Aarhus University
Department of Engineering
Electrical & Computer Engineering

Introduction to Machine Learning

Alexandros Iosifidis

Solutions to Course Exercises

November 24, 2017

Contents

Chapter 1

Introduction

No exercises in Introduction!

Chapter 2

Unsupervised Learning

2.1 Show that, when the training data are centered, the projection matrix obtained by applying PCA (Eq. (2.16)) solves the Regression problem of Eq. (2.17).

Solution
We want to show that:

$$\begin{aligned} \mathbf{W}^* &= \arg\min \|\mathbf{W}\bar{\mathbf{Y}} - \bar{\mathbf{X}}\|_F^2, \\ subject\ to &: \mathbf{W}^T\mathbf{W} = \mathbf{I}, \end{aligned} \tag{2.1}$$

is equivalent to:

$$\begin{aligned} \mathbf{W}^* &= \arg\max Tr\left(\mathbf{W}^T\bar{\mathbf{X}}\bar{\mathbf{X}}^T\mathbf{W}\right), \\ subject\ to &: \mathbf{W}^T\mathbf{W} = \mathbf{I}, \end{aligned} \tag{2.2}$$

where:

$$\bar{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}\mathbf{1}_N^T \tag{2.3}$$

($\mathbf{1}_N \in \mathbb{R}^N$ is a vector of ones).
The first optimization problem can be written as:

$$\begin{aligned} \|\mathbf{W}\bar{\mathbf{Y}} - \bar{\mathbf{X}}\|_F^2 &= \|\mathbf{W}\mathbf{W}^T\bar{\mathbf{X}} - \bar{\mathbf{X}}\|_F^2 \\ &= Tr\left((\mathbf{W}\mathbf{W}^T\bar{\mathbf{X}} - \bar{\mathbf{X}})^T(\mathbf{W}\mathbf{W}^T\bar{\mathbf{X}} - \bar{\mathbf{X}})\right) \\ &= Tr\left(\bar{\mathbf{X}}^T\mathbf{W}\mathbf{W}^T\mathbf{W}\mathbf{W}^T\bar{\mathbf{X}} - 2\bar{\mathbf{X}}^T\mathbf{W}\mathbf{W}^T\bar{\mathbf{X}} + \bar{\mathbf{X}}^T\bar{\mathbf{X}}\right) \\ &= Tr\left(\bar{\mathbf{X}}^T\bar{\mathbf{X}} - \bar{\mathbf{X}}^T\mathbf{W}\mathbf{W}^T\bar{\mathbf{X}}\right) \\ &= const. - Tr\left(\bar{\mathbf{X}}^T\mathbf{W}\mathbf{W}^T\bar{\mathbf{X}}\right) \\ &= const. - Tr\left(\mathbf{W}^T\bar{\mathbf{X}}\bar{\mathbf{X}}^T\mathbf{W}^T\right). \end{aligned} \tag{2.4}$$

In the above we used the constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}$. Thus, in order to minimize the first optimization problem, we need to maximize the second.

2.2 Cluster the following data in two clusters:

$$\mathbf{X} = \begin{bmatrix} -1 & 0 & -0.5 & -1.5 & -2 & 0 & -1 & 1 & 1.3 & 0.7 & 2.5 & 0 \\ 0 & -1 & -0.5 & -1.5 & 0 & -2 & -1.3 & 1 & 0.7 & 1.3 & 1 & 1 \end{bmatrix}. \quad (2.5)$$

Initialize the cluster mean vectors by using:

$$\mathbf{M} = \begin{bmatrix} -1 & -0.9 \\ -1 & 0 \end{bmatrix} \quad (2.6)$$

and apply three iterations of the batch K-Means algorithm.

Solution
The distances of the data to the class mean vectors in the first iteration are:

$$\begin{bmatrix} 1.0000 & 1.0000 & 0.7071 & 0.7071 & 1.4142 & 1.4142 & 0.3000 & 2.8284 & 2.8601 & 2.8601 & 4.0311 & 2.236 \\ 0.1000 & 1.3454 & 0.6403 & 1.6155 & 1.1000 & 2.1932 & 1.3038 & 2.1471 & 2.3087 & 2.0616 & 3.5440 & 1.345 \end{bmatrix} \quad (2.7)$$

Thus, the labels of the data in the first iteration are:

$$\mathbf{l} = [2\ 1\ 2\ 1\ 2\ 1\ 1\ 2\ 2\ 2\ 2\ 2] \quad (2.8)$$

and the mean vectors become:

$$\mathbf{M} = \begin{bmatrix} -0.6250 & 0.2500 \\ -1.4500 & 0.5625 \end{bmatrix}. \quad (2.9)$$

For the second iteration we have:

$$\begin{bmatrix} 1.4977 & 0.7701 & 0.9582 & 0.8764 & 1.9983 & 0.8325 & 0.4039 & 2.9399 & 2.8858 & 3.0526 & 3.9709 & 2.528 \\ 1.3707 & 1.5824 & 1.3005 & 2.7049 & 2.3192 & 2.5747 & 2.2431 & 0.8683 & 1.0590 & 0.8639 & 2.2921 & 0.503 \end{bmatrix} \quad (2.10)$$

$$\mathbf{l} = [2\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2] \quad (2.11)$$

$$\mathbf{M} = \begin{bmatrix} -0.8333 & 0.7500 \\ -1.0500 & 0.8333 \end{bmatrix}. \quad (2.12)$$

For the second iteration we have:

$$\begin{bmatrix} 1.0631 & 0.8348 & 0.6431 & 0.8043 & 1.5696 & 1.2637 & 0.3005 & 2.7502 & 2.7593 & 2.8060 & 3.9133 & 2.2129 \\ 1.9383 & 1.9808 & 1.8276 & 3.2414 & 2.8735 & 2.9309 & 2.7593 & 0.3005 & 0.5659 & 0.4693 & 1.7579 & 0.7683 \end{bmatrix}. \tag{2.13}$$

$$\mathbf{l} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2] \tag{2.14}$$

$$\mathbf{M} = \begin{bmatrix} -0.8571 & 1.1000 \\ -0.9000 & 1.0000 \end{bmatrix}. \tag{2.15}$$

2.3 Cluster the data in 2.2 by applying the batch Fuzzy K-Means algorithm. Use the same initialization for the cluster mean vectors and a value of $\gamma = 2$.

Solution
The distances of the data to the class mean vectors in the first iteration are:

$$\begin{bmatrix} 1.0000 & 1.0000 & 0.7071 & 0.7071 & 1.4142 & 1.4142 & 0.3000 & 2.8284 & 2.8601 & 2.8601 & 4.0311 & 2.2361 \\ 0.1000 & 1.3454 & 0.6403 & 1.6155 & 1.1000 & 2.1932 & 1.3038 & 2.1471 & 2.3087 & 2.0616 & 3.5440 & 1.3454 \end{bmatrix} \tag{2.16}$$

and the membership values are:

$$\begin{bmatrix} 0.0099 & 0.6441 & 0.4505 & 0.8392 & 0.3769 & 0.7063 & 0.9497 & 0.3656 & 0.3945 & 0.3419 & 0.4360 & 0.2658 \\ 0.9901 & 0.3559 & 0.5495 & 0.1608 & 0.6231 & 0.2937 & 0.0503 & 0.6344 & 0.6055 & 0.6581 & 0.5640 & 0.7342 \end{bmatrix} \tag{2.17}$$

leading to cluster mean vectors:

$$\mathbf{M} = \begin{bmatrix} -0.9899 & 0.4899 \\ -2.9875 & 1.6875 \end{bmatrix}. \tag{2.18}$$

