Lab 4: Support Vector Machines

Santiago Bernal
Rodrigo Arias

January 8, 2018

1 Exercise 1

In the file `ex1.py` three generators are used to provide 3 different random datasets. The SVM algorithm is applied to the train set of each dataset, and is plotted in the figures 1, 2 and 3. The train data is shown with circle markers, while the test data uses crosses. The SVM decision regions are also filled with color, in order to visually see the behavior of the classifier.

We see that the datasets 1 and 3 are linearly separable, so a linear function should be used. The Gaussian kernel is used in the latter, but is behaving similarly to a linear kernel, so the classification is correct. The dataset 2, in contrast, is clearly not linearly separable, so a linear function should not be used. In the figure 4 a Gaussian kernel was used, to see the difference. We see a better classification as expected.

The figure 2 shows a correct classification of 100%, even if the classifier is not accurate. That can be explained because the test set provided with the generator, is only choosing elements that are placed in the outermost groups.

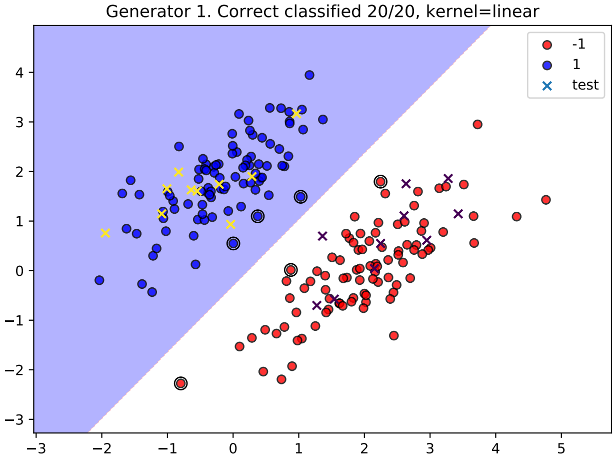

Figure 1: Dataset of the generator 1

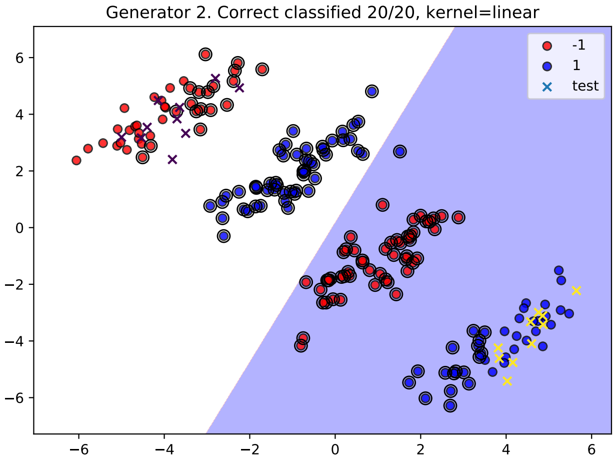

Figure 2: Dataset of the generator 2

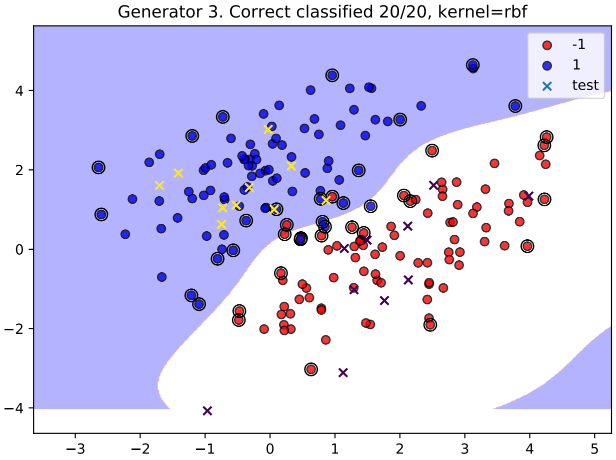

Figure 3: Dataset of the generator 3

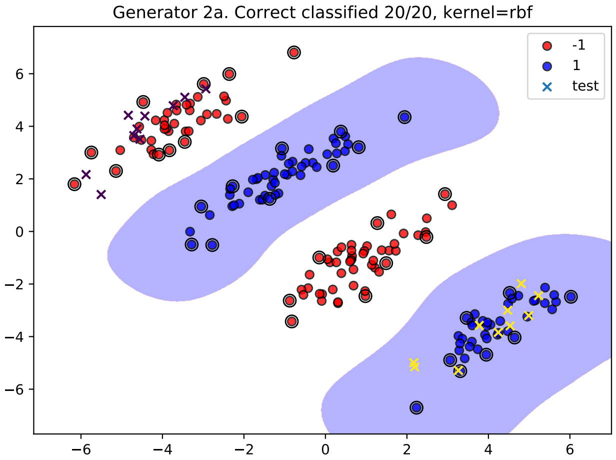

Figure 4: Dataset of the generator 2 using a Gaussian kernel

2 Exercise 2

In the file `ex2.py` we adapted the previous parser for ARFF files, to read the training and testing datasets. The function `svm_classify` implements the SVM algorithm, and computes the time and score (the accuracy, defined as the number of correct classified classes over the total). We only take into account the numeric part of the datasets; the nominal features are ignored. Also we scale the data in order to facilitate the work to SVM [4]. For the kernel functions considered we use three of the predefined ones in sklearn: `linear`, `rbf` (Gaussian), and `sigmoid`.

The result of each configuration for each fold is stored in memory, until all results are finally computed. Then we use the mean accuracy of each configuration, to reduce the noise. We use the provided folds for the datasets, which are divided into 10 groups of folds, so we take the mean of the 10 results.

