Advanced Learning Models: Data Challenge

David Emukpere **Habib Slim**

For this data challenge, our task was to predict whether input DNA sequence regions were binding or not to specific transcription factors.

We achieved 71.26% overall accuracy on the three datasets (**72.20**% accuracy on the public leaderboard and **70.33**% accuracy on the private leaderboard), and we will describe here the kernel functions (1) and classifiers (2) we experimented with for the challenge.

We provide in section 5 the full sources containing all of the methods mentioned in this report.

1. Kernel functions

In order to improve the performance of the classifiers used, we have considered and implemented various kernel functions operating on string inputs.

Gapped kernels. We implement and experiment with gapped substring spectrum kernels (Lodhi et al., 2002), which allow gaps of pre-specified length to appear in the input sequences. We use $g = 11$ for the length of g-mers and $k = 9$.

Spectrum kernels. The main kernel function that we used throughout the project was the spectrum kernel (Leslie et al., 2001). We adjust the length of k-mers empirically depending on the dataset we are training for, based on validation accuracy.

Linear combinations of spectrum kernels. We experiment with weighted linear combinations of spectrum kernels with different k lengths, and tune it for each dataset.

We tune hyperparameters empirically using a $1500 : 500$ train/test split on each dataset, with a simple grid search algorithm.

Weighted Degree Kernel. We also experiment with weighted degree kernels (Rätsch & Sonnenburg, 2004) with and without shifts, which use positional information from multiple k-mers in the input sequences to extract features.

In that sense, this kernel function is analogous to linear combinations of spectrum kernels with varying k values. We experiment with $k = 5$ and $S = 2$ for shift lengths.

2. Classifiers

Alongside the previously mentioned kernel functions, we have implemented the following classifiers:

Kernel Ridge Regression. We implement KRR using the `scipy` package for linear algebra, and tuning the λ parameter using grid search.

Kernel Logistic Regression. We implement KLR using `scipy`, following the iteratively reweighted least squares method. We use the difference in 2-norm of the α vector between two iterations as an adjustable stop-criterion, and tune the λ regularization parameter used for solving the weighted kernel ridge regression problem separately.

C-SVM. We implement the C-SVM formulation using the `cvxopt` package, which provides quadratic programming solvers.[1]

Bootstrap aggregating. Finally, we write specialized classes to perform bagging using each of the aforementioned classifiers. We randomly exclude 5% of the training set in a maximum of $K = 7$ models, and aggregate votes/average probabilities to make a global decision.

3. Results

Method. We evaluate kernel functions and classifiers using a $1500 : 500$ train/test split from the training set of dataset 2, and with the best-performing hyperparameters found. The accuracy figures provided are averaged over $N = 7$ cross-validation splits.

Kernel functions. We compare the kernel functions considered by tuning their hyperparameters on a single split and measuring their average accuracy, with various classifiers. Results are given in table 1. Using linear combinations of spectrum kernels gave overall the best validation accuracies, while computing Gram matrices fairly fast. This order was also verified in the other two datasets.

Classifiers. Using a selected combination of spectrum kernels for dataset 2, we compare the accuracies of our classifiers in figure 1.

[1] Our C-SVM implementation follows a tutorial by Mathieu Blondel (Google Brain).

KERNEL	ACCURACY (%)
SPECTRUM LINEAR COMB.	70.51 ± 1.53
GAPPED	69.00 ± 1.41
WDS	64.33 ± 1.93

Table 1. Validation accuracies on dataset 2 for various kernel functions and with the KRR classifier.[3]

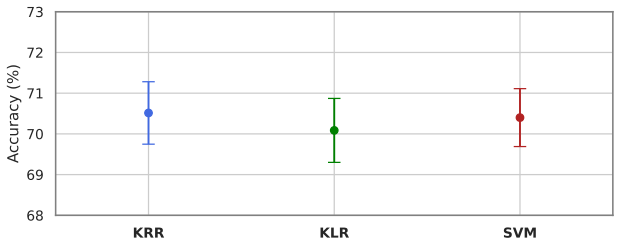

Figure 1. Validation accuracies on dataset 2 for each classifier.

Overall, the choice of classifiers doesn't seem to have a big impact comparatively to the choice of an effective kernel.

Effects of bagging. We evaluate the bagged models using a single split, and gradually increase the number of models in the ensemble. Again, we use the same combination of linear spectrum kernels with the three classifiers. Results are given in figure 2 below.

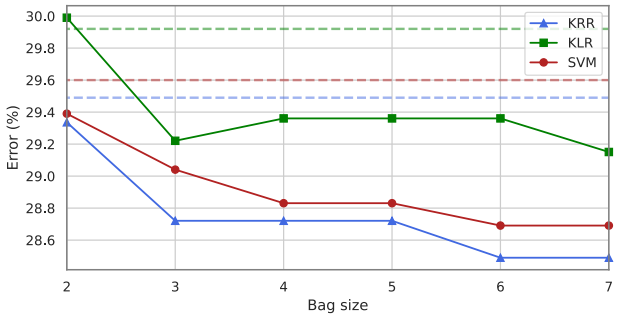

Figure 2. Validation errors and size of ensembles (dashed: error for single classifier trained on the full data)

Overall, bagging only marginally improved the overall accuracy: we observe at most a 1% decrease in error with $N = 7$.

Weighting spectrum kernels. We experiment with alternatives to uniform weighting for the linear combinations of spectrum kernels by weighing longer sequences more, but find that this does not significantly improve validation accuracy.

[3]Gapped kernels and WDK kernels accuracies are only evaluated over two splits because of the high computation time.

Data augmentation. We also experiment with data augmentation by computing reverse complements of training sequences, but measure no significant improvements in accuracy.

4. Conclusion and possible improvements

Overall, the private leaderboard results indicate that we were significantly overfitting on the public leaderboard data.

Our choice of kernel functions could have benefited from a more robust selection process like cross-validation, but came at the cost of very long computational times - which was hard to handle during the project given the limited computational resources available to us.

In order to improve our overall results, numerous directions could have been taken. Adding a tolerance to strings within small Hamming distances to each other could have benefited our spectrum kernel when computing occurrences: this is essentially what the Mismatch Kernel (Leslie et al., 2004) proposes to do.

5. Sources

The entirety of our source code can be found in the following GitHub repository:

`https://github.com/HabibSlim/AdvancedLearningModels`

Instructions to reproduce our two submissions are given in the README file.

References

Leslie, C., Eskin, E., and Noble, W. S. The spectrum kernel: A string kernel for svm protein classification. In *Biocomputing 2002*, December 2001. doi: 10.1142/9789812799623_0053.

Leslie, C. S., Eskin, E., Cohen, A., Weston, J., and Noble, W. S. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, January 2004. doi: 10.1093/bioinformatics/btg431.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444, 2002.

Rätsch, G. and Sonnenburg, S. *Accurate Splice Site Prediction for Caenorhabditis Elegans*, pp. 277–298. 01 2004.