For the second iteration we have distances equal to:

$$\begin{bmatrix} 2.9875 & 2.2204 & 2.5353 & 1.5725 & 3.1537 & 1.3983 & 1.6876 & 4.4565 & 4.3407 & 4.6086 & 5.2991 & 4.1086 \\ 2.2511 & 2.7318 & 2.4011 & 3.7577 & 3.0079 & 3.7199 & 3.3384 & 0.8561 & 1.2773 & 0.4408 & 2.1244 & 0.8442 \end{bmatrix} \tag{2.19}$$

and the membership values are:

$$\begin{bmatrix} 0.3622 & 0.6022 & 0.4728 & 0.8510 & 0.4764 & 0.8762 & 0.7965 & 0.0356 & 0.0797 & 0.0091 & 0.1385 & 0.0405 \\ 0.6378 & 0.3978 & 0.5272 & 0.1490 & 0.5236 & 0.1238 & 0.2035 & 0.9644 & 0.9203 & 0.9909 & 0.8615 & 0.9595 \end{bmatrix} \tag{2.20}$$

leading to cluster mean vectors:

$$\mathbf{M} = \begin{bmatrix} -3.1325 & 2.6325 \\ -4.6207 & 3.3207 \end{bmatrix}. \tag{2.21}$$

For the third iteration we have distances equal to:

$$\begin{bmatrix} 5.0891 & 4.7877 & 4.8899 & 3.5220 & 4.7575 & 4.0842 & 3.9465 & 6.9764 & 6.9251 & 7.0529 & 7.9573 & 6.434 \\ 4.9216 & 5.0595 & 4.9407 & 6.3496 & 5.6998 & 5.9364 & 5.8776 & 2.8374 & 2.9400 & 2.7961 & 2.3245 & 3.509 \end{bmatrix} \tag{2.22}$$

and the membership values are:

$$\begin{bmatrix} 0.4833 & 0.5276 & 0.5052 & 0.7647 & 0.5894 & 0.6787 & 0.6893 & 0.1419 & 0.1527 & 0.1358 & 0.0786 & 0.229 \\ 0.5167 & 0.4724 & 0.4948 & 0.2353 & 0.4106 & 0.3213 & 0.3107 & 0.8581 & 0.8473 & 0.8642 & 0.9214 & 0.770 \end{bmatrix} \tag{2.23}$$

leading to cluster mean vectors:

$$\mathbf{M} = \begin{bmatrix} -3.1189 & 2.6189 \\ -3.4474 & 2.1474 \end{bmatrix}. \tag{2.24}$$

2.4 Consider the following vectors (forming the columns of $\mathbf{X}$

$$\mathbf{X} = \begin{bmatrix} -1 & 0 & -0.5 & -1.5 & -2 & 0 & -1 & 1 & 1.3 & 0.7 & 2.5 & 0 \\ 0 & -1 & -0.5 & -1.5 & 0 & -2 & -1.3 & 1 & 0.7 & 1.3 & 1 & 1 \end{bmatrix}. \tag{2.25}$$

1. Center the data in $\mathbf{X}$.
2. Standardize the data in $\mathbf{X}$
3. Normalize the data in $\mathbf{X}$ using their l_2 norm.

.

Solution
We have:

$$\mathbf{m} = [-0.417 \ \ -0.1083]^T \tag{2.26}$$

and

$$\mathbf{s} = [1.276 \ \ 1.1373]^T. \tag{2.27}$$

The centered data matrix is:

$$\bar{\mathbf{X}} = \begin{bmatrix} -0.95 & 0.04 & -0.45 & -1.45 & -1.95 & 0.04 & -0.95 & 1.04 & 1.34 & 0.74 & 2.54 & 0.041 \\ 0.10 & -0.89 & -0.39 & -1.39 & 0.10 & -1.89 & -1.19 & 1.10 & 0.80 & 1.40 & 1.10 & 1.10 \end{bmatrix}. \tag{2.28}$$

The matrix with the standardized data is:

$$\hat{\mathbf{X}} = \begin{bmatrix} -0.75 & 0.03 & -0.35 & -1.14 & -1.53 & 0.03 & -0.75 & 0.81 & 1.05 & 0.58 & 1.99 & 0.03 \\ 0.09 & -0.78 & -0.34 & -1.22 & 0.09 & -1.66 & -1.04 & 0.97 & 0.71 & 1.23 & 0.97 & 0.97 \end{bmatrix}. \tag{2.29}$$

The matrix with the normalized data is:

$$\tilde{\mathbf{X}} = \begin{bmatrix} -1 & 0 & -0.707 & -0.707 & -1 & 0 & -0.609 & 0.7071 & 0.8804 & 0.474 & 0.928 & 0 \\ 0 & -1 & -0.707 & -0.707 & 0 & -1 & -0.792 & 0.7071 & 0.474 & 0.887 & 0.371 & 1 \end{bmatrix}. \tag{2.30}$$

2.5 Consider the following vectors (forming the columns of $\mathbf{X}$

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 2 & 3 & 3 & 4 \\ 1 & 2 & 3 & 2 & 3 & 4 \end{bmatrix}. \tag{2.31}$$

Project this data to the one-dimensional space determined by PCA.

Solution

We have:

$$\mathbf{m} = [2.5 \;\; 2.5]^T. \tag{2.32}$$

$$\bar{\mathbf{X}} = \begin{bmatrix} -1.5 & -0.5 & -0.5 & 0.5 & 0.5 & 1.5 \\ -1.5 & -0.5 & 0.5 & -0.5 & 0.5 & 1.5 \end{bmatrix}. \tag{2.33}$$

The covariance matrix is:

$$\mathbf{S}_T = \bar{\mathbf{X}} * \bar{\mathbf{X}}^T = \begin{bmatrix} 5.5 & 4.5 \\ 4.5 & 5.5 \end{bmatrix}. \tag{2.34}$$

The eigenvalues of $\mathbf{S}_T$ are $\lambda = \{10, 1\}$. The eigenvector corresponding to the maximal eigenvalues is:

$$\mathbf{v} = [0.7071 \;\; 0.7071]^T. \tag{2.35}$$

The projected data is:

$$\mathbf{x} = \mathbf{v}^T\mathbf{X} = [1.414 \;\; 2.828 \;\; 3.5355 \;\; 3.5355 \;\; 4.242 \;\; 5.656]. \tag{2.36}$$

Chapter 3

Probability-based Learning

3.1 Two classes formed by the following samples:

$$c_1 = \begin{bmatrix} -1 & 0 & -0.5 & -1.5 & -2 & 0 & -1 \\ 0 & -1 & -0.5 & -1.5 & 0 & -2 & -1.3 \end{bmatrix} \tag{3.1}$$

$$c_2 = \begin{bmatrix} 1 & 1.3 & 0.7 & 2.5 & 0 \\ 1 & 0.7 & 1.3 & 1 & 1 \end{bmatrix}. \tag{3.2}$$

The probability of a sample x given class $c_k,\ k = 1, 2$ is calculated as a function of the relative distance of x from the corresponding class mean vector $\mathbf{m}_k$, i.e.:

$$p(\mathbf{x}|c_k) = \frac{e^{-\|\mathbf{x}-\mathbf{m}_k\|_2}}{\sum_{l=1}^{K} e^{-\|\mathbf{x}-\mathbf{m}_l\|_2}}. \tag{3.3}$$

