Statistical Pattern Recognition
Assignment 5

Ali Gholami

Department of Computer Engineering & Information Technology
Amirkabir University of Technology

https://aligholamee.github.io
aligholami7596@gmail.com

Abstract

Keywords. *Dimensionality Reduction, Principal Component Analysis, Fisher Linear Discriminant Analysis, Feature Subset Selection, Sequential Feature Selection, Data Visualization & Representation.*

1 Sequential Feature Selection

Given the following objective function, use **SFS**, **SBS** and **Plus-2 Minus-1 Selection** to select 3 features:

$$J(x) = 5x_1 + 7x_2 + 4x_3 + 9x_4 + 3x_5 - 2x_1x_2 + 2x_1x_2x_3 - 2x_2x_3 - 4x_1x_2x_3x_4 + 3x_1x_3x_5$$

Solution

SFS

In this method, we start feature selection from an empty set. We add features one by one and compute the value of the objective function with respect to each of the features being added. The feature with the largest objective function will be selected. The iteration goes on until all features all covered. We then select ideal features (subset with k features and maximum objective). The actual algorithm is as following:

1. Start with the empty set $Y_0 = \emptyset$
2. Select the next best feature $x^+ = argmax[J(Y_k + x)]$
3. Update $Y_{k+1} = Y_k + x^+; \quad k = k + 1$
4. Go to 2

Below is the demonstration of iterations taken to completely explore the search space. The first iteration is:

- $J(x_1) = 5$
- $J(x_2) = 7$
- $J(x_3) = 4$
- $J(x_4) = 9$
- $J(x_5) = 3$

According to the heuristic nature of sequential subset selection, we'll choose x_4 as the first best feature. We'll then generate subsets containing combination of features with x_4:

- $J(x_4x_1) = 14$
- $J(x_4x_2) = 16$
- $J(x_4x_3) = 13$
- $J(x_4x_5) = 12$

Thus, features x_4 and x_2 are selected until now. We'll drive the 3 sized subsets:

- $J(x_4x_2x_1) = 19$
- $J(x_4x_2x_3) = 18$
- $J(x_4x_2x_5) = 22$

Three best features selected by the algorithm are x_4, x_2 and x_5.

SBS

This method initiates the feature selection procedure using a complete subset of features. It then removes each feature and evaluates the objective function. The feature that causes the lowest decrease in the objective function will be remove (useless feature!). We'll stop when we reach a satisfying 3 sized feature subset. The algorithm is formally working as follows:

1. Start with the full set $Y_0 = X$
2. Remove the worst feature $x^- = argmax[J(Y_k - x)]$
3. Update $Y_{k+1} = Y_k - x^-; \quad k = k + 1$
4. Go to 2

Applying this algorithm on the given objective function yields the following results:

- $J(x_1x_2x_3x_4x_5) = 25$

And the results of removing each of the features:

- $J(x_1x_2x_3x_4) = 19$
- $J(x_1x_2x_3x_5) = 20$
- $J(x_1x_2x_4x_5) = 22$
- $J(x_1x_3x_4x_5) = 24$
- $J(x_2x_3x_4x_5) = 21$

It is obvious that x_2 is the most useless feature among these. We'll remove x_2 and obtain the feature subset with 4 features: x_1, x_3, x_4 and x_5.

- $J(x_1x_3x_4x_5) = 24$

We can obtain the following subsets:

- $J(x_1x_3x_4) = 18$
- $J(x_1x_3x_5) = 15$
- $J(x_1x_4x_5) = 17$
- $J(x_3x_4x_5) = 16$

x_5 will be removed since it has the lowest effect on the greatness of evaluation. The proper feature subset includes: x_1, x_3 and x_4.

2 PCA & FLDA

In this problem, dimensionality reduction with a two-feature two-class dataset is explored. Consider the following dataset and the test sample: $x = [0.851.15]^T$

- **Class 1**: [[0.8, 1.2], [0.9, 1.4], [1.2, 1.4], [1.1, 1.5]]
- **Class 2**: [[0.8, 1.1], [0.6, 1], [0.65, 1.1], [0.75, 0.9]]

(a) Demonstrate the preprocessing steps (need to show step-by-step details). Calculate the **mean** of each class (m_1 and m_2). Calculate the **covariance** matrix of each class.

