The ZipCode Dataset

Lúcia Moreira, Irene Azzini

March 18, 2019

Summary

A short study on the effect of expressiveness of models using different classification approaches was performed for the zipcode dataset in the package ElemStatLearn. The methods considered for studying its expressiveness were coded regression models (linear model, Lasso and Ridge fitting were also considered, and polynomial fitting up to degree 4) and classification models (kNN, LDA and QDA). Test accuracy was used as a fair quantitative figure between the methods. A binary approach for classes '2' and '3' was used and a non-binary approach for all the 10 (0-9 digits) classes was considered as well.

Lasso fitting reduced variables in the coded linear regression model for the 10 classes from 256 variables to 150 variables (more than 100 variables reduction) and increased test accurracy from 9% to 23%, Ridge performed in a similar way for improving test accuracy of the 10 classes coded regression problem. For the binary regression Lasso fitting reduced variables in about 140 items and increased accurracy from 54% to 96%, Ridge fitting also increased to 97% the accuracy showing the potency of improving the linear coded regression in a binary system with a very expressive number of variables. Performing a coded polynomial regression fitting up to degree 4 to the multiclass problem kept test accuracy at 9% level, alike in the binary approach that kept accuracy around 53%. The test confusion matrices for the coded polynomial fittings in the 10 classes problem showed a lot of strong misclassifications when using coded regression in the multiclass classification problem. Despite coded polynomial fitting being a good educational tool for learning the expressiveness of a model in the bias-variance trade-off, in this type of classification problem showed to be not relevant due to the low test accuracy.

Regarding the classification methods, kNN showed to be the best approach (for the 10 classes and binary poblem) due to its own classificative nature . Such method presented an accuracy of ca. 94.5% for k between 3 and 6 considering the classification of all all the 10 classes in the dataset despite the high dimension of the variables and presenting a higher accuracy than using LDA (88%) under the same conditions. In the binary approach, test accuracy of kNN is around 97% for K between 3 and 8 and LDA´s accuracy is not that behind. In the binary problem, the difference between LDA and QDA was minimum. QDA gave as problem of overfitting in the case of all the 10 classes so it was not possible to make a comparison between the 10 classes and binary problems.

Introduction [1]

It is well-known that a potential disadvantage of using a parametric approach (such as regression, LDA and QDA) for fitting data to a function is that the model chosen will usually not match the true unknown form of that function because of the huge assumption about the structure. If the chosen model is too far from the true function, then the estimate will be poor. An alternative is try to address this problem by choosing more flexible models that can fit many different possible functional forms. However, fitting a more flexible model requires estimating a greater number of parameters. These more complex models can lead to a phenomenon known as overfitting the data, which essentially means they follow the errors, or noise, too closely.

On the other hand, non-parametric methods (such as KNN) do not make explicit assumptions about a functional form. Instead they seek an estimate of the function that gets as close to the data points as possible without being too rough or wiggly. Such approaches can have a major advantage over parametric approaches: by avoiding the assumption of a particular functional form for the data, they have the potential to accurately fit a wider range of possible shapes for the same data. Moreover, the non-parametric approach completely avoids the danger of not fitting the data well, since no assumption about the form of the function is made.

However, a non-parametric approach suffers from a major disadvantage: since they do not reduce the problem of estimating a function to a small number of parameters, a very large number of observations is required and needed to be stored in order to obtain an accurate estimate for the function (a poor reduction capability).

Other point to take into consideration when deciding for a particular method is if we are dealing with a regression or a classification problem. Variables can be characterized as either quantitative or qualitative (also known as categorical). With a quantitative response we have regression problems, while qualitative responses are often referred to as classification problems. However, distinction is not always easy provided that any qualitative predictors are properly coded before the analysis is performed. In the present assignment, the response classes were coded using their mathematical value in the regression approach.