Classify the following samples:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_5] = \begin{bmatrix} 0 & 1 & -1 & 0.7 & -0.2 \\ 0 & 1 & 0 & -0.2 & 1.5 \end{bmatrix}. \tag{3.4}$$

Solution
The a priori probability of each class is given by:

$$P(c_k) = \frac{N_k}{\sum_{l=1}^{K} N_l} \tag{3.5}$$

Thus, we have $P(c_1) = 7/12 = 0.5833$ and $P(c_2) = 5/12 = 0.4167$.
We have $\mathbf{m}_1 = [-0.8571 \ \ -0.9]^T$ and $\mathbf{m}_2 = [1.1 \ \ 1]^T$.
The probability $p(\mathbf{x}|c_k)$ is given by:

$$p(\mathbf{x}|c_k) = \frac{e^{-\|\mathbf{x}-\mathbf{m}_k\|_2}}{\sum_{l=1}^{K} e^{-\|\mathbf{x}-\mathbf{m}_l\|_2}}. \tag{3.6}$$

Thus, based on the Bayes' decision rule classifies $\mathbf{x}$ to the class with the maximal:

$$P(c_k|\mathbf{x}) = p(\mathbf{x}|c_k)P(c_k). \tag{3.7}$$

We calculate $P(c_k|\mathbf{x})$ for $k = 1, 2$ and we select the class corresponding to the highest value.
Thus, we have:

$$p(\mathbf{x}|c_1) = [0.5606\ \ 0.0719\ \ 0.8045\ \ 0.3911\ \ 0.2505] \tag{3.8}$$

$$p(\mathbf{x}|c_2) = [0.4393\ \ 0.9280\ \ 0.1954\ \ 0.6088\ \ 0.7494] \tag{3.9}$$

leading to:

$$P(c_1|\mathbf{x}) = [0.327\ \ 0.04198\ \ 0.469292\ \ 0.22819\ \ 0.1461] \tag{3.10}$$

$$P(c_2|\mathbf{x}) = [0.183\ \ 0.38668\ \ 0.081457\ \ 0.25367\ \ 0.312256]. \tag{3.11}$$

By calculating the difference $D = P(c_1|\mathbf{x}) - P(c_2|\mathbf{x})$:

$$D = [0.14397\ \ -0.3447\ \ 0.38783\ \ -0.02548\ \ -0.166], \tag{3.12}$$

we classify $\mathbf{x}_1$ and $\mathbf{x}_2$ to class c_1 and , $\mathbf{x}_2$, $\mathbf{x}_4$ and $\mathbf{x}_5$ to class c_2.

3.2 Two classes formed by the following samples:

$$c_1 = \begin{bmatrix} -1 & 0 & -0.5 & -1.5 & -2 & 0 & -1 \\ 0 & -1 & -0.5 & -1.5 & 0 & -2 & -1.3 \end{bmatrix} \tag{3.13}$$

$$c_2 = \begin{bmatrix} 1 & 1.3 & 0.7 & 2.5 & 0 \\ 1 & 0.7 & 1.3 & 1 & 1 \end{bmatrix}. \tag{3.14}$$

The probability of a sample x given class $c_k,\ k = 1, 2$ is calculated as a function of the relative distance of x from the corresponding class mean vector $\mathbf{m}_k$, i.e.:

$$p(\mathbf{x}|c_k) = \frac{e^{-\|\mathbf{x}-\mathbf{m}_k\|_2}}{\sum_{l=1}^{K} e^{-\|\mathbf{x}-\mathbf{m}_l\|_2}}. \tag{3.15}$$

Moreover, the loss function $\lambda(\alpha_j|c_k)$ is given by:

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{3.16}$$

Classify the following samples:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_5] = \begin{bmatrix} 0 & 1 & -1 & 0.7 & -0.2 \\ 0 & 1 & 0 & -0.2 & 1.5 \end{bmatrix}. \tag{3.17}$$

Solution
The a priori probability of each class is given by:

$$P(c_k) = \frac{N_k}{\sum_{l=1}^{K} N_l} \tag{3.18}$$

Thus, we have $P(c_1) = 7/12 = 0.5833$ and $P(c_2) = 5/12 = 0.4167$.
We have $\mathbf{m}_1 = [-0.8571 \quad -0.9]^T$ and $\mathbf{m}_2 = [1.1 \;\; 1]^T$.
The probability $p(\mathbf{x}|c_k)$ is given by:

$$p(\mathbf{x}|c_k) = \frac{e^{\|\mathbf{x}-\mathbf{m}_k\|_2}}{\sum_{l=1}^{K} e^{\|\mathbf{x}-\mathbf{m}_l\|_2}}. \tag{3.19}$$

Thus:

$$P(c_k|\mathbf{x}) = p(\mathbf{x}|c_k)P(c_k), \tag{3.20}$$

and the risk of taking action α_i (i.e. deciding that the correct class is c_i) is:

$$R(\alpha_i|\mathbf{x}) = \sum_{l=1}^{K} \lambda(\alpha_i|c_k)P(c_k|\mathbf{x}). \tag{3.21}$$

Given $R(\alpha_1|\mathbf{x})$ and $R(\alpha_1|\mathbf{x})$, we choose the action corresponding to the minimum risk.
Thus, we have:

$$p(\mathbf{x}|c_1) = [0.5606 \;\; 0.071966 \;\; 0.8045 \;\; 0.39118 \;\; 0.2505] \tag{3.22}$$

$$p(\mathbf{x}|c_2) = [0.4393 \;\; 0.92803 \;\; 0.1954 \;\; 0.60881 \;\; 0.74941] \tag{3.23}$$

leading to:

$$P(c_1|\mathbf{x}) = [0.32703 \;\; 0.04198 \;\; 0.46929 \;\; 0.22819 \;\; 0.14617] \tag{3.24}$$

$$P(c_2|\mathbf{x}) = [0.183 \;\; 0.38668 \;\; 0.081457 \;\; 0.253673 \;\; 0.31225]. \tag{3.25}$$

The risks are:

$$R(\alpha_1|\mathbf{x}) = [0.18306\ \ 0.38668\ \ 0.081457\ \ 0.253673\ \ 0.312256] \tag{3.26}$$

$$R(\alpha_2|\mathbf{x}) = [0.32703\ \ 0.04198\ \ 0.46929\ \ 0.22819\ \ 0.14617]. \tag{3.27}$$

By calculating the difference $D = R(\alpha_1|\mathbf{x}) - R(\alpha_2|\mathbf{x})$:

$$D = [0.14397\ \ -0.3447\ \ 0.387835\ \ -0.02548\ \ -0.1660], \tag{3.28}$$

we classify $\mathbf{x}_1$ and $\mathbf{x}_3$ to class c_1 and $\mathbf{x}_2$, $\mathbf{x}_4$ and $\mathbf{x}_5$ to class c_2.

