02457 Non-Linear Signal Processing: Exercise 3

This exercise is based on C.M. Bishop: *Neural Networks for Pattern Recognition*, chapter 3.

Your task is to use the MATLAB software to illustrate and discuss the linear model using single layer networks and Fisher's linear discriminant.

Print and comment on the figures produced by the software main3a.m to main3c.m as outlined below at the three **checkpoints**.

Linear Models

Let the function $y(\mathbf{x})$ be a function of the vector $\mathbf{x}$, where $\mathbf{x} = (x_1, \ldots, x_d)^\top$. The functional form of $y(\mathbf{x})$ is unknown, but we have a data-set, $\mathcal{D} = \{(\mathbf{x}^n, t^n)\}$, $n = 1, \ldots, N$ of N corresponding values of $\mathbf{x}$ and $y(\mathbf{x})$, and we wish to find a model of the function using the information in the data.

Let us model the function $y(\mathbf{x})$ with a linear model given by

$$y(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i \quad = \quad w_0 + \mathbf{w}^\top \mathbf{x}, \tag{1}$$

where $\mathbf{w}$ is a weight vector. This corresponds to restricting $y(\mathbf{x})$ to a d-dimensional hyperplane in $(d+1)$-dimensional space.

The constant term in equation (1) can be included in the weight vector, $\mathbf{w}$, where another term is also added to $\mathbf{x}$, such that $\mathbf{x} = (1, x_1, \ldots, x_d)^\top$. This reduces equation (1) to

$$y(\mathbf{x}) = \sum_{i=0}^{d} w_i x_i \quad = \quad \mathbf{w}^\top \mathbf{x}. \tag{2}$$

The weight-vector, $\mathbf{w}$, that models the given data-set (training-set) best is found through minimizing an error function. Here we shall use the sum-of-squares error function given by

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2} \sum_{n=1}^{N} \{y(\mathbf{x}^n; \mathbf{w}) - t^n\}^2 & (3) \\ &= \frac{1}{2} \sum_{n=1}^{N} \left\{\mathbf{w}^\top \mathbf{x}^n - t^n\right\}^2. & (4) \end{aligned}$$

Introducing the matrix, $\mathbf{X}$, where $\mathbf{X}^\top = (\mathbf{x}^1\ \mathbf{x}^2\ \ldots \mathbf{x}^N)$ and the vector, $\mathbf{t} = (t^1, t^2, \ldots, t^N)^\top$, equation (4) can be rewritten as

$$E(\mathbf{w}) = \frac{1}{2} \left(\mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w} + \mathbf{t}^\top \mathbf{t} - 2\mathbf{w}^\top \mathbf{X}^\top \mathbf{t}\right). \tag{5}$$

Since equation (5) is quadratic in $\mathbf{w}$, the exact value of $\mathbf{w}$ minimizing $E(\mathbf{w})$ can be found analytically by equating the derivative of equation (5) to zero. This gives the normal equations for the least-squares problem:

$$\mathbf{X}^\top \mathbf{X} \mathbf{w} = \mathbf{X}^\top \mathbf{t}. \tag{6}$$

Solving for $\mathbf{w}$ gives the optimal $\mathbf{w}$. Since $\mathbf{X}$ is an $N \times (d+1)$ matrix, $\mathbf{X}^\top \mathbf{X}$ is a $(d+1) \times (d+1)$ square matrix. Thus the solution to equation (6) is given by

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{t} \quad \equiv \quad \mathbf{X}^\dagger \mathbf{t}, \tag{7}$$

where $\mathbf{X}^\dagger$ is a $(d+1) \times N$ matrix known as the *pseudo-inverse* of $\mathbf{X}$. $\mathbf{X}^\dagger$ has the property that $\mathbf{X}^\dagger \mathbf{X} = \mathbf{I}$, whereas $\mathbf{X}\mathbf{X}^\dagger \neq \mathbf{I}$ in general.