(b) Using **Fisher's Linear Discriminant** to find a projection vector (w) which optimally separates the projections of these two classes.

(c) Is the vector derived from *FLD* along the same direction as the $m_1 - m_2$? Plot both of them on the same figure.

(d) Using **Principal Component Analysis** to reduce the dimension to 1 and plot the principal component on the same figure.

(e) Comment on the differences between **FLD** and **PCA** and $m_1 - m_2$. Make up a scenario where **FLD** will be aligned, perpendicular to $m_1 - m_2$, if possible at all.

(f) Project the test sample x onto w derived from **FLD** and determine its label.

(g) Project the test sample x onto the principal axis from **PCA** and determine its label.

Solution

(a) Mean of each class k can be calculated using (2.1).

$$m_k = \frac{1}{|n_k|}\sum_{i=1}^{n_k} x_i \tag{2.1}$$

Thus, for each class we'll have to following results:

- $m_1 = \frac{1}{4}[0.8 + 0.9 + 1.2 + 1.1 \quad 1.2 + 1.4 + 1.4 + 1.5]^T = [1 \quad 1.3]$
- $m_2 = \frac{1}{4}[0.8 + 0.6 + 0.65 + 0.75 \quad 1.1 + 1 + 1.1 + 0.9]^T = [0.7 \quad 1]$

Covariance matrix for each class k can be calculated using (2.2).

$$\Sigma_k = \frac{1}{n_k - 1}\sum_{i=1}^{n_k}(x - m_k)(x - m_k)^T \tag{2.2}$$

Thus, the results for each class will be:

- $\Sigma_1 = \frac{1}{3}\begin{bmatrix} -0.2 & -0.1 & 0.2 & 0.1 \\ -0.1 & 0.1 & 0.1 & 0.2 \end{bmatrix}\begin{bmatrix} -0.2 & -0.1 \\ -0.1 & 0.1 \\ 0.2 & 0.1 \\ 0.1 & 0.2 \end{bmatrix} = \begin{bmatrix} 0.03 & 0.01 \\ 0.01 & 0.02 \end{bmatrix}$
- $\Sigma_2 = \frac{1}{3}\begin{bmatrix} 0.1 & -0.1 & -0.05 & 0.05 \\ 0.1 & 0 & 0.1 & -0.1 \end{bmatrix}\begin{bmatrix} 0.1 & 0.1 \\ -0.1 & 0 \\ -0.05 & 0.1 \\ 0.05 & -0.1 \end{bmatrix} = \begin{bmatrix} 0.008 & 0 \\ 0 & 0.01 \end{bmatrix}$

(b) In order to find a proper projection vector (w), we shall compute the **within class scatter** matrix. Using the definition of S_w:

$$S_w = S_1 + S_2 \tag{2.3}$$

which is equal to the addition of scatter matrices. Scatter matrix of class k can be obtained from its covariance matrix using (2.4).

$$S_k = (|n_k| - 1)\Sigma_k \tag{2.4}$$

Replacing the results from the first part into (2.4) yields the following results:

- $S_1 = 3 \begin{bmatrix} 0.03 & 0.01 \\ 0.01 & 0.02 \end{bmatrix} = \begin{bmatrix} 0.09 & 0.03 \\ 0.03 & 0.06 \end{bmatrix}$
- $S_2 = 3 \begin{bmatrix} 0.008 & 0 \\ 0 & 0.01 \end{bmatrix} = \begin{bmatrix} 0.024 & 0 \\ 0 & 0.03 \end{bmatrix}$

Thus, the within class scatter matrix can be written as following:

$$S_w = \begin{bmatrix} 0.117 & 0.03 \\ 0.03 & 0.09 \end{bmatrix}$$