AFRICAN MASTERS OF MACHINE INTELLIGENCE

Data Challenge:Project Report

Amel Abdelraheem

July 5, 2022

1 PROBLEM FORMULATION

Use machine learning algorithms - in particular linear models utilizing kernels - to classify DNA sequences as belonging to the SARS-CoV-2 (Covid-19) read *or* not.

2 APPROACH

(i) Embeddings

- Pre-computed embeddings were provided, these embeddings were generated following a spectrum kernel approach with kmers of length 3. Most experiments were carried using these vectors
- New embeddings were generated following similar approach, however before generating the final vectors, a compression was made to add low-occurring kmers to the most similar high-occurring kmer as a way to account for errors in the sequences. No normalization was done to the counts of kmers. The similarity was computed based on the Jaccard similarity between sets
- Another set of embeddings was generated without the compression step (referred to in the report as only counts).

(ii) Backbones

Two backbones were used:

a) Logistic regression
b) Support vector machines (SVM)

(iii) Kernels

Three Kernels were used:

a) Radial basis kernel (RBF)

$$K(x,x') = \exp(-\frac{||x-x'||^2}{2\sigma^2}) \tag{2.1}$$

b) d-Polynomial kernel

$$K(x,x') = (xx'+1)^d \tag{2.2}$$

c) Laplace Kernel

$$K(x,x') = \exp(-\gamma|x-x'|) \tag{2.3}$$

3 Results and Observations

Model	**Kernel**	**C**	**sigma**	lambda	**Val. %**	**Test. %**
Logistic Regression	-	-	-	0.01	89%	87%
Logistic Regression	RBF	-	median	0.001	95%	**96.6%**
SVM	RBF	1	1	0.001	95.2%	90.8%
SVM	Polynomial (1st)	1	-	0.001	95.2%	-
SVM	Polynomial (3rd)	1	-	0.001	95.2%	-
SVM	Polynomial (5th)	1	-	0.001	95.2%	92.4%
SVM	Laplace	1	gamma = 1	0.001	50.1%	-
SVM	RBF	1	0.1 quantile	0.001	62%	-
SVM	RBF	1	0.9 quantile	0.001	93%	-
SVM	RBF	1	median	0.001	-	91.6%
Logistic Regression	Polynomial(3rd)	-	-	0.01	87.7%	-
Logistic Regression w/ new embed.	Polynomial(3rd)	-	-	0.01	94.75%	**95%**
Logistic Regression w/ only counts	Polynomial(3rd)	-	-	0.01	95.75%	95.2%

Table 3.1: Table showing experiments carried. *Test.%* is taken from private leaderboard

(i) **Architecture changes**

- Logistic regression's performance was boosted quite a bit with the introduction of kernels.
- Polynomials performance seems to saturate (different degrees seems to maps the same decision function).
- The laplacian kernel seems to have a poor performance on this data.

(ii) **Embeddings changes**

- Looking at the two validation results from table 3.1, we can see that the same settings with different embeddings, we see that the linear kernel benefitted significantly from the simple change in embeddings. However, offline experiments suggests the RBF kernel actually suffered drastically from the switch in embeddings.

4 Future Considerations

(i) More time on hyperparameter tuning might improve the overall performance especially for the laplacian kernel
(ii) How embeddings are calculated seems to have a strong influence on performance, suggesting more time spent there would yield stronger classifiers.
(iii) The use of more kernels can also be explored

5 Codes

Code to produce the top 2 submissions can be found here Github repo.
Complete codes used for this challenge can be found here Code, more details about how to run the codes and reproduce the rest of results from table 3.1 can be found inside the notebook