3.3 Solve the classification problem in 3.2 using the following risks:

$$\mathbf{\Lambda} = \begin{bmatrix} 0.4 & 0.8 \\ 0.6 & 0.2 \end{bmatrix}. \tag{3.29}$$

Solution
The a priori probability of each class is given by:

$$P(c_k) = \frac{N_k}{\sum_{l=1}^{K} N_l} \tag{3.30}$$

Thus, we have $P(c_1) = 7/12 = 0.5833$ and $P(c_2) = 5/12 = 0.4167$.
We have $\mathbf{m}_1 = [-0.8571\ \ -0.9]^T$ and $\mathbf{m}_2 = [1.1\ \ 1]^T$.
The probability $p(\mathbf{x}|c_k)$ is given by:

$$p(\mathbf{x}|c_k) = \frac{e^{\|\mathbf{x}-\mathbf{m}_k\|_2}}{\sum_{l=1}^{K} e^{\|\mathbf{x}-\mathbf{m}_l\|_2}}. \tag{3.31}$$

Thus:

$$P(c_k|\mathbf{x}) = p(\mathbf{x}|c_k)P(c_k), \tag{3.32}$$

and the risk of taking action α_i (i.e. deciding that the correct class is c_i) is:

$$R(\alpha_i|\mathbf{x}) = \sum_{l=1}^{K} \lambda(\alpha_i|c_k)P(c_k|\mathbf{x}). \tag{3.33}$$

Given $R(\alpha_1|\mathbf{x})$ and $R(\alpha_1|\mathbf{x})$, we choose the action corresponding to the minimum risk.
Thus, we have:

$$p(\mathbf{x}|c_1) = [0.5606 \;\; 0.07196 \;\; 0.8045 \;\; 0.3911 \;\; 0.2505] \tag{3.34}$$

$$p(\mathbf{x}|c_2) = [0.43936 \;\; 0.92803 \;\; 0.19549 \;\; 0.6088 \;\; 0.7494] \tag{3.35}$$

leading to:

$$P(c_1|\mathbf{x}) = [0.32703 \;\; 0.04198 \;\; 0.46929 \;\; 0.22819 \;\; 0.14617] \tag{3.36}$$

$$P(c_2|\mathbf{x}) = [0.18306 \;\; 0.3866 \;\; 0.08145 \;\; 0.25367 \;\; 0.3122]. \tag{3.37}$$

The risks are:

$$R(\alpha_1|\mathbf{x}) = [0.2772 \;\; 0.326136 \;\; 0.25288 \;\; 0.29421 \;\; 0.3082] \tag{3.38}$$

$$R(\alpha_2|\mathbf{x}) = [0.2328 \;\; 0.10252 \;\; 0.29786 \;\; 0.18764 \;\; 0.15015]. \tag{3.39}$$

By calculating the difference $D = R(\alpha_1|\mathbf{x}) - R(\alpha_2|\mathbf{x})$:

$$D = [-0.04443 \;\; -0.2236 \;\; 0.04498 \;\; -0.10656 \;\; -0.1581], \tag{3.40}$$

we classify $\mathbf{x}_3$ to class c_1 and $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_4$ to class c_2.

3.4 Two classes formed by the following samples:

$$c_1 = \begin{bmatrix} -1 & 0 & -0.5 & -1.5 & -2 & 0 & -1 \\ 0 & -1 & -0.5 & -1.5 & 0 & -2 & -1.3 \end{bmatrix} \tag{3.41}$$

$$c_2 = \begin{bmatrix} 1 & 1.3 & 0.7 & 2.5 & 0 \\ 1 & 0.7 & 1.3 & 1 & 1 \end{bmatrix}. \tag{3.42}$$

We assume that each class follows a Normal Distribution with covariance matrix:

$$\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \tag{3.43}$$

The probability of a sample x given class $c_k,\ k = 1, 2$ is calculated as a function of the relative distance of x from the corresponding class mean vector $\mathbf{m}_k$, i.e.:

$$p(\mathbf{x}|c_k) = \frac{e^{-\|\mathbf{x}-\mathbf{m}_k\|_2}}{\sum_{l=1}^{K} e^{-\|\mathbf{x}-\mathbf{m}_l\|_2}}. \tag{3.44}$$

Classify the following samples:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_5] = \begin{bmatrix} 0 & 1 & -1 & 0.7 & -0.2 \\ 0 & 1 & 0 & -0.2 & 1.5 \end{bmatrix}. \tag{3.45}$$

Solution
The a priori probability of each class is given by:

$$P(c_k) = \frac{N_k}{\sum_{l=1}^{K} N_l} \tag{3.46}$$

Thus, we have $P(c_1) = 7/12 = 0.5833$ and $P(c_2) = 5/12 = 0.4167$.
We have $\mathbf{m}_1 = [-0.8571 \ \ -0.9]^T$ and $\mathbf{m}_2 = [1.1 \ \ 1]^T$.
The probability $p(\mathbf{x}|c_k)$ is given by:

$$p(\mathbf{x}|c_k) = \frac{e^{-(\mathbf{x}-\mathbf{m}_k)^T \mathbf{\Sigma}^{-1} (\mathbf{x}-\mathbf{m}_k)}}{\sum_{l=1}^{K} e^{-(\mathbf{x}-\mathbf{m}_l)^T \mathbf{\Sigma}^{-1} (\mathbf{x}-\mathbf{m}_l)}}. \tag{3.47}$$

Thus, based on the Bayes' decision rule:

$$P(c_k|\mathbf{x}) = p(\mathbf{x}|c_k) P(c_k), \tag{3.48}$$

and we select the class corresponding to the highest value.
Thus, we have:

$$p(\mathbf{x}|c_1) = [0.6388 \ \ 0.00525 \ \ 0.9888 \ \ 0.1431 \ \ 0.1828] \tag{3.49}$$

$$p(\mathbf{x}|c_2) = [0.3611 \ \ 0.9947 \ \ 0.01115 \ \ 0.8568 \ \ 0.8171] \tag{3.50}$$

leading to:

$$P(c_1|\mathbf{x}) = [0.3726 \ \ 0.0030 \ \ 0.57682 \ \ 0.08348 \ \ 0.10669] \tag{3.51}$$

$$P(c_2|\mathbf{x}) = [0.1504 \ \ 0.41447 \ \ 0.0046 \ \ 0.35703 \ \ 0.34045]. \tag{3.52}$$

By calculating the difference $D = P(c_1|\mathbf{x}) - P(c_2|\mathbf{x})$:

$$D = [0.22216 \;\; -0.4114 \;\; 0.5721 \;\; -0.27355 \;\; -0.23376], \tag{3.53}$$

we classify $\mathbf{x}_1$ and $\mathbf{x}_3$ to class c_1 and $\mathbf{x}_2$, $\mathbf{x}_4$ and $\mathbf{x}_3$ to class c_2.

3.5 Solve the classification problem in 3.4 using the covariance matrix:

$$\mathbf{\Sigma} = \frac{1}{2}(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2), \tag{3.54}$$

where $\mathbf{\Sigma}_k$ is the actual covariance matrix of class c_k.

Solution
In this case the covariance matrix is:

$$\mathbf{S} = \begin{bmatrix} 3.3686 & -0.89 \\ -0.89 & 1.85 \end{bmatrix}. \tag{3.55}$$

and its inverse:

$$\mathbf{S}^{-1} = \begin{bmatrix} 0.3401 & 0.1636 \\ 0.1636 & 0.6193 \end{bmatrix}. \tag{3.56}$$

Thus, we have:

$$p(\mathbf{x}|c_1) = [0.5955 \;\; 0.01035 \;\; 0.9121 \;\; 0.405761 \;\; 0.02382] \tag{3.57}$$

$$p(\mathbf{x}|c_2) = [0.40448 \;\; 0.98964 \;\; 0.087884 \;\; 0.59423 \;\; 0.97617] \tag{3.58}$$

leading to:

$$P(c_1|\mathbf{x}) = [0.3473 \;\; 0.00604 \;\; 0.53206 \;\; 0.23669 \;\; 0.013897] \tag{3.59}$$

$$P(c_2|\mathbf{x}) = [0.16853 \;\; 0.41235 \;\; 0.036618 \;\; 0.24759 \;\; 0.40673]. \tag{3.60}$$

By calculating the difference $D = P(c_1|\mathbf{x}) - P(c_2|\mathbf{x})$:

$$D = [0.17885 \;\; -0.40630 \;\; 0.49544 \;\; -0.010904 \;\; -0.39284], \tag{3.61}$$

we classify $\mathbf{x}_1$ and $\mathbf{x}_3$ to class c_1 and $\mathbf{x}_2$, $\mathbf{x}_4$ and $\mathbf{x}_5$ to class c_2.

