Math 521 Homework 4

Due on Tuesday, April 17, 2018

Kristin Holmbeck

Contents

List of Figures

Theory

1. Eigen-relationship

Consider the two eigenvector problems

$$C_x \boldsymbol{u} = \lambda_x \boldsymbol{u}$$

and

$$C_s \boldsymbol{v} = \lambda_s \boldsymbol{v}$$

where the matrices are related by $C_x = C_s + \alpha I$, where α is a real number and I is the usual identity matrix. Show that if $\boldsymbol{u}$ is an eigenvector of C_x, then it is also an eigenvector of C_s associated with eigenvalue $\lambda_s = \lambda_x - \alpha$.

$$\begin{aligned}
C_x \boldsymbol{u} &= \lambda_x \boldsymbol{u} = C_s \boldsymbol{u} + \alpha I \boldsymbol{u} \\
C_s \boldsymbol{u} &= \lambda_x \boldsymbol{u} - \alpha I \boldsymbol{u} \\
C_s \boldsymbol{u} &= (\lambda_x - \alpha) \boldsymbol{u} \\
\text{but} \quad C_s \boldsymbol{v} &= \lambda_s \boldsymbol{v} \\
\implies \lambda_s &= \lambda_x - \alpha \qquad \text{is an eigenvalue of } C_s \text{ with associated eigenvector } \boldsymbol{u}
\end{aligned}$$

2. Invertibility of a particular symmetric matrix

Let $A \in \mathbb{R}^{m \times n}$. Show that the matrix M defined as

$$M = \alpha^2 I + AA^T, \quad \alpha \neq 0 \in \mathbb{R}$$

is nonsingular, where $I = I_m$ and α is a nonzero real number.

THIS IS WRONG. TRY AGAIN

If M is singular, then it has at least one eigenvalue of 0. Since AA^T is symmetric, the diagonal entries are non-negative, that is, $(AA^T)_{i,i} = (a_{i,i})^2 \geq 0, \forall i = 1, \ldots n$. Examining the trace of the matrix M,

$$tr(M) = \sum i = 1^n M_{i,i} = \sum i = 1^n \left(\alpha^2 + (AA^A)_{i,i}\right) \geq n\alpha^2 > 0$$

Since the trace of a matrix is the sum of its eigenvalues, $(tr(A) = \sum_{i=1}^n \lambda_i)$, $tr(M) > 0$ implies 0 is not an eigenvalue of M. Hence, M is nonsingular.

3. Between-Class Scatter Matrix simplification

Show that the between-class scatter matrix, S_B, in the multi-class *Fisher Discriminant Analysis* is given by

$$S_B = \sum_{i=1}^{M} n_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T,$$

where M is the total number of distinct classes, n_i is the number of data points in class i, $\boldsymbol{m}_i$ is the class mean of the i^{th} class, and m is the mean across all n data points. You may use the facts that

$$S_T = S_B + S_W, \quad S_W = \sum_{i=1}^{M} \sum_{x \in D_i} (x - \boldsymbol{m}_i)(x - \boldsymbol{m}_i)^T, \quad \text{and} \quad S_T = \sum_{i=1}^{n} (x_i - \boldsymbol{m})(x_i - \boldsymbol{m})^T$$

Given what we know above, and the fact that the ith class mean $\boldsymbol{m}_i = \frac{1}{n_i}\sum_{x\in D_i}\boldsymbol{x}$ and the mean $\boldsymbol{m}$ across all data points is given by $\boldsymbol{m} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$, we will start with evaluating $S_B = S_T - S_W$.

