MSA220 Exam

Calvin Smith
Gothenburg University

June 2020

Contents

1 Exercise 1: Classification and variable selection

1.1 Exploring the dataset

We are working with a high-dimensional dataset with binary labels. The data consists of a training set and a validation set. The training set consists of a feature matrix $\boldsymbol{X}_1 \in \mathbb{R}^{n_1 \times p}$, where $n_1 = 323$ and $p = 800$, and a vector of class labels $\boldsymbol{y}_1 \in \{0,1\}^{n_1}$. The validation set consists of a feature matrix $\boldsymbol{X}_2 \in \mathbb{R}^{n_2 \times p}$ and vector of class labels $\boldsymbol{y}_2 \in \{0,1\}^{n_2}$, where $n_2 = 175$.

Firstly, we should acknowledge that we are in the $p > n$ setting, meaning that we have more variables than observations. Thus, we have to beware of the curse of dimensionality as well as the instability (or inability) of estimating coefficients when, for example, using a traditional approach like logistic regression.

Further, when looking at the vector of class labels $\boldsymbol{y}_1$ we can see that out of 323 observations, 247 belong to the class ”0” and 76 observations belong to class ”1”. Thus, we have highly unbalanced data.

Another important aspect of data exploration before actually performing any analysis is to check for correlation between features. For this purpose, we compute the correlation matrix of $\boldsymbol{X}_1$. From this we can compute a table showing which variables are highly correlated with each other and their respective correlations. As a limit we only include correlations bigger than or equal to $|0.7|$. Table 1 is a subset of the full table and shows that feature 768 has a high correlation with 23 other feature. Similar results are obtained for several other features in the training dataset. Thus, we get an indication that the data contains highly correlated features which should be taken into account when performing further analysis, especially when choosing appropriate models.

	var1	var2	corr
1	637	768	0.71
2	702	768	0.72
3	743	768	0.74
4	120	768	0.75
5	49	768	0.75
6	471	768	0.75
7	151	768	0.76
8	283	768	0.77
9	681	768	0.78
10	325	768	0.80
11	757	768	0.81
12	104	768	0.81
13	75	768	0.81
14	118	768	0.82
15	566	768	0.82
16	556	768	0.82
17	83	768	0.85
18	470	768	0.85
19	136	768	0.86
20	434	768	0.87
21	496	768	0.87
22	137	768	0.90
23	154	768	0.91

Table 1: Table showing that feature 768 has a high correlation with 23 other feature in the training set.

Hence, we can conclude that our data consists of correlated features with class responses that are highly imbalanced.

1.2 Methods

1.2.1 Elastic net

Choosing an appropriate model for our data is all about our prior knowledge of the data and the reasonable assumptions we might make about its ”behaviour”. So what do we actually know/don’t know about our data?

- High-dimensional with $p > n$.
- Unbalanced class responses.
- Some features are highly correlated.
- No domain specific knowledge. That is, we don’t have any idea of the process that generated the data. Which variables are relevant or not? Could all the variables be relevant or is the solution sparse?

Further, we have to consider the goals of modelling. We want to have high predictive strength and we want to find the features that actually have an impact on the response. With respect to this, we will proceed with the assumption of sparsity. That is, we are assuming that out of the 800 features, most of them are probably not significant.

The main reasons for choosing elastic net regularized regression are:

- We are betting on sparsity, and the elastic net procedure will set some coefficients to exactly zero.
- However, since we have features with high correlations, we want to adjust for this as well. That is, we want to avoid setting the ”wrong” coefficients to exactly zero and instead shrinking correlated variables towards each other and thus ”sharing” responsibility.
- As you might have noticed, we are aiming at striking a balance between LASSO regression and Ridge regression.

Another reason for choosing Elastic net over LASSO ($\alpha = 1$) and Ridge ($\alpha = 0$) is the fact that these two methods are special cases of the Elastic net. In our setting, we do not believe that we have sufficient knowledge about the data to be confident enough to choose either $\alpha = 1$ or $\alpha = 0$. Choosing a more flexible model seems like the best approach in this case and assuming that either LASSO or Ridge is in fact the best model for the data, then the elastic net will produce results that confirm this.

1.2.2 Random Forest

The next method that we will use to analyze the data is Random Forest (RF). Since it is not a model based classification method (in our case) and instead uses partitioning of the feature space, it is an interesting comparison to Elastic net.

