Technical University of Denmark

02450 Introduction to Machine Learning and Data Modelling

Assignment 2

Classification and Regression

Poul Kjeldager Sørensen
s093294

Martin Kasban Tange
s093280

April 9th 2013

Contents

1 Introduction

This report is made in regard to the course 02450 Introduction to Machine Learning and Data Modelling at DTU (Technical University of Denmark). In this second assignment we are to ...

All the work carried out in this report are done by Tange, M. K. and Sørensen, P. K. and the code can be found on github[1]. Do note that the code have only been used to generate plots and illustrations for the report and some playing around. There have been done no effort into readability or re-usability of the code.

1.1 Data

The same data as from assignment one have been used, a feature representation of the MNIST dataset. The representation consist of 272 features calculated from vertical, horizontal and radial histograms together with two profiles; in-out, out-in. Refer the first assignments for the full description.

[1] https://github.com/mktange/IntroMachineLearning

2 Regression

2.1 Problem

Since there is no obvious regression problem for our data set, we need to manufacture one ourselves. The way we have chosen to accomplish this is to make an attempt at predicting *one* of the features from the all the *other* features. This means that we extract one of the features from our feature set and make this our resulting `y` vector.

To keep it simple we will only look within a single class which in this report is the one for the digit 4 (chosen at random).

2.2 Forward selection

Doing forward selection between all remaining 271 features will take a tremendous amount of time. Therefore we choose to only look within a certain part of the attributes (horizontal/vertical/radial histograms or in-out/out-in profiles), and limit it to these.

For our tests in this report we chose to stick to the in-out section of our features, which brings the feature count down to 72, which is manageable. From these we chose different features to try to predict. Here are two of these cases:

Predicting feature 'In-out 50'

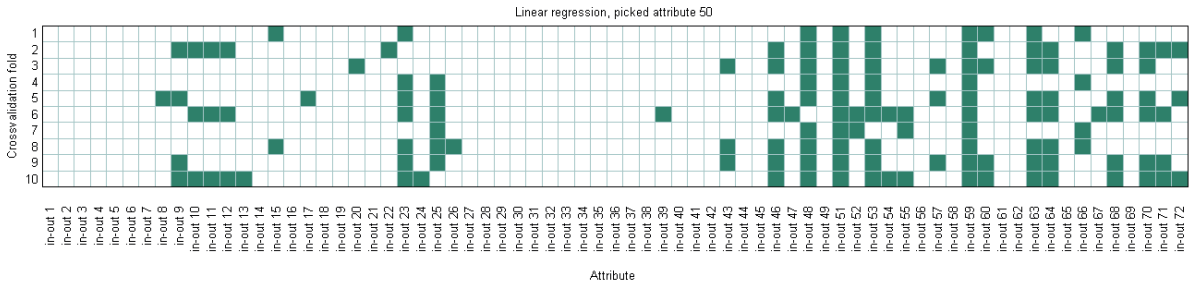

Figure 2.1: Linear regression with forward feature selection when for predicting feature `'In-out 50'`. The feature selection for each of the cross-validation is shown. Not many features are selected in order to achieve the best outcome in each cross-validation.

From this we can see that it tends to be features near the extracted feature (no. 50), which makes sense since they are usually quite correlated (see previous report), when dealing within the same section.

The resulting squared error for this feature, both with and without the feature selection:

```
Linear regression without feature selection:
- Training error:     0.05
- Test error:         0.06
- R^2 train:         0.95
- R^2 test:         0.94
Linear regression with sequential feature selection:
- Training error:     0.06
- Test error:         0.06
- R^2 train:         0.95
- R^2 test:         0.94
```

Quite decent error rate when trying to predict the feature, which means that it would be viable to predict it with this approach-

Predicting feature 'In-out 31'

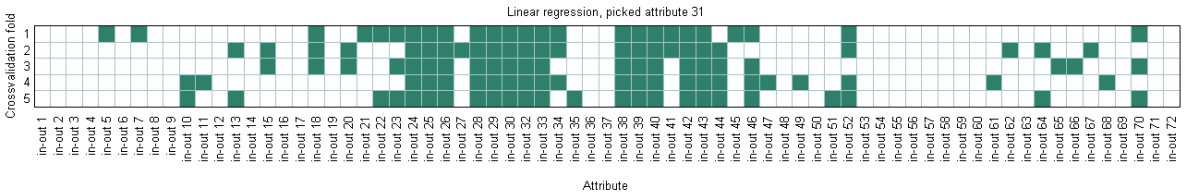

Figure 2.2: Linear regression with forward feature selection when for predicting feature `'In-out 31'`. The feature selection for each of the cross-validation is shown. More features needed when trying to predict this certain feature compared to attribute 50 from before.