Chapter 4

Linear Methods

4.1 Show that maximizing the LDA criterion in Eq. (4.22) is equivalent to minimizing $\mathcal{J}_2(\mathbf{W})$:

$$\mathcal{J}_2(\mathbf{W}) = \frac{Tr(\mathbf{W}^T\mathbf{S}_w\mathbf{W})}{Tr(\mathbf{W}^T\mathbf{S}_T\mathbf{W})} \tag{4.1}$$

Solution
We use the equality $\mathbf{S}_T = \mathbf{S}_w + \mathbf{S}_b$ and we have:

$$\mathcal{J}(\mathbf{W}) + 1 = \frac{Tr(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{Tr(\mathbf{W}^T\mathbf{S}_w\mathbf{W})} + \frac{Tr(\mathbf{W}^T\mathbf{S}_w\mathbf{W})}{Tr(\mathbf{W}^T\mathbf{S}_w\mathbf{W})} = \frac{1}{\mathcal{J}_2(\mathbf{W})}. \tag{4.2}$$

4.2 Two classes formed by the following samples:

$$c_1 = \begin{bmatrix} -1 & 0 & -0.5 & -1.5 & -2 & 0 & -1 \\ 0 & -1 & -0.5 & -1.5 & 0 & -2 & -1.3 \end{bmatrix} \tag{4.3}$$

$$c_2 = \begin{bmatrix} 1 & 1.3 & 0.7 & 2.5 & 0 \\ 1 & 0.7 & 1.3 & 1 & 1 \end{bmatrix}. \tag{4.4}$$

Classify the following samples using the Nearest Centroid and Nearest Neighbor classifiers:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_5] = \begin{bmatrix} 0 & 1 & -1 & 0.7 & -0.2 \\ 0 & 1 & 0 & -0.2 & 1.5 \end{bmatrix}. \tag{4.5}$$

Solution
The class centroids are equal to:

$$\mathbf{M} = \begin{bmatrix} -0.857 & 1.1 \\ -0.9 & 1 \end{bmatrix} \tag{4.6}$$

The distances of samples from the class centers are:

$$\mathbf{D}_{nc} = \begin{bmatrix} 1.24 & 2.65 & 0.911 & 1.7072 & 2.488 \\ 1.48 & 0.1 & 2.325 & 1.264 & 1.39 \end{bmatrix} \tag{4.7}$$

Thus, we classify $\mathbf{x}_1$ and $\mathbf{x}_3$ to c_1 and $\mathbf{x}_2$, $\mathbf{x}_4$ and $\mathbf{x}_5$ to c_2.
The minimum distances of each sample from the data of each class is equal to:

$$\mathbf{D}_{nn} = \begin{bmatrix} 0.707 & 2.12 & 0 & 1.063 & 1.7 \\ 1 & 0 & 1.3 & 1.08 & 0.53 \end{bmatrix}. \tag{4.8}$$

Thus, we classify $\mathbf{x}_1$, $\mathbf{x}_3$ and $\mathbf{x}_4$ to c_1 and $\mathbf{x}_2$ and $\mathbf{x}_5$ to c_2.

4.3 Two classes formed by the following samples:

$$c_1 = \begin{bmatrix} -1 & 0 & -0.5 & -1.5 & -2 & 0 & -1 \\ 0 & -1 & -0.5 & -1.5 & 0 & -2 & -1.3 \end{bmatrix} \tag{4.9}$$

$$c_2 = \begin{bmatrix} 1 & 1.3 & 0.7 & 2.5 & 0 \\ 1 & 0.7 & 1.3 & 1 & 1 \end{bmatrix}. \tag{4.10}$$

Calculate the data projections in the 1-dimensional feature space determined by applying Fisher Discriminant Analysis.

Solution
The within-class scatter matrix is equal to:

$$\mathbf{S}_w = \begin{bmatrix} 6.737 & -1.78 \\ -1.78 & 3.7 \end{bmatrix}. \tag{4.11}$$

The between-class scatter matrix is equal to:

$$\mathbf{S}_b = \begin{bmatrix} 3.8304 & 3.7186 \\ 3.7186 & 3.61 \end{bmatrix}. \tag{4.12}$$

The eigenvalues of the matrix $\mathbf{S} = \mathbf{S}_w^{-1}\mathbf{S}_b$ are $\lambda = \{1.2218, 0\}$. The eigenvector corresponding to the maximal eigenvalue is:

$$\mathbf{v} = [-0.546 \quad -0.837]^T. \tag{4.13}$$

Thus, the projected data is:

$$\begin{aligned} x &= \mathbf{v}^T\mathbf{X} \\ &= [0.546 \ 0.837 \ 0.691 \ 2.075 \ 1.092 \ 1.67 \ 1.63 \ -1.38 \ -1.29 \ -1.47 \ -2.2 \ -0.83]. \end{aligned}$$

4.4 Two classes formed by the following samples:

$$c_1 = \begin{bmatrix} -1 & 0 & -0.5 & -1.5 & -2 & 0 & -1 \\ 0 & -1 & -0.5 & -1.5 & 0 & -2 & -1.3 \end{bmatrix} \tag{4.14}$$

$$c_2 = \begin{bmatrix} 1 & 1.3 & 0.7 & 2.5 & 0 \\ 1 & 0.7 & 1.3 & 1 & 1 \end{bmatrix}. \tag{4.15}$$

Train a Perceptron using the above training data. Initialize the weights as $\mathbf{w}(0) = [0.1 \ 0.1 \ 0]^T$ and use a learning rate equal to $\eta = 0.01$. Use as positive class c_1 and as negative class c_2.
Classify the following samples using the trained model:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_5] = \begin{bmatrix} 0 & 1 & -1 & 0.7 & -0.2 \\ 0 & 1 & 0 & -0.2 & 1.5 \end{bmatrix}. \tag{4.16}$$

Solution
For $\mathbf{w} = \mathbf{w}(0)$ the network's response for all samples is:

$$\mathbf{o} = \begin{bmatrix} -0.1 & -0.1 & -0.1 & -0.3 & -0.2 & -0.2 & -0.23 & 0.2 & 0.2 & 0.2 & 0.35 & 0.1 \end{bmatrix}. \tag{4.17}$$

This means that all samples are not classified correctly. The update vector is equal to:

$$\triangle\mathbf{w} = [-11.5 \quad -11.3 \ \ 2]^T \tag{4.18}$$

and the new weight vector becomes:

$$\mathbf{w}(1) = \mathbf{w}(0) + \eta\triangle\mathbf{w} = [-0.015 \quad -0.013 \ \ 0.02]^T. \tag{4.19}$$