So once again, with respect to what we do and do not know about our data, why could RF be a good approach?

- We don’t have to make any strong assumptions about the underlying structure of the data.
- Correlation among features is not a problem for the RF algorithm, at leat not when it comes to building trees. However, correlated features might be an issue when it comes to variable importance.

- We can use the variable importance produced by RF to possibly determine influential features.

1.3 Results

The data has been scaled and centered before performing the analysis.

1.3.1 Elastic net

The Elastic net algorithm has two hyperparameters, λ and α, that have to be chosen. Using the **train**-function we perform a 10-fold cross-validation on the training set $\boldsymbol{X}_1$ and $\boldsymbol{y}_1$ over $\alpha \in \{0, 1\}$ and a range of λ-values. The sequence will then choose the values of α and λ that lead to the best Accuracy of the cross-validated elastic net fits. The result of the cross-validation yielded $\alpha = 0.9$ and $\lambda = 0.019$.

From this, we can use the **predict**-function on the validation set $\boldsymbol{X}_2$ to predict the class labels based on our fitted model with $\alpha = 0.9$ and $\lambda = 0.019$. Comparing the obtained predictions with the vector of class labels $\boldsymbol{y}_2$ we obtain the confusion matrix in table 2.

		Actual	
		0	1
Pred	0	131	21
	1	3	20

Table 2: Confusion matrix for predictions from the Elastic net model.

Since we have a highly imbalanced dataset we need to take this into account when evaluating the performance of our predictions. For this purpose we can use the **Balanced Accuracy** defined as

$$A_B := \frac{1}{2}(A_+ + A_-)$$

where A_+ and A_- are the accuracies obtained for positive (0) and negative (1) outcomes. From the confusion matrix in table 2 we get the Balanced Accuracy

$$A_B = 0.7327.$$

Further, the Elastic net algorithm produces a sparse solution with a total of 72 (out of 800) non-zero coefficients. Table 3 shows the ten biggest positive and negative estimated coefficients.

	var	beta
1	233	-0.69
2	324	-0.55
3	620	-0.40
4	152	-0.38
5	397	-0.37
6	595	-0.35
7	667	-0.30
8	584	-0.27
9	643	-0.21
10	437	-0.20

	var	beta
1	289	0.48
2	34	0.35
3	588	0.34
4	534	0.33
5	347	0.31
6	740	0.28
7	583	0.18
8	472	0.16
9	515	0.15
10	674	0.15

Table 3: Ten biggest negative and positive coefficients estimated by the Elastic net algorithm.

1.3.2 Random Forest

The RF algorithm has a number of different parameters that could potentially influence the outcome. The main ones being the number of trees to grow, the number of variables $q < p$ to choose from at each split and the minimum leaf node size. The algorithm has been run several times with different (reasonable) variations of these values to find the combination that gives us the highest Balanced Accuracy.

Further, we can also directly deal with the problem of the unbalanced classes in our training data, since this can give us an estimate that is biased towards the most prevalent class. When applying the Rf algorithm on our training data and then predicting on the validation set, the classifier assigned every single observation (except for one) in the test set to 0. To deal with this problem, we can try to oversample from class 1 or undersample from class 0 (or both). This means that instead of bootstrap sampling n out of n observations with replacement to build each tree, we can sample a specified number of observations (with replacement) from each class.

Finally, the RF algorithm that gives us the best predictive strength in terms of Balanced Accuracy uses the following parameter settings:

- ntree = 600 (number of trees)
- mtry = 100 (number of variables at each split)
- min.node.size = 5 (minimum leaf node size)
- Sampling (with replacement) 66 observations from class 0 and 76 observations from class 1. Notice that 76 is the number of observations with class 1 in y_1.

		Actual	
		0	1
Pred	0	125	23
	1	9	18

Table 4: Confusion matrix for predictions from the RF algorithm.

