Machine Learning SS2013

Ulrike von Luxburg

Assignment 01

Arne Schröder, Falk Oswald, Angel Bakardzhiev

April 17, 2013

Matlab Implementation

First, we introduce and briefly describe our M files, included in the attached zip file.

- **knnClassifySingle.m** - function, that uses k-nearest neighbours method to predict label of single datum
- **knnClassify.m** - function, that uses k-nearest neighbours method to predict labels
- **evaluateK.m** - evaluates knnClassify for different k-values and returns the minimal k
- **loss01.m** - Gets as input a prediction calculated by the knnClasifiy and correct labels y. The function returns the average error (empirical risk with respect to the 0-1 loss) for this prediction.
- **drawNumber.m** - visualize a number using *imagesc*
- **doExercise1.m** - loads all training and test data for exercise 1, calls knnClassify and plots the result
- **doExercise2.m** - loads all training and test data for exercise 2, calls knnClassify and plots the result
- **Assignment01.m** - the main script, calls doExercise1 and doExercise2 with different parameters

Questions

1.7. Plot the training and the test errors. Do results change between different runs? Why?

Yes, the results change between different runs. The reason is, that we use random training and test data. For each run the data is different, so we get different results.

1.9. More training examples. How does the performance of kNN classifier change?

The performance of the classifier is the same like before for the test data, increases however approximately by factor 10 for the training data.

1.10. Unbalanced classes. How does the performance of kNN classifier change?

The error of the classifier increases approximately by factor 1/2 for the training data and by factor 40 for the test data.

2.5. Run your algorithm to classify digit 3 from 8 and compare its performance with results from digit 2 versus 3.

The results from both runs are similar in general. Over all the classification between 3 and 8 has errors higher by factor 3.

Exercise 5

Question 1

What are false positive, false negative and average error of you(r) calssifier?

Answer

To answer this question we simply need to fill out the following table:

	spam	not spam	overall
classified as spam			
classified as not spam			
overall	60 %		

We know that 85 % of spam is classified as such, which gives us that $60\% \cdot 85\% = 51\%$ of mails are spam and are classified as spam.

As 60 % of all mail is spam, 40 % of mail is not. This means that if 5 % of all non-spam mails are classified as spam this is 2 % of all mails.

	spam	not spam	overall
classified as spam	51 %	2 %	
classified as not spam			
overall	60 %	40 %	

Subtraction now tells us that 9 % of mails are spam but not classified spam and 38 % are spam and correctly classified.

	spam	not spam	overall
classified as spam	51 %	2 %	53 %
classified as not spam	9 %	38 %	47 %
overall	60 %	40 %	100 %

Therefore, the false positive rate is 2 % and the false negative rate is 9 %.

Task 2

Find a classification algorithm with false positive rate 0. Find a classification algorithm with false negative rate 0.

Solution

An algorithm with false positive rate of 0 can be achieved by not classifying anything as spam (resulting in a 40 % false negative rate).

Conversely, an algorithm with false negative rate of 0 can be achieved by classifying everything as spam (resulting in a 60 % false positive rate).

Question 3

Which entries of X' lead to a false positive error and which ones to a false negative error in the Bayes classification?

Answer

The false positives are those classified as 2 but being 1, in this case this applies to 2.5. The false negatives are those classified as 1 but being 2, in this case this applies to 2.

Task 4

Sketch the approximate Bayesian decision boundary by hand with respect to the following loss functions

- 0-1 loss
- Unsymmetric los(s): $\ell(\text{spam}, \text{non-spam}) = 1$, $\ell(\text{non-spam}, \text{spam}) = 100$.

Solution

The first is the same as if there was no loss function. In the following graph, one can see a with solid lines the decision curves for none or a 0-1 loss function and with dashed lines are the decision curves for the asymmetric loss. Magenta being in favour for 1 (non-spam) and black being in favour for 2 (spam):