Additionally, we need some type of measure for how well model predictions actually match the data. Usually we choose a method that gives the lowest test error, as opposed to the lowest training error. The problem is that many statistical methods specifically minimize the training error. And for these methods, the training error can be indeed quite small, but the test error is often much larger. So in order to minimize the test error, the statistical learning method to be selected should simultaneously achieve low variance and low bias. This is when the Bias-Variance Trade-Off comes in! This is referred to as a trade-off because it is easy to obtain a method with extremely low bias but high variance (for instance, by drawing a curve that passes through every single training observation) or a method with very low variance but high bias (by fitting a horizontal line to the data). The challenge lies in finding a method for which both the variance and the squared bias are low. So the correct level of flexibility (expressiveness) is critical to the success of any statistical learning method. Below we discuss the bias-variance trade-off for the particular data set chosen for this assignment when using different statistical learning methods and draw some conclusions. The answer to the questions indicated in the assignent will be introduced when discussing each method as most as possible and if it applies.

Results and discussion

Regression Methods

Two coded regression methods were considered: linear and polynomial. Figure 1 shows the train and test error obtained after regression for the 2´s and 3´s data subset. Train errors are quite small for all the polynomial degrees (from linear up to 4), with train error decreasing with polynomial grade and consistent with the fact that as expressiveness of the model increases training error decreases. On the other hand, test errors are substantial for all the fittings (poor accuracy of the regression method, 53.57%), also showing that increasing expressiveness increases test error, probably due to a higher rate of the variance increase with expressiveness in comparison to bias and overfitting. Taking into account test error in a regression setting, linear fitting should be chosen once presented the lower test error. Lasso and Ridge fitting were also considered for improving the linear coded regression. Both methods increased accuracy for values higher than 95%, showing that indeed Lasso and Ridge linear regressions work quite well in a binary classification problem and in the presence a high number of variables.

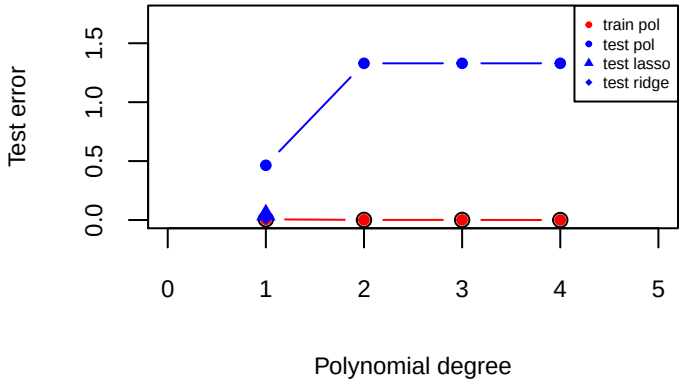

Figure 1 - Train and test error for different polynomial degrees for the 2´s and 3´s subset.

Methods	Test.error	Accuracy
Linear	0.46	53.57
Polyn 2	1.33	53.85
Polyn 3	1.33	53.85
Polyn 4	1.33	53.85
Lasso	0.04	95.60
Ridge	0.02	97.53

Table 1 - Test error and accuracy for different polynomial degrees for 2´s and 3´s subset.

Figure 2 shows the train and test error for all the 10 digits in the zipcode dataset. Nearly the same conclusions from the binary approach can be used for the non-binary one. The coded regression method gives a poor fitting to this classification problem (accuracy 9.12%). The Lasso fitting decreased test error from 26.6 to 3.8 showing the potency of the method for decreasing the number of variables while increasing accurracy (23.06%). Ridge fitting gave similar results: test error (3.78) and accuracy (23.12%).A coded regression method should not be chosen for this 10-class problem due to poor accuracy.

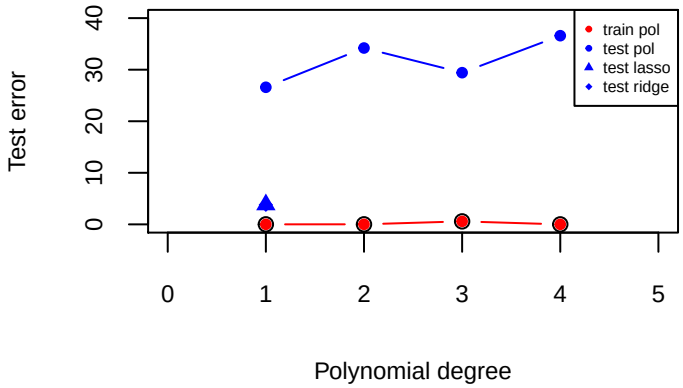

Figure 2 - Train and test error for different polynomial degrees for all the 10 classes.

