (also consistent with the algorithms complexities)

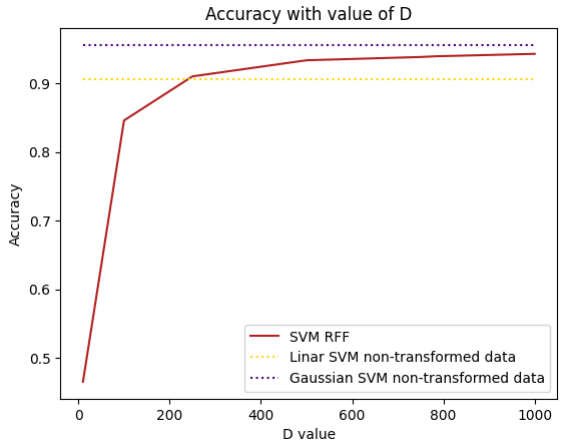

Figure 4: Accuracy depending on D value

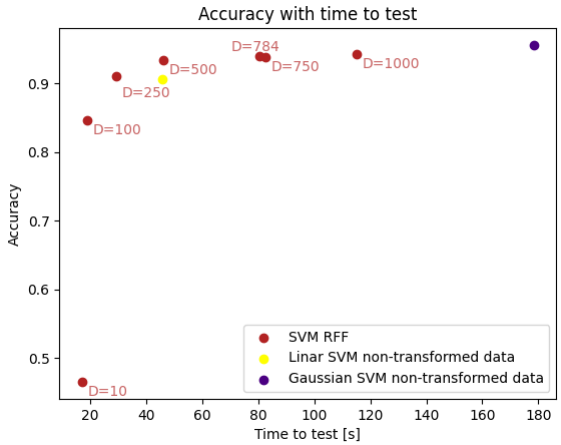

Figure 5: Accuracy with time needed to test

On the fourth plot we can see that the accuracy of the RFF (Random Fourier Features SVM) outperform the SVM with linear kernel quickly, around D=250 while the dimension of SVM with linear kernel's dimension is 784. The Gaussian kernel SVM is just not outperformed by the RFF but it takes much more time for almost the same accuracy as we can see on the fifth plot. The evolution of the accuracy increases quickly at as D increase when D is low then it tends to stagnate.