Again we can see that the features close to the picked one (no. 31) are often used in the feature selection. A lot more features has been selected for each cross-validation, and this explains why it was much slower at spitting out our result.

The resulting squared error for this feature, both with and without the feature selection:

```
Linear regression without feature selection:
- Training error:     0.54
- Test error:         0.57
- R^2 train:         0.90
- R^2 test:         0.89
Linear regression with sequential feature selection:
- Training error:     0.55
- Test error:         0.57
- R^2 train:         0.90
- R^2 test:         0.89
```

This time it is much worse, and it does *not* seem very viable to predict this certain feature with basic linear regression.

Overall

Some features are quite possible to predict, while others are much more difficult using only linear regression. Predicting a single feature with *all* of our original 272 features might be better these cases, but we are not really able to test this because of the immense computation-time for this report.

2.3 New data observation

As noted in the two examples above, the features which have something to say about the missing feature tend to be close to it. This makes sense since we deal in pixel histograms and profiles, which tend to not vary much from one feature to the ones right next to it – it is usually a more gradual change in each way (see previous report with all features shown side-by-side for some observations).

2.4 Artificial Neural Network

For both our examples we fitted an ANN models to the same data. We fitted all of our ANN models with just 10 hidden units, did 5-fold cross-validation and averaged their error rates.

In-out 50

The average error rate for the `'In-out 50'`:

```
Error rate: 0.012
```

It already had a good error rate using basic linear regression, but it got even better with ANN from 0.06 to 0.012.

In-out 31

For `'In-out 31'`:

```
Error rate: 0.035
```

Which greatly improves the error rate compared to the linear regression. From a huge 0.57 error rate on the test data, to only 0.035!

3 Classification

3.1 Problem

We choose to solve the obvious classification problem for the MNIST data set, which is to classify the digit of each observation.

3.2 Classification techniques

There have been looked into a few different methods for classification and by means of paired t-tests it have been found if one method is significant different from the others.

3.2.1 Decision Trees

Since we have a lot of attributes, which do not hold any significant meaning besides an seemingly arbitrary integer value, the decision tree solution is likely to not be very precise in determining the class. It will also be quite a bit tree since it has to value in each of the different attributes, and determine what the meaning of that specific attribute might be in the classification.

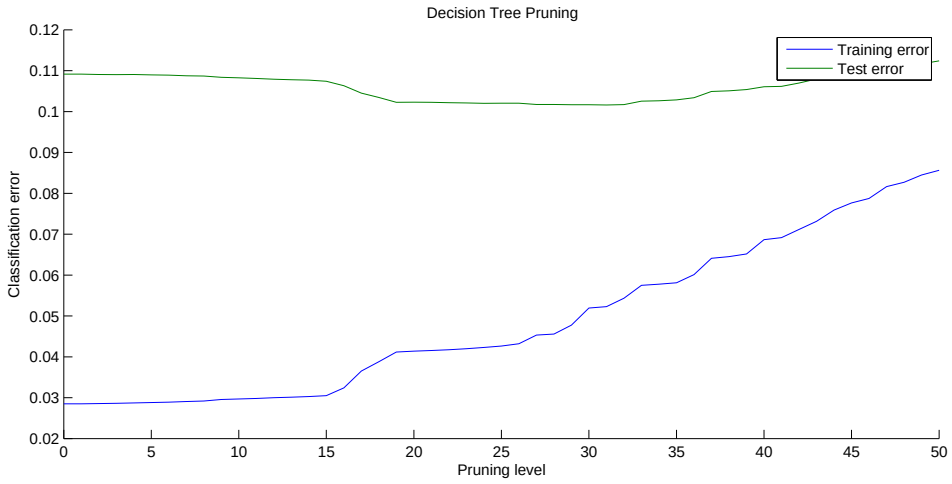

Figure 3.1: test

3.2.2 K-Nearest Neighbors (KNN)

As the computation demand of KNN is significant larger then the other two, we first estimated parameters for a subset of the data and then did the same again for a smaller subset of the parameters. This way we was able to find the optimal parameter in feasible time.

