Machine Learning Prediction of Measurement Conditions from Gene Expression Levels

Juliette Boucher Grenon, Sabrina Wang

Introduction

In the experiment on epigenetics and memory, Johannes Graffl's lab group has measured gene expression levels in mouse brain under three different measurement conditions. For a given set of labelled, 5000 measured cells with 32285 genes each, we wish to be able to use machine learning means predict the measurement condition for a future, unlabelled cell.

Data Inspection

To begin with, we wish to investigate into the behaviour of the predictors. First we randomly selected a few genes to observe their correlation, but the randomly selected, standardised features do not give much correlation. Then we ran an analysis through the features and found a total of 471 pairs with correlation close to 1. We conclude most features are uncorrelated therefore need to be treated separately.

Then we had a look at the structure of the data through UMAP and t-SNE plots by reducing the data into 2 dimensions. Fig. 1 shows that although there seems to be 3 main clusters in the data, they don't correspond at all to the labels we are trying to predict, and there does not seem to be a clear decision boundary. We then tried K-means and hierarchical clustering and got similarly poor results. Hence models using clustering or neighboring relationships are probably not a good idea.

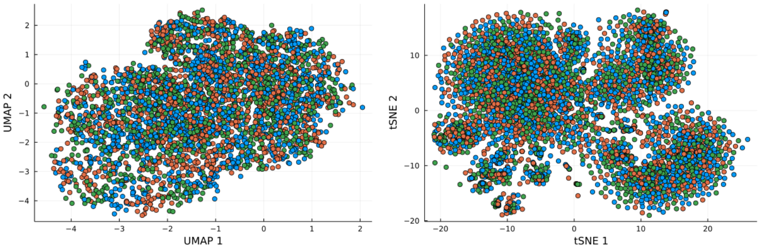

Figure 1. Left: UMAP plot with 50 neighbours, Right: t-SNE plot with 80 neighbours and 1000 iterations

Lastly we looked at the PCA plots. Fig. 2 shows again that there are a few features with correlation but no overall correlation. Looking at the proportion of variance explained we can see that the first few PCs are much more important than others, but still not enough to accurately predict the data.

Data Transformation

We wish to reduce the number of features to make computation more efficient. First we removed all the constant features with 0 standard deviation as they would not contribute to the prediction. Then we found all the pairs of features that gave correlation close to 1 and kept only one of the pair. This reduces the number of features to 25858 features. Lastly we standardised the features. The proportion of variance explained in Fig. 2 suggests that 4798 features are enough to capture 99% of the data, and 4976 features are enough to capture 99.9% of the data. (Note that we would need to do this again later but just for the training set when we split up the data into training and validation).

Machine Learning Methods

We split up the provided data of 5000 cells into 3500 for training and 1500 for validation. We then proceeded to test a variety of machines with standard parameters to form a better idea of which machines would yield the best results. We then tuned these machines, either manually or using the TunedModel function. At this point, the results were

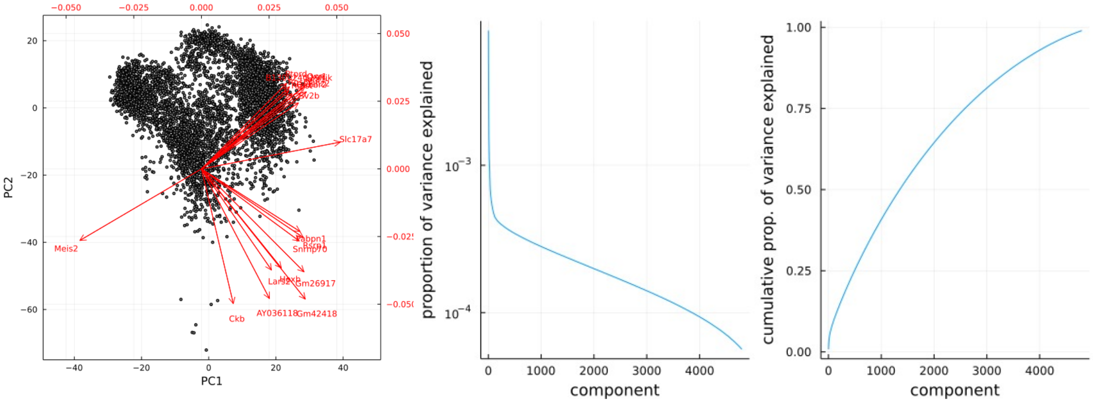

Figure 2. Left: Biplot of 20 largest variance features, Middle and Right: proportion of variance explained

mostly good, but we still needed to further increase the tuning speed and get even better predictions. To do so, for most models we transformed the training data into the PCs, kept only the first 3500PCs of the training set (gives more than 99.9% of the variance of the training set) and calculated validation error by projecting the validation set on to the same PCs. As expected, clustering do not corresponding to the classes, KNN-classification does not yield promising results. Neural networks of different shapes and different activation functions are tested. A tuned Tree method gave good, though not the best results.

Best Linear Model: Logistic Classifier with Regularisation

A logistic classifier is defined with L2 Ridge penalty. The optimum regularisation λ is tuned using a tuned model, with 5-fold cross-validation, and trained using the PCA data. This gives a validation mis-classification rate of 0.104 at $\lambda = 1e-5$. Note that without transforming into PCA space, the validation error obtained is 0.1126, this shows by keeping only 3500 features in the training set, we de-noise the data and avoid a certain degree of overfitting.

Best Nonlinear Model: Neural Network Classfier with ADAMW optimiser

We experimented with neural networks with different number of neurons and different number of layers, as well as different activation functions (tanh, sigmoid and relu). We found out that the tanh and sigmoid functions are slow and have the vanishing gradient problem which make them yield poor results, thus relu is the optimum activation function. The optimum structure for the neural network consists of 4 hidden layers of (100, 50, 30, 50) neurons respectively, with a bach size of 32. Using the ADAMW optimiser with 60 epochs acts as gradient descent to find the minimum loss parameters which allows us to efficiently tune the model without overfitting. Note that here we used the non-PCA-transformed, cleaned and standardised data set as the PCA-transformed data give a slightly higher misclassification rate. This model gives a validation misclassfication rate of 0.116.

Conclusion

In this project, we have found the methods of Logistic Classifier and Neural Network Classifier the most adequate for predicting the labels. Looking at their mis-classification rates, we would conclude that linear methods are sufficient to classify the data. Interestingly, we noticed that for most methods, the misclassification rates related with the "KAT5" condition seemed to dominate, for both false positives and false negatives. A step towards a better overall classification could therefore come from either finding genes that allow for better discrimination between this condition and others, or finding a method that is better at differentiating similar looking inputs. Finally, to generate the predictions for the test set with the best certain we have, we re-trained the selected machines on whole cleaned (and PCA transformed) data set instead of just the training, and used these on the cleaned (and PCA transformed) test set.