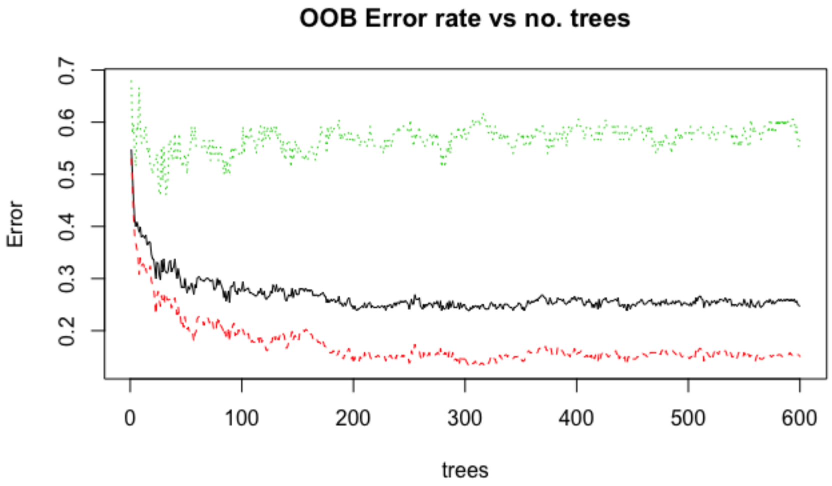

Figure 1: The figure shows the OOB (Out of bag) error rate vs number of trees. The black line is the overall OOB error. The red line is OOB Error for class "0" and green is for class "1".

The results of the predictions on the training set are presented in table 4 and we get the Balanced Accuracy

$$A_B = 0.6859.$$

Further, using the variable importance feature of the RF algorithm, we can get an idea of which variables are most important.

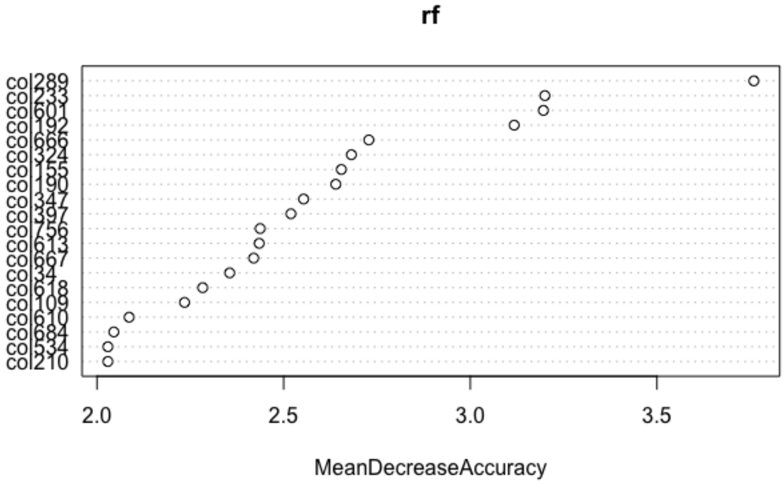

Figure 2: The figure shows the 20 features with the highest Mean Decrease Accuracy.

1.4 Discussion

When it comes to evaluating and comparing the performance of these two methods, we saw that the Elastic net achieved a Balanced Accuracy of 0.7327 while RF was slightly worse with 0.6859. The results are not impressive for either methods, although Elastic net still performs slightly better. However, considering the fact that our data is highly unbalanced, the results are not terrible, and at least better than chance. Further, a reasonable explanation as to why the Elastic net produces better predictions could be our assumption of sparsity. The RF algorithm does not make any such assumption, and instead uses the whole feature space to classify, and when we have a lot of features that are unimportant, this might make the Elastic net method more appropriate and thus leading to better predictions.

Further, when determining the best predictors there are a few things to keep in mind. Do we want to be able to interpret the data? Or, are we merely interested in discovering influential features, without caring about the magnitude of the effects? Elastic net produces interpretable results, it gives us a sparse solutions with coefficient estimates for the significant features. Thus, we have knowledge of which features have a significant effect on the outcome and the magnitude of these effects. RF however, is an algorithm that is more focused on predictability. We can get an idea of which features are influential from the variable importance, but we have no idea of how or to what degree these features influence the outcome. Another problem with the variable importance measure arises when we have correlated features. The RF algorithm can technically choose any of the correlated features as a predictor, and when one of them is used the importance of the other(s) will drop. Meaning that we might draw false conclusions that the chosen feature has a high impact on the outcome and the other(s) are not important. Elastic net on the other hand, solves the problem (or is supposed to at least) of multicollinearity that LASSO can not, leading to a higher confidence that the chosen (non zero coefficients) features are in fact the best predictors.