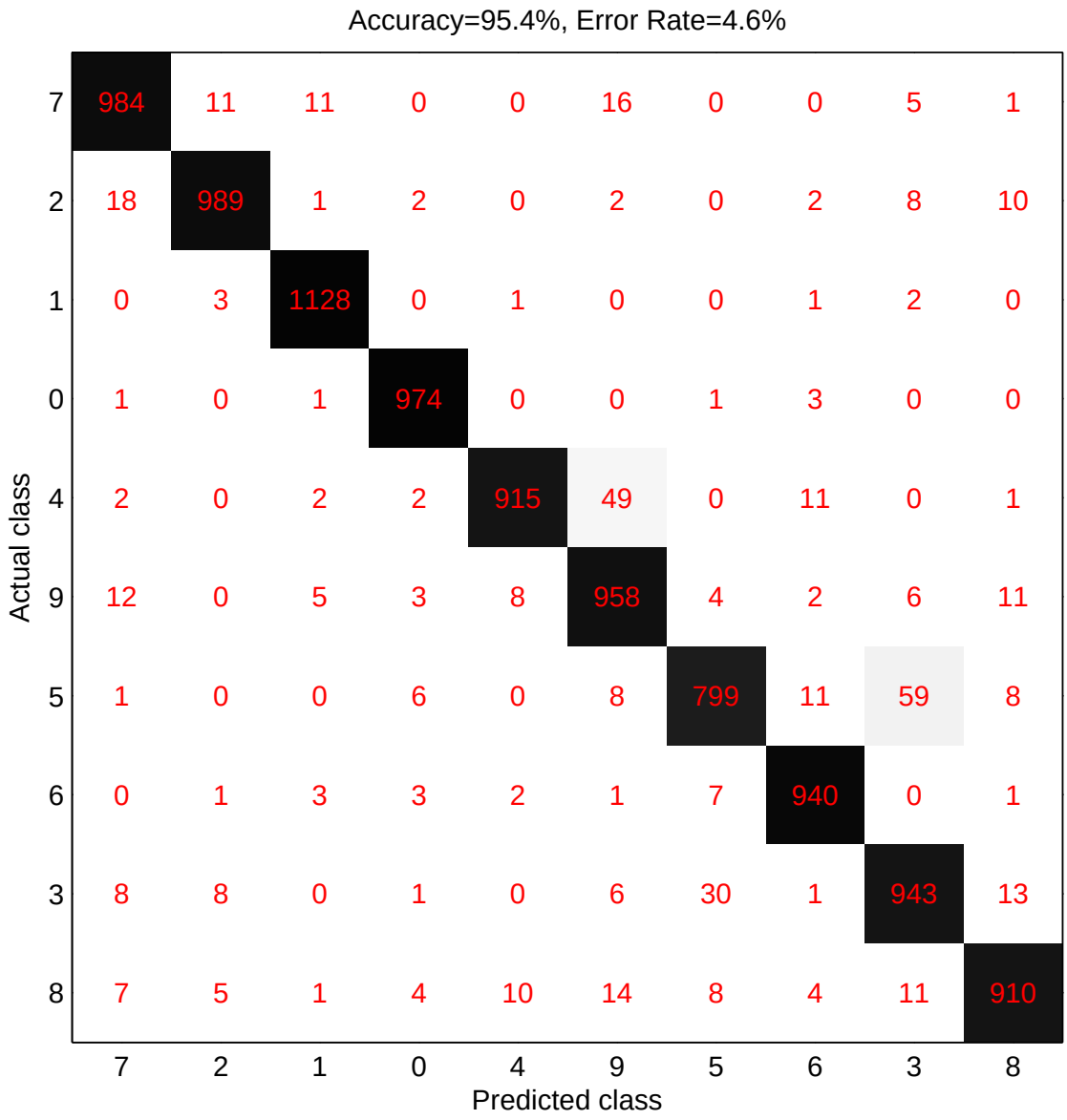

Figure 3.2: The confusion matrix for KNN classifications with 5 neighbors. It can be seen how 3 and 5 get confused the most.