Using $\mathbf{w}(1)$ the network's response for all samples is:

$$\mathbf{o} = \Big[0.035 \quad 0.033 \quad 0.034 \quad 0.062 \quad 0.05 \quad 0.046 \quad 0.0519 \quad -0.008 \quad -0.0086 \quad -0.0074 \quad -0.0305 \quad 0.0 \tag{4.20}$$

which means that only the last sample is misclassified. Thus, the update vector becomes:

$$\triangle\mathbf{w} = [0 \quad -1 \quad -1]^T \tag{4.21}$$

and the new weight vector becomes:

$$\mathbf{w}(2) = \mathbf{w}(1) + \eta\triangle\mathbf{w} = [-0.015 \quad -0.023 \quad 0.01]^T. \tag{4.22}$$

$\mathbf{w}(2)$ classifies correctly all the training samples.
Using $\mathbf{w}(2)$ the network's output for the test samples is:

$$\mathbf{o} = [0.01 \quad -0.028 \quad 0.025 \quad 0.0041 \quad -0.0215], \tag{4.23}$$

and we classify $\mathbf{x}_1$, $\mathbf{x}_3$ and $\mathbf{x}_4$ to c_1 and $\mathbf{x}_2$ and $\mathbf{x}_3$ to c_2.

4.5 Show that for a multi-class classifier with K linear discriminant functions of the form:

$$g_k(\mathbf{x}) = \mathbf{w}_k^T\mathbf{x} + w_{k0},\ k = 1, \ldots, K, \tag{4.24}$$

the decision regions are convex by showing that if $\mathbf{x}_1 \in \mathcal{R}_k$ and $\mathbf{x}_2 \in \mathcal{R}_k$, then $\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in \mathcal{R}_k$ for $0 \leq \lambda \leq 1$.

Solution
Since both $\mathbf{x}_1$ and $\mathbf{x}_2$ belong to $\mathcal{R}_k$, we have:

$$g_k(\mathbf{x}_1) = \mathbf{w}_k^T\mathbf{x}_1 + w_{k0} > g_l(\mathbf{x}_1),\ \ l \neq k \tag{4.25}$$

and

$$g_k(\mathbf{x}_2) = \mathbf{w}_k^T\mathbf{x}_2 + w_{k0} > g_l(\mathbf{x}_2),\ \ l \neq k. \tag{4.26}$$

Let us now consider the point $\mathbf{z} = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$. For any discriminant function l, we have:

$$\begin{aligned} g_l(\mathbf{z}) &= \mathbf{w}_l^T\mathbf{z} + w_{l0} \\ &= \mathbf{w}_l^T[\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2] + w_{l0} \\ &= \lambda(\mathbf{w}_l^T\mathbf{x}_1 + w_{l0}) + (1-\lambda)(\mathbf{w}_l^T\mathbf{x}_2 + w_{l0}). \end{aligned} \tag{4.27}$$

Eq. 4.26 from Eqs. 4.25 and 4.26 becomes:

$$\begin{aligned} g_l(\mathbf{z}) &= \lambda(\mathbf{w}_l^T\mathbf{x}_1 + w_{l0}) + (1-\lambda)(\mathbf{w}_l^T\mathbf{x}_2 + w_{l0}) \\ &\leq \lambda(\mathbf{w}_k^T\mathbf{x}_1 + w_{k0}) + (1-\lambda)(\mathbf{w}_l^T\mathbf{x}_2 + w_{l0}) \\ &\leq \lambda(\mathbf{w}_k^T\mathbf{x}_1 + w_{k0}) + (1-\lambda)(\mathbf{w}_k^T\mathbf{x}_2 + w_{k0}) \\ &= \mathbf{w}_k^T[\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2] + w_{k0} \\ &= \mathbf{w}_k^T\mathbf{z} + w_{k0} \\ &= g_k(\mathbf{z}). \end{aligned} \tag{4.28}$$

That is, $\mathbf{z}$ belongs in the same region as $\mathbf{x}_1$ and $\mathbf{x}_2$. Since the above holds for any value of $0 \leq \lambda \leq 1$, region $\mathcal{R}_k$ is convex.

4.6 Discuss why if samples from two categories are distinct (i.e. no feature point is labelled by both categories), there always exists a nonlinear mapping to a higher dimension that leaves the points linearly separable.

Solution
Let us denote the samples by $\mathbf{x}_i\mathbb{R}^D,\ i = 1, \ldots, N$ and let us assume that the first N_1 of them belong to class c_1 and the remaining $N_2 = N - N_1$ belong to class c_2. We will approach this problem from two points of view. Let us assume that there is a nonlinear mapping from $\mathbb{R}^D$ to $\mathbb{R}^N$ using a nonlinear function $\phi(\cdot)$ such that the matrix $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)] \in \mathbb{R}^N$ is nonsingular. Then, by assigning a target value equal to $t_i = 1$ for all samples in c_1 and a target value equal to $t_i = -1$ for all samples in c_2, the following LMS problem:

$$\mathcal{J}(\mathbf{w}) = \|\mathbf{w}^T\mathbf{\Phi} - \mathbf{t}^T\|_2^2 \tag{4.29}$$

is solved by:

$$\mathbf{w} = \mathbf{\Phi}^{-1}\mathbf{t}^T. \tag{4.30}$$

This means, that there exists a decision function passing from the origin that can correctly classify all vectors $\phi(\mathbf{x}_i)$.
Another way to show this is to define a specific mapping as follows:

- we augment the vectors $\mathbf{x}_i$ belonging to class c_1 with a value of b, i.e. we set $\tilde{\mathbf{x}}_i = [\mathbf{x}_i^T\ b]^T$ and
- we augment the vectors $\mathbf{x}_i$ belonging to class c_1 with a value of $-b$, i.e. we set $\tilde{\mathbf{x}}_i = [\mathbf{x}_i^T\ -b]^T$.

Then, the decision function $f(\tilde{\mathbf{x}}) = \mathbf{w}^T\tilde{\mathbf{x}}$ can classify all vectors $\tilde{\mathbf{x}}_i$ if $\mathbf{w} = [001]^T$.

4.7 Suppose that for each training point $\mathbf{x}_i$ in class c_1 there exists a (symmetric) point in class c_2 equal to $-\mathbf{x}_i$. Prove that the separating hyperplane or LMS solution passes through the origin.

Solution

The LMS algorithm minimizes $J_s(\mathbf{a}) = \|\mathbf{Ya}-\mathbf{b}\|^2$ and we are given the problem

$$J_s(\mathbf{a}) = \left\| \begin{bmatrix} \mathbf{1}_n & \mathbf{Y}_1 \\ \mathbf{1}_n & \mathbf{Y}_1 \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{a}_r \end{bmatrix} - \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \right\|^2$$