Checkpoint 3.1:

Use the program main3a.m to create a training-set with a 2-dimensional input variable and a 1-dimensional output variable. Compare the estimated weight vector with the true one and the dependence on both the noise level and number of points in the training-set.

Time Series Prediction

An example where the linear model can be used is in time series prediction. To illustrate this, consider the example of the sunspot measurements. The number of sunspots oscillates almost periodically over a period of some years. The average number of sunspots has been measured yearly since 1700. Imagine we want to predict the average number of sunspots next year. The linear model can be used for this.

Let the number of sunspots in year n be x_n. Let's assume that the number of sunspots in year n only depends on the number of sunspots in the previous d years. This is reasonable since the there must be a limit as to how far back one can expect a correlation. This can be expressed as

$$x_n = f(x_{n-1}, x_{n-2}, \ldots x_{n-d}). \tag{8}$$

Approximating the function f with a linear model gives

$$x_n = w_0 + \sum_{j=1}^{d} w_j x_{n-j}. \tag{9}$$

This corresponds to equation (1), and hence is the same problem given by equations (2) to (7), where the training set is given by

$$\left.\begin{array}{rcl} \mathbf{x}^n & = & (1, x_{n-d}, \ldots, x_{n-1})^\top \\ t^n & = & x_n \end{array}\right\} \quad n = 1, \ldots, N-d-1. \tag{10}$$

The weights can be found using equation (7), and the predicted value, x_{n+1}, can be found from

$$x_{n+1} = y(\mathbf{x}^n) = \mathbf{w}^\top \mathbf{x}^n. \tag{11}$$

Checkpoint 3.2:

Use the program main3b.m to perform a time series prediction of the number of sunspots. Compare the actual measurements with the predicted values as a function of the number of weights, d, (hence years) included in the model.

Fisher's Linear Discriminant

In exercise 2, we saw that a multidimensional variable can be projected onto the directions of largest covariance by a coordinate transformation to the coordinate system spanned by the eigenvectors of the covariance matrix. This may facilitate classification of the data. However, there are also some cases, where the direction that maximizes class separation doesn't correspond to any of the eigenvectors. In such a case, the coordinate transformation does not solve the problem. However, the direction of maximum class separation can be found using Fisher's linear discriminant.

Consider a two-class problem in which there are N_1 points of class C_1 and N_2 points of class C_2. The mean vectors of the two classes are given by

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in C_1} \mathbf{x}^n \tag{12}$$

$$\mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in C_2} \mathbf{x}^n. \tag{13}$$

Let the projection of a data vector, $\mathbf{x}$, onto a the direction of maximum class separation be

$$y = \mathbf{w}^\top \mathbf{x}. \tag{14}$$

This is the direction along which the probability density functions of the two classes, $p(y|C_1)$ and $p(y|C_2)$, overlap the least. It can be shown by maximizing Fisher's criterion (equation (3.82) in *Bishop*) that the direction vector for the projection, $\mathbf{w}$, is given by

$$\mathbf{w} \propto \mathbf{S}_w^{-1}(\mathbf{m}_2 - \mathbf{m}1), \tag{15}$$

where $\mathbf{S}_w$ is the total within-class covariance matrix, given by

$$\mathbf{S}_w = \sum_{n \in C_1} (\mathbf{x}^n - \mathbf{m}_1)(\mathbf{x}^n - \mathbf{m}_1)^\top + \sum_{n \in C_2} (\mathbf{x}^n - \mathbf{m}_2)(\mathbf{x}^n - \mathbf{m}_2)^\top. \tag{16}$$

Checkpoint 3.3:

Use the program main3c.m to find the direction maximizing class separation for a two-class problem. Compare the projection of the data-set onto one-dimension with the projections found using eigenvector transformation as illustrated in exercise 2. Compose different data-sets and compare the performance of the two methods in each case.

Challenge:

Modify the program main3b.m to predict the number of sunspots two or more years ahead. Compare the performance with the case of one year ahead.

DTU, September 1999,

Karam Sidaros