Exercises for
Pattern Recognition
Peter Fischer, Shiyang Hu
Assignment 4, 11/14.11.2014

SOLUTION

General Information:

Exercises (1 SWS):	Tue 12:15 – 13:45 (0.154-115) and Fri 08:15 – 09:45 (0.151-115)
Certificate:	Oral exam at the end of the semester
Contact:	peter.fischer@fau.de, shiyang.hu@fau.de

Feature Transforms

Exercise 1 In this exercise we will refresh your knowledge of the Singular Value Decomposition (SVD).

(a) What is the relationship between the SVD of a square matrix A and A^T.
$A = USV^T$
$A^T = \left(USV^T\right)^T = VSU^T$
(same singular values, roles of U and V interchanged)

(b) What is the relationship between the SVD of A and AA^T.
$A = USV^T$
$AA^T = USV^T\left(USV^T\right)^T = USV^TVS^TU^T = US^2U^T$

(c) Find a relationship between the singular values and the eigenvalues of a matrix $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{A}^{\boldsymbol{T}}$.
$AA^T\boldsymbol{v} = \lambda\boldsymbol{v}$
$US^2U^T\boldsymbol{v} = \lambda\boldsymbol{v}$
$S^2U^T\boldsymbol{v} = \lambda U^T\boldsymbol{v}$
$S^2\boldsymbol{w} = \lambda\boldsymbol{w}$
S is diagonal matrix with singular values, hence $S^2 = diag(\sigma_i^2) = \sum\limits_i \sigma_i^2 e_i$.
The eigenvalues of B are exactly the squares of the singular values of B.

Exercise 2 Linear discriminant analysis (LDA) is used to transform features such that two classes can be discriminated by a linear decision boundary. Use LDA for classification in the MATLAB Classification toolbox.

(a) Compute the LDA feature transform $\phi(\boldsymbol{x})$ during the training phase.
See the solution in LDA.m

(b) In the classification step, use the following decision rule:

$$y^* = \underset{y}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{\mu}_y)\|^2 - \log(p_y) \right\}$$

(c) What is the relationship between LDA classification and nearest neighbor classification?

For equal priors, LDA is equivalent to nearest neighbor classification with the class centroids (mean value of all the samples from this class). As a distance function for nearest neighbor classification, the Mahalanobis distance is used.

Exercise 3 The excersise addresses the Principal Component Analysis (PCA) for dimensionality reduction. On the course website you can find a short Matlab script to create a set of random points in 3-space. Your goal is to find a linear projection into 2-space, such that the original points can be reconstructed with minimal error.

(a) Compute the pricipal component of your data, i.e. the unit vector $\boldsymbol{w}$ such that the variance in its 1D subspace is maximized.
Input: Collection of D-dimensional features $X^* = (\boldsymbol{x}_0^* ... \boldsymbol{x}_1^*) \in \mathbb{R}^{D\times n}$ (no class memberships required)

Output: Transformation $Y = W^T X$, where W is an orthogonal matrix and $X = (\boldsymbol{x}_0^* - \boldsymbol{\mu}, \ldots, \boldsymbol{x}_n^* - \boldsymbol{\mu})$ are the de-meaned features. The vector components of Y are sorted by decreasing variance and w_1 pointing in the direction of the most variance of the data (principal axis).

Note: Zero-mean data can be enforced simply by de-meaning the data before PCA and adding the mean after PCA.

Derivation: Goal: Project data into a 1D space in direction of a unit vector $\boldsymbol{w}$, such that the variance of the projected data is maximal.

Mean and Variance of projected data:

$$\mu_1 = \frac{1}{n}\sum_i \boldsymbol{w}^T \boldsymbol{x_i} = 0$$

and the variance (note that Σ_1 denotes a scalar!)

$$\Sigma_1 = \frac{1}{n-1}\sum_i \left(\boldsymbol{w}^T \boldsymbol{x_i} - \underbrace{\boldsymbol{w}^T \mu}_{\text{zero mean}} \right)^2 = \frac{1}{n-1}\sum_i \boldsymbol{w}^T \boldsymbol{x_i} \boldsymbol{x}_i^T \boldsymbol{w} = \boldsymbol{w}^T \Sigma \boldsymbol{w}$$

where Σ is the covariance matrix of the data

$$\Sigma = \frac{1}{n-1}\sum \boldsymbol{x_i}\boldsymbol{x}_i^T = \frac{1}{n-1} X X^T$$

Conditional maximization given $\boldsymbol{w}^T \boldsymbol{w} = 1$ using Lagrange multiplier.

$$\boldsymbol{w_1} = \underset{\boldsymbol{w}}{argmax}\, \boldsymbol{w}^T \Sigma \boldsymbol{w} + \lambda(1 - \boldsymbol{w}^T \boldsymbol{w})$$

Derivative wrt. $\boldsymbol{w}$ shall be zero

$$2\Sigma \boldsymbol{w} - 2\lambda \boldsymbol{w} \overset{!}{=} 0 \iff \Sigma \boldsymbol{w} \overset{!}{=} \lambda \boldsymbol{w}$$

λ must be an eigenvalue of Σ and since we want to maximize this quantity, $\boldsymbol{w_1}$ is the Eigenvector with the largest Eigenvalue of Σ.

Using SVD of X and the relationship between eigenvalues of A and singular values of $B = AA^T$:

$$\Sigma = \frac{1}{n-1} X X^T = \frac{1}{n-1} U D^2 U^T$$

we can see that the solution is the first column of U.

(b) Implement PCA to reduce the feature space to $d = 2$ using Singular Value Decomposition (SVD). Hint: De-mean the data.

(c) Visualize the reduced features in 2D.

(d) Reproject the reduced features into the original space and compute the mean absolute error.
PCAexample.m