First, a quick run with the default values and the different kernel function selects the best one. Then each configuration of parameters is build by choosing one value for `C` $\in \{1, 4, 6, 8\}$ and another for `cgamma` $\in \{1, 4, 5, 8\}$. A total of 16 configurations are run for each fold using the best kernel as shown in the table 1. The `gamma` parameter of SVM is the division of `cgamma` by the number of features.

conf.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`C`	1	1	1	1	4	4	4	4	6	6	6	6	8	8	8	8
`cgamma`	1	4	5	8	1	4	5	8	1	4	5	8	1	4	5	8

Table 1: The configuration number and the parameters used

Results are compared in terms of accuracy and efficiency. We selected a small dataset `hepatitis` and a bigger one `vowel` to perform the experiments.

2.1 Evaluation

In order to compare the accuracy results of each configuration, a statistical test can be used to determine if the improvement is significant, or is just because of random deviations. A t-test was first considered, but was discarded as it would need a pairwise comparison between all pairs of configurations [1]. An ANOVA test can be used in a group of observations, so it was used here.

The null hypothesis is defined as: the mean of all the groups is the same. For the first dataset, we obtain a p-value of 1.0, so we cannot reject the hypothesis. The box plot in the figure 5 shows that almost all configurations lead to a similar accuracy.

For the dataset vowel, it can be graphically shown in the figure 6 that some configurations have a very remarkable difference. Performing an ANOVA test gives a p-value of 5.75×10^{-22} which is a very significative. We reject the hypothesis that all the accuracies have the same mean.

A more detailed observation can lead to the conclusion that only the configurations 1, 5, 9 and 13, the ones with `cgamma` = 1, are significatively different from the rest. If we remove those observations from the ANOVA test, we get a new p-value of 0.353. So we cannot reject that the other configurations have the same mean, thus we conclude that there is no significant change in the other parameters tested to be produced by a real change in accuracy.