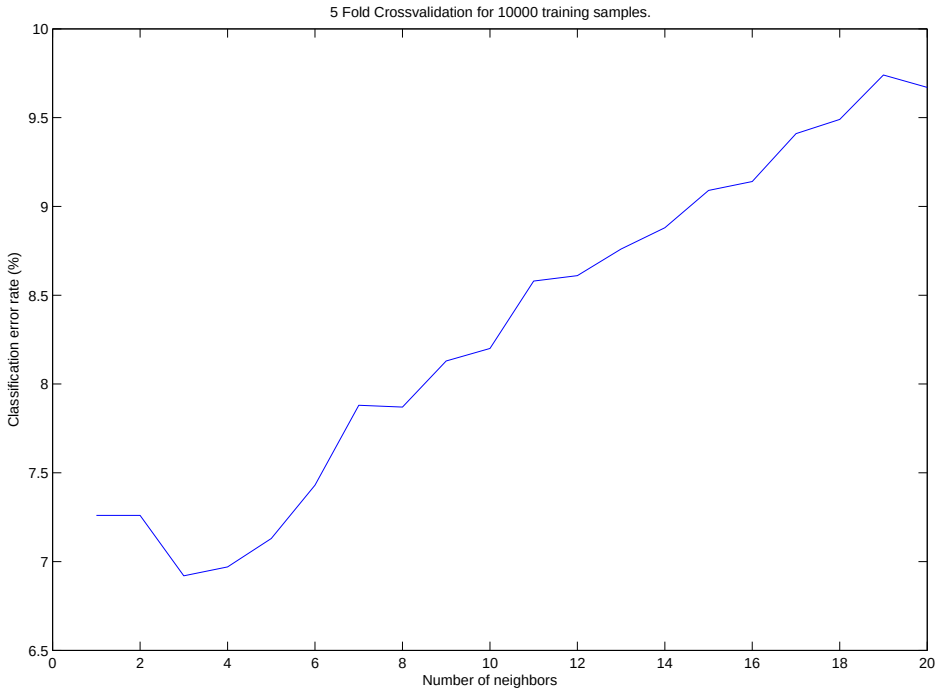

Figure 3.3: The five fold cross validation for 10000 samples.

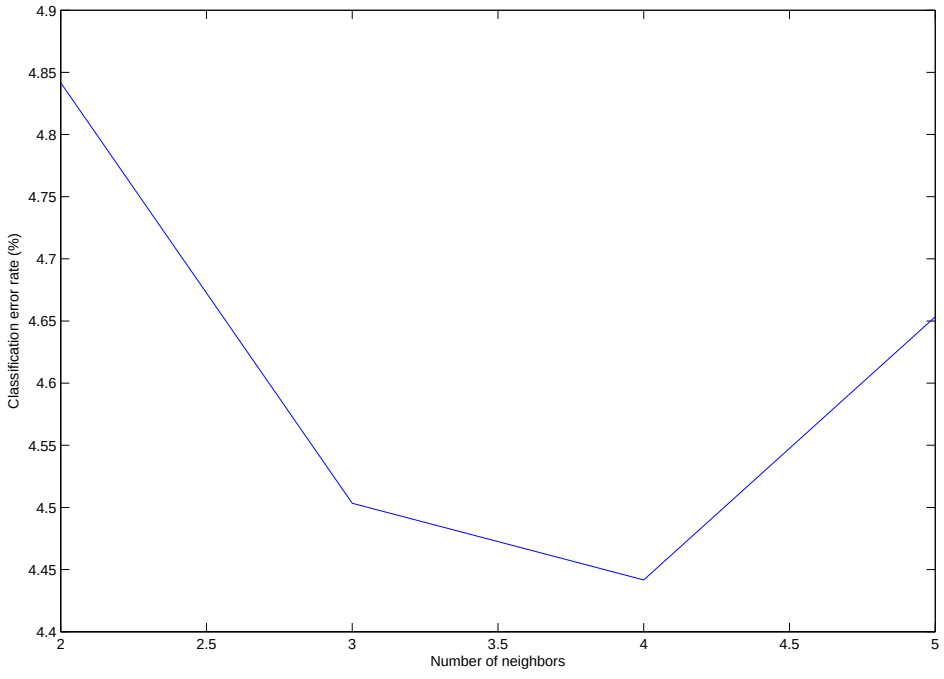

Figure 3.4: The cross validation for the full dataset.