Furthermore, note that $\forall \boldsymbol{x}_i \in X$ (the entire data set), $\boldsymbol{x}_i \in \bigcup_{j=1}^{M} D_j$, the union of all distinct classes, and also $\boldsymbol{m}_i n_i = \sum_{x\in D_i}\boldsymbol{x}$.

$$
\begin{aligned}
S_T - S_W &= \sum_{i=1}^{n}(x_i - \boldsymbol{m})(x_i - \boldsymbol{m})^T - \sum_{i=1}^{M}\sum_{x\in D_i}(x - \boldsymbol{m}_i)(x - \boldsymbol{m}_i)^T \\
&= \sum_{i=1}^{M}\sum_{x\in D_i}\left[(x - \boldsymbol{m})(x - \boldsymbol{m})^T - (x - \boldsymbol{m}_i)(x - \boldsymbol{m}_i)^T\right] \\
&= \sum_{i=1}^{M}\sum_{x\in D_i}\left[(xx^T - \boldsymbol{m}x^T - x\boldsymbol{m}^T + \boldsymbol{m}\boldsymbol{m}^T) - (xx^T - \boldsymbol{m}_i x^T - x\boldsymbol{m}_i{}^T + \boldsymbol{m}_i\boldsymbol{m}_i{}^T)\right] \\
&= \sum_{i=1}^{M}\sum_{x\in D_i}\left[(\boldsymbol{m}\boldsymbol{m}^T - \boldsymbol{m}x^T - x\boldsymbol{m}^T) - (\boldsymbol{m}_i\boldsymbol{m}_i{}^T - \boldsymbol{m}_i x^T - x\boldsymbol{m}_i{}^T)\right] \\
&= \sum_{i=1}^{M}\left[n_i\boldsymbol{m}\boldsymbol{m}^T - n_i\boldsymbol{m}_i\boldsymbol{m}_i{}^T + \sum_{x\in D_i}(\boldsymbol{m}_i x^T + x\boldsymbol{m}_i{}^T - \boldsymbol{m}x^T - x\boldsymbol{m}^T)\right] \\
&= \sum_{i=1}^{M}\left[n_i\boldsymbol{m}\boldsymbol{m}^T - n_i\boldsymbol{m}_i\boldsymbol{m}_i{}^T + \boldsymbol{m}_i\sum_{x\in D_i}x^T + \boldsymbol{m}_i{}^T\sum_{x\in D_i}x - \boldsymbol{m}\sum_{x\in D_i}x^T - \boldsymbol{m}^T\sum_{x\in D_i}x\right] \\
&= \sum_{i=1}^{M}\left[n_i\boldsymbol{m}\boldsymbol{m}^T - n_i\boldsymbol{m}_i\boldsymbol{m}_i{}^T + \boldsymbol{m}_i n_i\boldsymbol{m}_i{}^T + \boldsymbol{m}_i{}^T n_i\boldsymbol{m}_i - \boldsymbol{m}n_i\boldsymbol{m}_i{}^T - \boldsymbol{m}^T n_i\boldsymbol{m}_i\right] \\
&= \sum_{i=1}^{M} n_i\left[\boldsymbol{m}\boldsymbol{m}^T + \boldsymbol{m}_i{}^T\boldsymbol{m}_i - \boldsymbol{m}\boldsymbol{m}_i{}^T - \boldsymbol{m}^T\boldsymbol{m}_i\right] \\
&= \sum_{i=1}^{M} n_i(\boldsymbol{m} - \boldsymbol{m}_i)(\boldsymbol{m} - \boldsymbol{m}_i)^T
\end{aligned}
$$

Computing

1. KL Procedure for Gappy Data

This project concerns the application of the KL procedure for incomplete data [3]. Let the complete data set be translation-invariant:

$$
f(x_m, t_\mu) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{k}\sin[k(x_m - t_\mu)],
$$

where $m = 1, \ldots, M$, with M dimension of the ambient space (size of the spatial grid), and $\mu = 1, \ldots, P$, with P the number of points in the ensemble.

Let $x_m = \dfrac{2\pi(m-1)}{M}$ and $t_\mu = \dfrac{2\pi(\mu-1)}{P}$.

