Pattern Recognition: Assignment 3

Due on Monday, May 7 2012, 14:00

Prof. Fred Hamprecht, Summer Term 2012

patternrecognition@hci.iwr.uni-heidelberg.de

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

Principal Component Analysis (PCA)

For this assignment you will apply a principal component analysis (PCA) to the handwritten digits dataset from the first assignment. You will implement a image compression using PCs. Furthermore, you will use them as a preprocessing step for knn classification to reduce the dimensionality of the feature space.

Data Description

In this assignment we again work with the mnist-digits dataset from the first assignment.

Prob. 1: Implementing PCA

(a) Implementation (6 points)

Write a function [lambdas, variance, pc] = pca(data). The input data is a p x n matrix of p dimensions and n samples. pc contains the principal components in its columns and is p x p. variance contains the variance associated with each principal component and is a p x 1 vector and lambdas is a pxn matrix, containing the weights associated with the principal components for each data column.
Using singular value decomposition (SVD) is the easiest way to implement PCA (suggested command: svd).

(b) Investigating the Principal Components (2 points)

Compute PCA on the 'training' digits dataset. Plot the first six PCs. How many of the 784 components are necessary to explain at least 60% of the overall variance, how many for at least 99%? (suggest command:

cumsum)

Prob. 2: PCA for Image Compression

(a) Compress with generic PCs (6 points)

Write a function `displayProjected( pc, data, lambda, n_comps )`, that displays the digit in data (768x1) as a linear combination of n_comps principal components (pc - the PCs, lambda - the lambdas of the data array).
Do a PCA on the training dataset and plot digit #3080 in the training dataset using all PCs and the PCs corresponding to 99% resp. 60% of the total variance.

Figure 1: Digit #3080 in the training dataset.

(b) Compress with optimized PCs (2 points)

This time do a PCA on training data depicting the digit five only. Plot the same digit as in the previous exercise with the same compression levels. Comment on the result.

Prob. 3: PCA as Preprocessing for Classification (4 points)

We provide the knn classifier from the first assignment in the matlab function `[ rate ] = knn( training, training_label, test, test_label, digit, k)`. Call the function with digit=5 and k=7; you should observe a correct classification rate (ccr) of 0.9930. Do PCA on the training dataset and use the lambdas as new features obtained from the original training and test dataset (Note: don't do a separate PCA on the test data, but project the test data using the training data PCs). How high is the ccr for digit 5 using all, 300, 100, and 10 PCs? Compare with the performance on all, 300, 100, and 10 of the original features (take the first n features in the data arrays). (Note: a ccr of 0.9100 can be obtained by classifying everything as non-five and is as good as random guessing).

Prob. 4 (Bonus): Eigenfaces (6 points)

Take pictures of yourself or a friend with different facial expressions. Downsample the images to 100x100 pixels and convert them to grayscale. Apply PCA to the pictures and plot the first six principal components.

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.