We assume we have $2n$ data points. Moreover, we let $\mathbf{b}_i = (b_i^1 \;\; b_i^2 \;\; \ldots b_i^n)^t$ for $i = 1, 2$ be arbitrary margin vectors of size n each, and $\mathbf{Y}_1$ is an n-by-2 matrix containing the data points as the rows, and $\mathbf{a} = (a_0, a_1, a_2)^t = (a_0, \mathbf{a}_r)^t$. Then we have

$$\begin{aligned} J_s(\mathbf{a}) &= \left\| \begin{matrix} a_0\mathbf{1}_n + \mathbf{Y}_1\mathbf{a}_r - \mathbf{b}_1 \\ -a_0\mathbf{1}_n + \mathbf{Y}_1\mathbf{a}_r - \mathbf{b}_1 \end{matrix} \right\|^2 \\ &= \sum_{i=1}^{n}(a_0 + \mathbf{y}_i^t\mathbf{a}_r - b_1^i)^2 + \sum_{i=1}^{n}(-a_0 + \mathbf{y}_i^t\mathbf{a}_r - b_2^i)^2 \\ &= \sum_{i=1}^{n}(a_0 - b_1^i)^2 + \sum_{i=1}^{n}(-a_0 - b_2^i)^2 + 2\sum_{i=1}^{n}\mathbf{y}_i^t\mathbf{a}_r(a_0 - b_1^i - a_0 - b_2^i) \\ &= \sum_{i=1}^{n}(a_0 - b_1^i)^2 + \sum_{i=1}^{n}(a_0 + b_2^i)^2 - 2\sum_{i=1}^{n}\mathbf{y}_i^t\mathbf{a}_r(b_1^i + b_2^i). \end{aligned}$$

Thus $\partial J_s/\partial a_0 = 0$ implies $a_0 = 0$ and the minimum of J must be at $(0, \mathbf{a}_r)^t$ for some $\mathbf{a}_r$. Hence we showed that $a_0 = 0$ which tells us that the separating plane must go through the origin.

Chapter 5

Kernel-based Learning

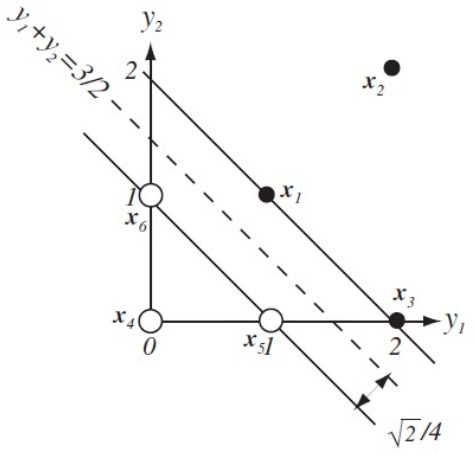
5.1 Two classes formed by the following samples:

$$c_1 = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & 0 \end{bmatrix} \tag{5.1}$$

$$c_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.2}$$

Plot these six points and construct by inspection the decision function of the Support Vector Machine classifier. Draw the weight vector, the optimal margin and the support vectors.

Solution



5.2 Show why the kernel matrix $\mathbf{K}$ needs to be positive definite in order the SVM classifier to have a solution.

Solution
The dual problem of SVM has the form:

$$\mathcal{J}_{\mathbf{a}} = \mathbf{a}^T(\mathbf{l}\mathbf{l}^T \circ \mathbf{K})\mathbf{a} + \mathbf{a}^T\mathbf{1} \tag{5.3}$$

The first derivative of $\mathcal{J}_{\mathbf{a}}$ with respect to $\mathbf{a}$ is:

$$\frac{\vartheta \mathcal{J}_{\mathbf{a}}}{\vartheta \mathbf{a}} = 2(\mathbf{l}\mathbf{l}^T \circ \mathbf{K})\mathbf{a} + \mathbf{1} \tag{5.4}$$

and the second is:

$$\frac{\vartheta^2 \mathcal{J}_{\mathbf{a}}}{\vartheta \mathbf{a}^2} = 2(\mathbf{l}\mathbf{l}^T \circ \mathbf{K}). \tag{5.5}$$

Thus, in order for $\mathcal{J}_{\mathbf{a}}$ to have a global maximum with respect to $\mathbf{a}$, $(\mathbf{l}\mathbf{l}^T \circ \mathbf{K})$ should be positive semi-definite. The matrix $\mathbf{l}\mathbf{l}^T$ is positive semi-definite, since for any $\mathbf{v}$ we have:

$$\mathbf{v}^T(\mathbf{l}\mathbf{l}^T)\mathbf{v} = (\mathbf{v}^T\mathbf{l})(\mathbf{l}^T\mathbf{v}) = c^2 \geq 0. \tag{5.6}$$

Thus, in order for the matrix $(\mathbf{l}\mathbf{l}^T \circ \mathbf{K})$ to be positive semi-definite, $\mathbf{K}$ needs to be positive semi-definite.

5.3 Show that the linear SVM classifier always finds the optimal hyper-plane in the linear case.

Solution
As shown above, if $\mathbf{K}$ is positive semi-definite, SVM can find the optimal $\mathbf{a}$ maximizing $\mathcal{J}_{\mathbf{a}}$. Thus, we need to show that the kernel function defined for the linear case is positive semi-definite. A matrix is positive semi-definite when:

$$\mathbf{v}^T\mathbf{K}\mathbf{v} \geq 0 \tag{5.7}$$

for any vector $\mathbf{v}$. For the linear kernel we have:

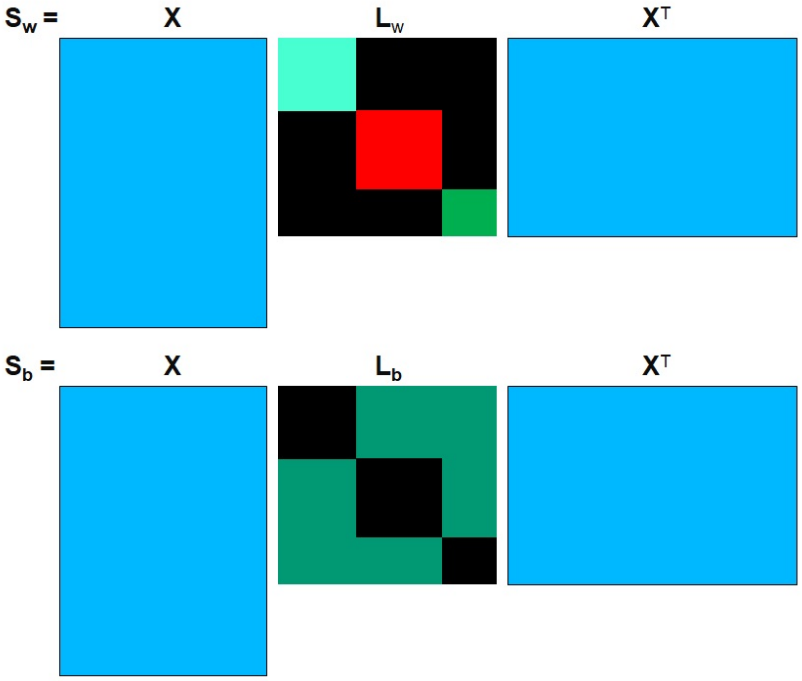
$$\begin{aligned} \mathbf{v}^T\mathbf{K}\mathbf{v} &= \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v} \\ &= (\mathbf{v}^T\mathbf{X}^T)(\mathbf{X}\mathbf{v}) \\ &= \mathbf{q}^T\mathbf{q} = \|\mathbf{q}\|_2^2 \geq 0. \end{aligned} \tag{5.8}$$

Thus, in order for $\mathcal{J}_{\mathbf{a}}$ to have a global maximum with respect to $\mathbf{a}$, $\mathbf{K}$ should be positive semi-definite.

5.4 Using Eqs. (5.34) and (5.35) imagine how KDA can be extended in order to exploit any type of pair-wise relationships between the data of a class (for $\mathbf{S}_w$) and between data of different classes (for $\mathbf{S}_b$). In order to do this, try to visualize the 'shape' of each matrix.