3.2.3 Naïve Bayes

We fitted a Naive Bayes model based on the exercise in doing so. It was quite fast but also had a average error rate of 15 % in a 5 fold cross validation.

3.3 New data observation

For some models it is more trivial to explain what is going on behind the scene, but with this dataset of 272 features some of the simplicity fades away. Using decision trees it is trivial to by hand to take a new sample and move your way down the tree until a leaf is reached. At each split one could look at the distributions of samples in the two subtrees and tell what the meaning of the feature is. We argue that because of our data size it is not feasible to manually move one observation down the tree even though it being trivial.

For neural networks we could also manually compute the activation of each neuron and so forth. This could be done efficiently with matrix multiplication but because of the data size it would take forever.

While understanding the theory and how a model classify a observation, it makes less sense in our case and we have put our faith into the toolbox.

3.4 Performance comparison

In Table 3.1 the paired t-test values can be seen. As none of them span across zero they are all significant different.

KNN vs Trees	[-0.0604 -0.0540]
KNN vs NB	[-0.1217 -0.1173]
NB vs Trees	[-0.0657 -0.0589]

Table 3.1: Paired T-Test

4 Discussion

4.0.1 Regression

In this assignment we had some difficulties with the topic of regression. All the features are arbitrary numbers and do not make much sense independent of the others. We choose to try out forward selection but found out that the dataset of 60000 samples and 272 features took a lot of time. Instead we decided to only take out one of the feature groupings, the In-Out profile and also only to look at one number. We picked digit 4.

Recall from assignment one we saw that features was highly correlated to the neighbor features. The was collected by steps of 5 degree and the distance from center to the first on pixel.

Not surprising this means that the forward selection tend to going forward until it finds one of these neighbors. If we should do this again, we would try to detect one feature from one grouping based on the another groping.

4.0.2 Classification

Classification is the task that extensively have been carried out by other scientists. One of the newest records on the dataset are below 40 errors of the 10000 test set. These results was found by deep neural networks and by generating random transformation and rotation of the data for each epocs. The architecture for these nets can be up to 5 levels and have over 1000 hidden units in some layers. Inspired by this we knew that we wouldn't beat that with our simpler models.

We found out that just fitting some of the simple neural networks on our data took extensive amounts of time and we refined our goals to learning how to evaluate the models vs each other. We trained a model based on Naive Bayes, Decision Trees and KNN. Finding that the KNN was the slowest of these. This also makes sense as it simplify the distribution of objects in parameter space, and we have to evaluate a new observation to all test objects.

We found the results of the KNN parameter estimation counter intuitive. We used cross validation to find the optimal parameter for number of neighbors, and to our surprise it quite fast started to get worse when more neighbors was added. As the method being computation demanding on the 60k samples, we sampled 10000 observations uniform and ran the cross validation over a larger span of the parameter value for how many neighbors and then did it again on the full data set for a smaller range. We managed to find a optimal parameter of 4.

4.0.3 Performance

We found the optimal parameter space for each method in a inner cross validation step and then did an outer five fold cross validation for these parameters and by paired t-tests we evaluated if one model was significant worse then the others. Technically we evaluate if they are significant differently, and if so one is worse then the other.

We found that they all performed significantly different.

5 Conclusion

We have in this assignment played around with some interesting models. We are not fully satisfied with the result of the report as we ended up with less illustrations then wanted. Also we have found out that our choose in dataset could have been better.

Lets elaborate a little on that. We first picked the dataset as we think it was nice to have a problem that we could relate to from our software technology bachelors. We also think it was nice that a lot of work had already been done and we could compare our work with this. We have solved the problem somewhat decent, but we still have a feeling that some of the questions raised in the assignments are not that easy to address on this dataset. One example is the regression stuff, but another one is also the fact that our features are arbitrary numbers. We now see how it could be beneficial to have a dataset like those we use in the exercises, as its easier to illustrate and explain how the models work.

We think we did a decent assignment with room for some improvements, and we will see if we can make up for it with the 3th report.