Methods	Test.error	Accuracy
Linear	26.59	9.12
Polyn 2	34.20	9.32
Polyn 3	29.42	9.07
Polyn 4	36.59	8.82
Lasso	3.81	23.07
Ridge	3.79	23.22

Table 2 - Test error and accuracy for different polynomial degrees for all the 10 classes.

Classification methods

Three classification methods were used, also considering a binary approach and a non-binary approach, kNN, LDA and QDA. Figure 3 shows that train error of kNN for different values of k for the 2´s and 3´s data subset and of LDA and QDA. It is possible to see that train eror is very small and increases with k as expressiveness decreases. Test error remains also small although bigger than train error and increases as expressiveness decreases. The range of k values between 3 and 8 is the one that we can consider optimal, because it has small value for train and test errors. The optimal k is k=5. LDA is a litte bit worse than kNN for the range of k that we consider optimal, indeed accuracy for kNN with k=5 is about 96.97% while for LDA is 96.1%; QDA is in the same order as LDA.

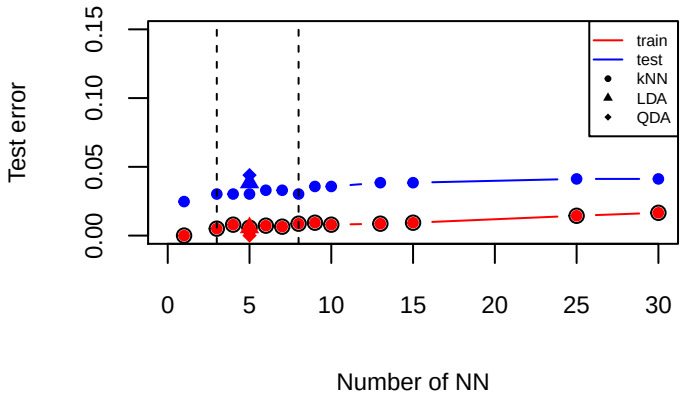

Figure 3 - kNN test and train errors for differente k values, LDA and QDA for classes 2 and 3.

Methods	kNN	kNN	kNN	kNN	kNN	kNN	LDA	QDA
k	3	4	5	6	7	8	-	-
Test.error	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04
Accuracy	96.98	96.98	96.98	96.70	96.70	96.98	96.15	95.60

Table 3 - Test error and accuracy for kNN, LDA and QDA for classes 2 and 3.

Figure 4 shows the train and test error of kNN and LDA for 10-class problem in the zipcode dataset. We can draw the same conclusions as in the binary approach. kNN is the best method for k values between 3 and 6, optimal for k equal to 6; indeed accuracy of kNN with k=6 is 94.27% while for LDA is 88.54%. QDA was not considered due to the huge number of parameters to be estimated in the 256 variables problems and for this case the small dataset available.

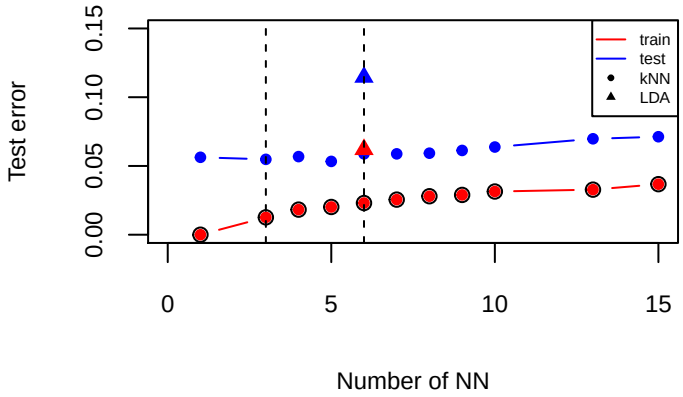

Figure 4 - kNN test and train errors for k values for all the 10 classes.