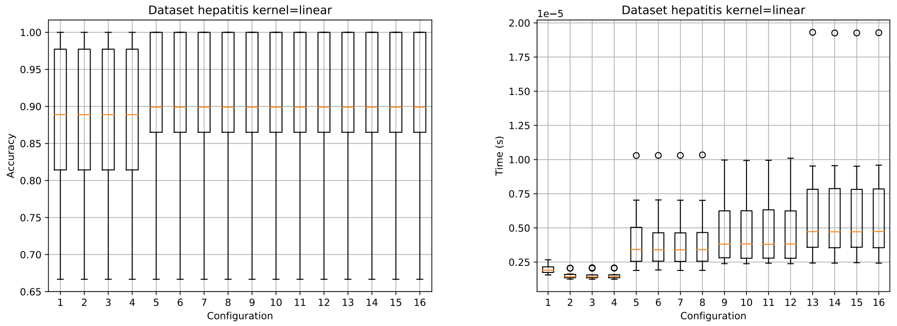

Figure 5: Accuracy and time of different configurations for hepatitis

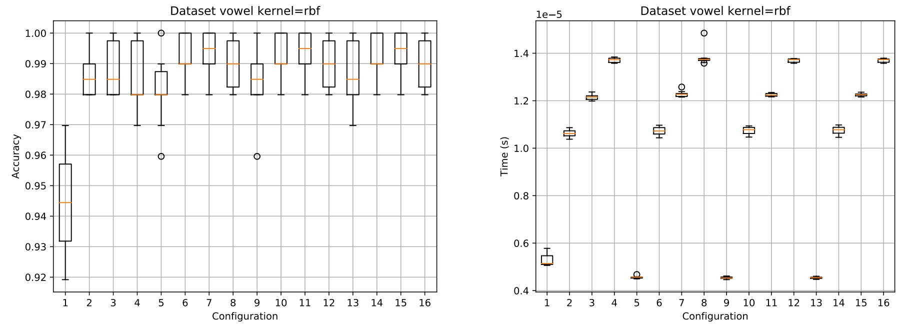

Figure 6: Accuracy and time of different configurations for vowel

A time analysis reveals that the linear kernel is faster than the Gaussian. The time has been scaled by the number of samples and features of each dataset. The linear time is around 2 and 5 µs while the Gaussian is between 5 and 14 µs

We also observe big fluctuations in time when changing the parameters. The linear kernel doesn't take into account the `cgamma` parameter. In contrast, for the Gaussian kernel has the biggest impact in the running time.

3 Conclusions

We have seen that different datasets lead to better results with different kernels and parameters.

The best kernel functions depend on the data, and how much we know beforehand. If we know that the data can be separated linearly, then a linear classifier will be the best option since it may save computational time and could prevent overfitting the data.

In the figure 2 we can visualize how the linear classifier doesn't classify the data correctly and then how the classification is made better when using the RBF classifier in the figure 4. For non-linear cases, a linear classifier may not have a good accuracy. The RBF (Radial Basis Function or Gaussian) is more commonly used since it can better adapt to more complex data shapes [2, 3].

Based on the results of the exercise, this statement can be reaffirmed since the best results for non-linear models where obtained using the RBF classifier. But overall, there is not a *best* kernel function to use for a SVM classifier since the results can be different for different models.

We have also shown the differences in performance of the different kernels used, as well as the effect of the parameters.

4 Execution

In order to execute the code, python 3 is needed, the version tested was 3.6.4. You will also need the packages scipy, numpy, matplotlib, pandas, sklearn. Also a helper function was used from the package mlxtend, in order to draw the SVM regions. Run the scripts from the directory where they are as `python ex1.py` or `python ex2.py`.

References

[1] Janez Demšar. 2006. *Statistical Comparisons of Classifiers over Multiple Data Sets.* J. Mach. Learn. Res. 7 (December 2006), 1-30.

[2] `https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-2-W3/281/2014/isprsarchives-XL-2-W3-281-2014.pdf`

[3] `https://www.kdnuggets.com/2016/06/select-support-vector-machine-kernels.html`

[4] `http://scikit-learn.org/stable/modules/svm.html#tips-on-practical-use`