Select an ensemble of masks $\{\boldsymbol{m}^{(\mu)}\}$, $\mu = 1,\ldots,P$. where 10% of the indices are selected to be zero for each mask. Each pattern in the incomplete ensemble may be written as

$$\tilde{\boldsymbol{x}}^{(\mu)} = \boldsymbol{m}^{(\mu)}.\boldsymbol{f}^{(\mu)},$$

where $\left(\boldsymbol{f}^{(\mu)}\right)_m = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{k}\sin[k(x_m - t_\mu)]$. Let $P = M = 64$ and $N = 3$.

a. Compute the eigenvectors of this ensemble using the gappy algorithm [3].

b. Plot the eigenvalues as a function of the iteration, and continue until they converge.

c. Plot your final eigenfunctions corresponding to the 10 largest eigenvalues.

d. Plot the element $\tilde{\boldsymbol{x}}^{(1)}$ and the vector $\tilde{\boldsymbol{x}}_D$ repaired according to Equation

$$\tilde{\boldsymbol{x}} \approx \tilde{\boldsymbol{x}}_D = \sum_{n=1}^{D} \tilde{a}_n \phi^{(n)}. \tag{1}$$

Determine the value of D that provides the best approximation to the original non-gappy pattern vector.

2. Linear Discriminant Analysis (LDA)

a. Write a MATLAB routine to produce an optimal projection direction, w, using the two-class LDA criterion

$$w = \arg\max_{w} J(w) = \arg\max_{w} \frac{w^T S_B w}{w^T S_W w},$$

where

$$S_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T \quad \text{and} \quad S_W = \sum_{i=1}^{M}\sum_{x\in D_i}(x - \boldsymbol{m}_i)(x - \boldsymbol{m}_i)^T$$

are the between-class scatter matrix and the within-class scatter matrix, respectively. That is, your code should take in a set of data points with a clear indication which points belong to class 1 and which points belong to class 2, and output a single vector w that is the solution of the generalized eigenvalue problem $S_B w = \lambda S_W w$.

b. Now, use your subroutine in part (a) to project the EEG data onto a real line. Particularly, we can form a data point in $\mathbb{R}^{104019}$ by concatenating the columns for each trial, therefore having 10 data points for task 2 and 10 data points for task 3. You would then project these 20 points onto the real line with the w found with part (a). Plot the projected data on the real line and distinguish the classes with different symbols. Do you see a clear separation? Analyze your results.

3. Maximum Noise Fraction (MNF) method

Construct a $n \times 10$ matrix (choose $n \geq 250$) to serve as a ground truth data set so that each column is a n-dimensional time series. Next, add **correlated noise** to each column to create a noisy data set, X. The goal of this problem is to implement the Maximum Noise Fraction method to recover the ground truth as closely as possible from the noisy data. Suppose the source of the noise is unknown, you may estimate the noise covariance, N^TN, using the difference matrix as $N^TN = \frac{1}{2}dX^TdX$ where if

$$X = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_p(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_p(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_n) & x_2(t_n) & \cdots & x_p(t_n) \end{bmatrix}$$

then

$$dX = \begin{bmatrix} x_1(t_2) - x_1(t_1) & x_2(t_2) - x_2(t_1) & \cdots & x_2(t_2) - x_p(t_1) \\ x_1(t_3) - x_1(t_2) & x_2(t_3) - x_2(t_2) & \cdots & x_2(t_3) - x_p(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_n) - x_1(t_{n-1}) & x_2(t_n) - x_2(t_{n-1}) & \cdots & x_2(t_n) - x_p(t_{n-1}) \end{bmatrix}$$

Notice that $X \in \mathbb{R}^{n\times p}$ and $dX \in \mathbb{R}^{(n-1)\times p}$. In your report, examine and elaborate on the effect of a D-mode reconstruction on a single noisy signal for various values of D (i.e., choose a single column to filter). In a single graph, visually display the result of the original signal, noisy signal, and filtered (de-noised) data (with your best choice of D) to compare. Use the graph legend to distinguish each.

Code

Gram-Schmidt

References

[1] Chang, Jen-Mei. *Matrix Methods for Geometric Data Analysis and Recognition*. 2014.

[2] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, Jul 1997.

[3] R. Everson and L. Sirovich. The karhunen-loeve transform for incomplete data. *J. Opt. Soc. Am., A*, 12(8):16571664, 1995.