	3	4	5	6	
Methods	kNN	kNN	kNN	kNN	LDA
k	3	4	5	6	-
Test.error	0.05	0.06	0.05	0.06	0.11
Accuracy	94.52	94.32	94.67	94.12	88.54

Table 4 - Test error and accuracy for kNN and LDA for all the 10 classes.

Conclusions

In order to have a complete view of the results obtained, we provide in figure 5 the results obtained with all types of models for classes 2 and 3 (left) and for all the 10 classes (right). We do not report the value of test error in linear regression for the too large values of this. For the binary approach, coded linear regression performed better with Lasso fitting showing an accuracy of 96% and for the Ridge fitting 97% accuracy. For the classification approach, test accuracy of kNN is also around 97% for k between 3 and 8 and similar accuracy for LDA and QDA. In such setting, a linear decision boundary is most probably present once logistic linear regression and LDA give similar results. QDA performed very slightly worse than LDA, since it fit a more flexible classifier than necessary while the KNN method also gave very good results as well. For the multiclass problem, the coded regression approach used by the authors failed to find accurate decision boundaries, however a multiclass logistic regression could have been performed using a multinomial logistic regression based functions available in R but the authors are not so far familiar with multinomial logistic regressions besides running out of time to perform this comparison. Despite of that, the glm() function from the kernlab library was used in the binary problem and it gave exactly the same results as the linear regression. Regarding the classification methods, k-NN showed to be the best approach presenting an accuracy of ca. 94.5% for k between 3 and 6 despite the high dimension of variables and higher than LDA accuracy (88%) under the same conditions. In summary, in the multiclass dataset, the much more flexible KNN method gave the best results because no assumptions are made about the shape of the decision boundary. Therefore, maybe in this multiclass problem, the decision boundary is non-linear. It was not possible to use QDA in this multiclass problem.

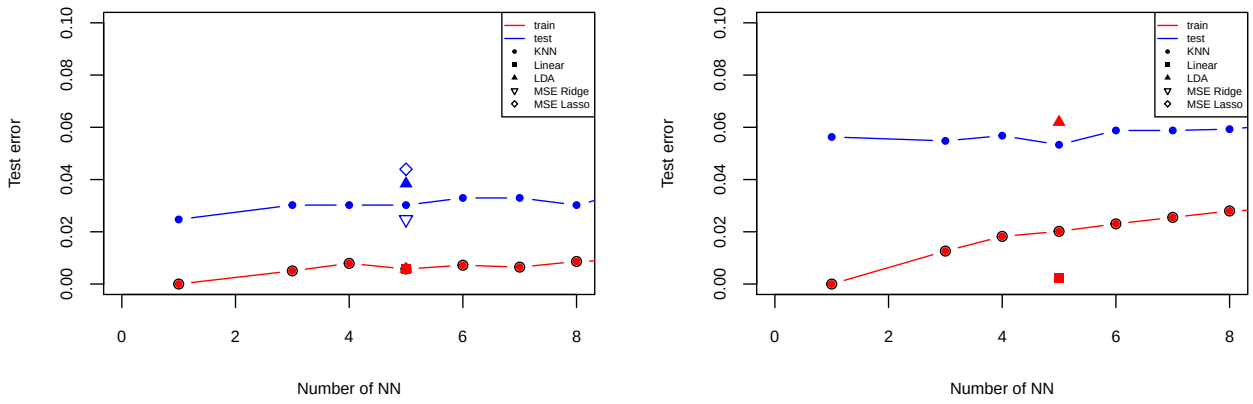

Figure 5 - Confrontation plot for classes 2 and 3 (left) and all classes (right)

References [1]- Very short summary from Chapter 2 in “An Introduction to Statistical Learning”, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. Springer, New York 2013.