Pattern Recognition: Assignment 2

Due on Monday, April 30 2012, 14:00

Prof. Fred Hamprecht, Summer Term 2012

patternrecognition@hci.iwr.uni-heidelberg.de

`http://hci.iwr.uni-heidelberg.de/MIP/Teaching/pr/`

Quadratic Discriminant Analysis (QDA)

This time, we are focusing on implementing and evaluating quadratic discriminant analysis (QDA). We will work on data with just two features. That way we can plot the feature space and gain visual insight in the workings of QDA.

Data Description

Suppose you are a lecturer and you want to predict if a student will pass the final oral exam of your class based on the average score of all exercise sheets and the score of the midterm exam (both in percentage points). You have historical data from previous terms, that you will use to train a QDA classifier. From each former student you know the two scores valued between 0 and 100 and the outcome of the final exam–passed or not-passed coded as 1 and 0.
The data is saved in `student-scores.mat` in two matrices 'training' and 'test'.

Prob. 1: Implementing QDA

(a) Training (4 points)

As a first step, implement a matlab function
`[mu0, mu1, covmat0, covmat1, p0, p1] = compute_qda(trainingy,trainingx)`
that accepts a 1 x n1 vector trainingy of input labels (that must be either 0 or 1) and a p x n1 matrix trainingx of features. The output should contain the p x 1 mean vectors, the p x p covariance matrices and the priors as scalars.

(b) Prediction (4 points)

Now, using the output from the previous section implement a function
`[qda_prediction] = perform_qda(mu0, mu1, covmat0, covmat1, p0,p1, testx)`
which predicts the labels of a p x n2 dataset testx using QDA.

Prob. 2: Applying QDA to the student scores dataset

(a) Training (4 points)

Apply the QDA prediction to the training data and compute the correct classification rate and visualize the results: A good way is to make a 2D plot of the data with regard to the two scores (e. g. using o in the plot comment), mark the class (0,1) with different colors and also add the QDA estimate (0,1) using a different symbol (x instead o) and different colors. (Suggested commands: hold, plot(A(1,:),A(2,:),og))

(b) Visualize the decision boundary (5 points)

It can be quite insightful to obtain an image of the decision boundary. Therefore, create a 100x100 grid of input values for the two scores (each running from 1 to 100) and perform the QDA for each combination of inputs. Make a plot to visualize the decision boundary (suggested command: imagesc). Using this plot and your results from the previous section inspect the quality of the QDA results on the training data. You should observe some misclassifications. Where do misclassifications on training data using QDA stem from (compared for example with kNN, that doesn't have any misclassifications on training data)?

(c) Prediction (3 points)

To get an idea of the generalization of the results, apply the QDA to the test data and compute the correct classification rate. Make a similar 2D plot as before.

Prob. 3 (Bonus): Compare with k-nearest neighbors (6 points)

Apply the k-nearest neighbors approach we implemented in assignment 1 on the training and test data. Plot the decision boundaries and classification results. Give a short comparison of kNN and QDA on this data.

Regulations

Please hand in the matlab code, figures and explanations (describing clearly which belongs to which). Non-trivial sections of your code should be explained with short comments, and variables should have self-explanatory names. Plots should have informative axis labels, legends and captions. Please enclose everything into a single PDF document (e.g. use the publish command of MATLAB for creating a LaTeX document and run latex, dvips and ps2pdf or copy and paste everything into an office document and convert to PDF). Please email the PDF to patternrecognition@hci.iwr.uni-heidelberg.de before the deadline specified below. You may hand in the exercises in teams of two people, which must be clearly named on the solution sheet (one email is sufficient). Discussions between different teams about the exercises are encouraged, but the code must not be copied verbatim (the same holds for any implementations which may be available on the WWW). Please respect particularly this rule, otherwise we cannot give you a passing grade. Solutions are due by email at the beginning of the next lecture (April 30, 14:00).