Machine Learning - Assignment 1

Gianpiero Cea 1425458

February 12, 2018

1 Initial Data Exploration and Preprocessing

At first I decided to plot various histograms of the distributions of the various features in both probeA and probeB: this suggested the presence of some sort of corruption to the data that was then easy to recognize as a swap between columns. So I wrote a function to reorder this values. After this I tried polynomial feature expanding: I expanded up to order 4 since any other expansion was not useful for both regression or classification. After this I defined the standard unit scaler and I standardised the probeB and the probeA data (the latter with the 'tna' and 'class' columns removed). I also did all possible pairplots between the 13 features of probeB and the 13 features of probeA without using 'class' and 'tna' features after standardisation. From a graphical inspection and also from the correlation plots of these features, it was clear that the variables add all the same kind of correlation between them and therefore I assumed it was reasonable to assume that we could treat probeB as a test set for the model trained on probeA. Finally I extensively used feature selection for both regression and classification: I tried to using the SelectFromModel function of sickit-learn to pick the more relevant features in all cases: this generally produced improvements in the model. Finally I generally used 10-fold cross validation to measure how well my models were doing.

2 Predicting Class (Classification)

It was clear from various classifiers that I trained with cross validation and feature selection that the 'tna' variable and other higher ords terms including 'tna' like 'tna**2' were really predictive in terms of the 'class'. Because of this I thought of first doing regression and the using my best guess (with the R2 score) for the missing TNA in probeB. I have used both methods that we have studied for classification: k Nearest Neighbour and Decision Tree. In order to find optimised hyperparameters for these models (including k for kNN, max depth for DT and the feature to select) I have tried a grid search with iterating through different possible values. Without using the 'tna'feature I got an AUC cross validation score of 0.750459254993 with DT trained on the second order features ['c3', 'm1**2', 'm1 n3', 'n3**2', 'p1**2']. Without 'tna' I also found the kNN classifier with k= 157 and feature selection ['c3', 'm1**2', 'm1 n3', 'n3**2', 'p1**2'] we get an AUC cross validation score of: 0.78377912046 . Using the actual 'tna' values I found as expected some really good scores: I got an AUC cross validation score of 0.923220441558 obtained with a decision tree of depth 5 and with the following features: ['c3 n3','m1**2' ,'m3**2','n3**2','n3 p1','p3 tna','tna**2']. Unfortunately using the guessed TNA values with regression I did not get any improvement, the best one being a score of 0.74 I obtained when training a decision tree classifier. So I decided to predict the class using a 157-NN classifier as above to predict the class.

3 Estimating TNA (Regression)

I tried 4 main methods of regression: plain Linear Regression, Ridge, Lasso and Decision Tree Regressor. As before I tried different kinds of parameters including the regularisation constant 'alpha' for Ridge and Lasso , the feature to select and the order of polynomial expansion. I have measured how well the estimators were doing by using both mean squared error and the determination score over 10 fold cross validation. By a somewhat systematic search i got the following results: using Ridge with alpha=10.5 gives a R2 score of about 0.765028763036 when applied to the order 2 data with feature select to be: ['c1', 'c2', 'c3', 'c1 c2', 'c1 c3', 'c1 n3' ,'c1 p1', 'c1 p2' ,'c3 n2', 'c3 n3','m1 n3' ,'n1 p1' ,'n2 p1' ,'n2 p2' ,'p1**2' ,'p2 p3']. With Lasso on the order 2 polynomial feature expansion without any feature selection we gave on 10-fold cross validation an average R2 score of 0.85003332207 and MSE of 1.223448364383. Even better with features =[u'c3', u'm1 n3', u'm3 n3', u'n1 n3', u'p1** 2'] and alpha = 0.0043 I got an R2 score for model of 0.851696864765 and a mean squared error for model of: 1.21039657058. The decision tree regressor we used had a R2 score of 0.715773519397 with max depth=2 and with the data used the order 2 feature expansion with only the following features: [u'c3', u'm1 n3', u'm3 n3', u'n1 n3', u'p1**2']. So at the end I have picked Lasso model on order 2 polynomial feature expansion and the alpha = 0.0043.

4 Final considerations

In order to investigate the most predictive features, I have considered the most important features of the decision tree. Quite clearly the 'tna' feature was by far the most predictive: this is also shown by the fact that the classifier trained with the actual 'tna' values got some really high AUC score (about 0.92). Other than 'tna', some of the most predictive feature seemed to be c3,m3,n3. This was also confirmed by running a statistical test with SelectKBest. Finally it looks like the features are linked to the target in a quadratic fashion, since order 2 polynomial expansion was in general better for classification. In particular the order 2 features [u'c3', u'c3 m3', u'm1 n3', u'm3 n3', u'p1**2'] seemed very predictive, and this was consistent in different models.