Solution

Let us assume that the data are ordered according to their class label. That is, the first N_1 samples belong to class c_1, the next N_2 samples belong to class c_2, etc. Then, $\mathbf{S}_w$ and $\mathbf{S}_b$ have the following form:



where the black regions in both $\mathbf{L}_w$ and $\mathbf{L}_b$ correspond to zeros. As we can see, $\mathbf{L}_w$ includes non-zero values in the regions corresponding to the samples if each class. In $\mathbf{L}_w$ these values are $\mathbf{L}_{ij} = \frac{1}{N_k}$ for $l_i = c_k$ and $l_j = c_k$. Changing these values, we can indicate different pair-wise relationships between each pair of samples of a class. For $\mathbf{L}_b$ we see that opposite (i.e. it models relationships between samples of different classes).

Chapter 6

Multilayer Neural Networks

6.1 Show that if an activation function of a three-layer neural network is linear, then this network is equivalent to a two-layer network. Based on that, explain why a three-layer neural network with linear activation functions cannot define nonlinear decision functions

Solution:

1. Consider a three-layer network with linear units throughout, having input vector $\mathbf{x}$, vector at the hidden units $\mathbf{y}$, and output vector $\mathbf{z}$. For such a linear system we have $\mathbf{y} = \mathbf{W}_1\mathbf{x}$ and $\mathbf{z} = \mathbf{W}_2\mathbf{y}$ for two matrices $\mathbf{W}_1$ and $\mathbf{W}_2$. Thus we can write the output as

$$\begin{aligned} \mathbf{z} &= \mathbf{W}_2\mathbf{y} = \mathbf{W}_2\mathbf{W}_1\mathbf{x} \\ &= \mathbf{W}_3\mathbf{x} \end{aligned}$$

for some matrix $\mathbf{W}_3 = \mathbf{W}_2\mathbf{W}_1$. But this equation is the same as that of a two-layer network having connection matrix $\mathbf{W}_3$. Thus a three-layer network with linear units throughout can be implemented by a two-layer network with appropriately chosen connections.

Clearly, a non-linearly separable problem cannot be solved by a three-layer neural network with linear hidden units. To see this, suppose a non-linearly separable problem can be solved by a three-layer neural network with hidden units. Then, equivalently, it can be solved by a two-layer neural network. Then clearly the problem is linearly separable. But, by assumption the problem is only non-linearly separable. Hence there is a contradiction and the above conclusion holds true.

6.2 Consider a three-layer neural network with D input neurons, L neurons in the hidden layer and K output neurons.

1. How many weights are in the network?
2. Compute the number of operations that need to be performed in order to classify a vector $\mathbf{x}$.

.

Solution

The neural network has two weight matrices:

- $\mathbf{W}_1 \in \mathbb{R}^{D \times L}$ connecting the input layer to the hidden layer
- $\mathbf{W}_2 \in \mathbb{R}^{L \times K}$ connecting the hidden layer to the output layer

Thus, it has $O = DL + LK$ weights.

We need $DL + LK$ multiplications and $D(L-1) + L(K-1)$ additions.

6.3 Use Eq. (6.2) to show why the input-to-hidden layer weights must be different from each other (e.g. initialized random) or else learning cannot proceed well.

Solution

Suppose the input to hidden weights are set equal to the same value, say w_o, then $w_{ij} = w_o$. Then we have

$$net_j = f(net_j) = \sum_{i=1}^{d} w_{ji} x_i = w_o \sum_i x_i = w_o \mathbf{x}.$$

This means that $o_j = f(net_j)$ is constant, say y_o. Clearly, whatever the topology of the original network, setting the w_{ji} to be a constant is equivalent to changing the topology so that there is only a single input unit, whose input to the next layer is x_o. As, a result of this loss of one-layer and number of input units in the next layer, the network will not train well.

6.4 Write the algorithm for optimizing the parameters of a three-layer neural network trained using $\mathbf{x}_i, i = 1, \ldots, N$ and the corresponding labels $l_i \in \{c_1, \ldots, c_K\}$.

Solution
Algorithm 1.

6.5 Write the algorithm for optimizing the parameters of an RBF network trained using $\mathbf{x}_i,\ i = 1, \ldots, N$ and the corresponding labels $l_i \in \{c_1, \ldots, c_K\}$.

Solution
Algorithm 2.

6.6 Show that an RBF network trained on $\mathbf{x}_i,\ i = 1, \ldots, N$ can approximate any continuous function (which is defined by N sampled values $t_i,\ i = 1, \ldots, N$). How can this be extended to any three-layer feedforward neural network?

Solution
If we use N RBF neurons and select i.i.d. prototypes, then the hidden layer output matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$ is invertible. Thus, the weight vector $\mathbf{w} = \mathbf{H}^{\dagger}\mathbf{t}$ can map $\mathbf{h}_i$ to t_i with zero error.

If we use a linear activation function for the output neuron, N hidden layer neurons with an activation function and weights generating a matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$ that is invertible, this network can also map $\mathbf{h}_i$ to t_i with zero error.

Algorithm 1: Three layer neural network training

Initialize the network and its weights $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, η, θ, $t = 0$
Do $t \leftarrow t + 1$
Shuffle the data and set $i = 0$
Set $\mathcal{J}(t) = 0$
  Do $i \leftarrow i + 1$
    Select (randomly) a vector $\mathbf{x}_i$
    Calculate $\mathbf{o}_i$ (the network's output for $\mathbf{x}_i$) using Eq. (6.3)
    Update $\mathcal{J}(t) \leftarrow \mathcal{J}(t) + \frac{1}{N}\mathcal{J}_i(t)$
    // Update the output layer weights
    For every $\{j, k\}$-pair $j = 1, \ldots, L$, $k = 1, \ldots, K$
      Calculate $\delta_j^{(2)}$ from Eq. (6.18)
      Calculate $\frac{\vartheta \mathcal{J}}{\vartheta W_{jk}^{(2)}}$ using Eq. (6.13)
      Update $W_{jk}^{(2)} \leftarrow W_{jk}^{(2)} - \eta \frac{\vartheta \mathcal{J}}{\vartheta W_{jk}^{(2)}}$
    // Update the hidden layer weights
    For every $\{i, j\}$-pair $i = 1, \ldots, D$, $j = 1, \ldots, l$
      Calculate $\frac{\vartheta \mathcal{J}}{\vartheta W_{jk}^{(1)}}$ using Eq. (6.19)
      Update $W_{jk}^{(1)} \leftarrow W_{jk}^{(1)} - \eta \frac{\vartheta \mathcal{J}}{\vartheta W_{jk}^{(1)}}$
    // Stop when no better solution
  until $i = N$
until $(\mathcal{J}(t) - \mathcal{J}(t = 1) < \theta$

Algorithm 2: RBF networks

Create target vectors $\mathbf{t}_i,\ i = 1, \ldots, N$ with values $t_{ik} = 1$ if $l_i = c_k$ and $t_{ik} = 0$ if $l_i \neq c_k$
Initialize the prototypes $\mathbf{z}_l,\ l = 1, \ldots, L$ of the L RBF neurons by clustering $\mathbf{x}_i,\ i = 1, \ldots, N$ in L clusters
Set the values $\sigma_l = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{z}_l\|_2^2$
Calculate the hidden layer output vectors $\mathbf{h}_i = \text{RBF}(\mathbf{x}_i),\ i = 1, \ldots, N$
Calculate the output weights as $\mathbf{W} = \mathbf{H}^{\dagger}\mathbf{T}^T$