Lecture Outline

Reading: Chapter 4 - Linear Models for Classification

This lecture covers

- Least Squares for Classification
- Fisher's Linear Discriminant
- Perceptron Algorithm

Notation

Input data vectors $\mathbf{x}$ are column vectors. We collect input vectors into an "observation" or data matrix where they are stored as row vectors

$$\mathbf{X} = \begin{bmatrix} -\mathbf{x}_1^T- \\ -\mathbf{x}_2^T- \\ \vdots \\ -\mathbf{x}_N^T- \end{bmatrix} \tag{1}$$

Likewise, 1-of-K-coded target vectors $\mathbf{t}$ are column vectors and we collect as

$$\mathbf{T} = \begin{bmatrix} -\mathbf{t}_1^T- \\ -\mathbf{t}_2^T- \\ \vdots \\ -\mathbf{t}_N^T- \end{bmatrix} \tag{2}$$

4.1.3 Least Squares for Classification

Consider a general classification problem with K classes, with a 1-of-K binary coding scheme for the target vector $\mathbf{t}$. Each class $\mathcal{C}_k$ is described by its own linear model so that

$$\begin{aligned} y_k(\mathbf{x}) &= \mathbf{w}_k^T\mathbf{x} + w_{k0} \\ &= \tilde{\mathbf{w}}_k^T\tilde{\mathbf{x}} \end{aligned} \tag{3}$$

where $k = 1, \ldots, K$ and the augmented vectors are $\tilde{\mathbf{w}}_k = [w_{k0}, w_{k1}, \ldots, w_{kD}]^T$ and $\tilde{\mathbf{x}} = [1, \mathbf{x}^T]^T$. We can conveniently group these class models together using vector notation

$$\mathbf{y}(\mathbf{x}) = \tilde{\mathbf{W}}^T\tilde{\mathbf{x}} \tag{4}$$

where $\tilde{\mathbf{W}}$ is a matrix whose kth column comprises the $(D+1)$-dimensional vector $\tilde{\mathbf{w}}_k$

$$\tilde{\mathbf{W}} = \begin{bmatrix} \tilde{\mathbf{w}}_1 & \tilde{\mathbf{w}}_2 & \ldots & \tilde{\mathbf{w}}_K \end{bmatrix}. \tag{5}$$

Once $\tilde{\mathbf{W}}$ is determined, new input $\mathbf{x}'$ is then assigned to the class for which the output is $y_k = \tilde{\mathbf{w}}_k^T\tilde{\mathbf{x}}'$ largest.

Solving for $\tilde{\mathbf{W}}$

We arrange a training data set $\{\mathbf{x}_n, \mathbf{t}_n\}$, as rows in the following observation and target matrices

$$\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{\mathbf{x}}_1^T \\ \tilde{\mathbf{x}}_2^T \\ \vdots \\ \tilde{\mathbf{x}}_N^T \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \mathbf{t}_2^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix} \tag{6}$$

i.e. $\tilde{\mathbf{X}}$ is a matrix whose nth row is $\tilde{\mathbf{x}}_n^T$ and $\mathbf{T}$ is a matrix whose nth row is the 1-of-K coded target vector $\mathbf{t}_n^T$.

We want to find $\tilde{\mathbf{W}}$ such that sum of the target error vector magnitudes is minimized

$$E_D(\tilde{\mathbf{W}}) = \frac{1}{2}\sum_{n=1}^{N}\left(\tilde{\mathbf{W}}^T\tilde{\mathbf{x}} - \mathbf{t}_n\right)^T \underbrace{\left(\tilde{\mathbf{W}}^T\tilde{\mathbf{x}} - \mathbf{t}_n\right)}_{\text{error vector}}. \tag{7}$$

Equation (7) can be compactly rewritten in terms of matrix Trace [the *trace* of a matrix is defined as $\mathrm{Tr}(\mathbf{A}) = \sum_{i=1}^{N}\mathbf{A}_{ii}$]

$$E_D(\tilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left[\left(\tilde{\mathbf{X}}\tilde{\mathbf{W}} - \mathbf{T}\right)^T\left(\tilde{\mathbf{X}}\tilde{\mathbf{W}} - \mathbf{T}\right).\right] \tag{8}$$

We now take the derivative of $E_D(\tilde{\mathbf{W}})$ with respect to $\tilde{\mathbf{W}}$, using the relation

$$\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}\left[(\mathbf{A}\mathbf{X}+\mathbf{B})^T(\mathbf{A}\mathbf{X}+\mathbf{B})\right] = 2\mathbf{A}^T(\mathbf{A}\mathbf{X}+\mathbf{B}) \tag{9}$$

where $\mathbf{A} = \tilde{\mathbf{X}}$, $\mathbf{X} = \tilde{\mathbf{W}}$, and $\mathbf{B} = -\mathbf{T}$, and equate to $\mathbf{0}$ to get

$$\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\tilde{\mathbf{W}} - \tilde{\mathbf{X}}^T\mathbf{T} = \mathbf{0}. \tag{10}$$

Solving for $\tilde{\mathbf{W}}$ leads to

$$\begin{aligned} \tilde{\mathbf{W}} &= (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^T\mathbf{T} \\ &= \tilde{\mathbf{X}}^\dagger\mathbf{T}. \end{aligned} \tag{11}$$

With the model parameters, the discriminant function is then

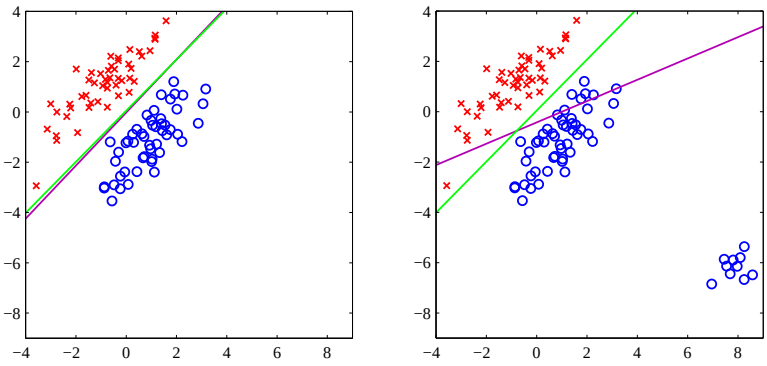
$$\mathbf{y}(\mathbf{x}) = \tilde{\mathbf{W}}^T\tilde{\mathbf{x}} \tag{12}$$

and we classify according to

$$\mathcal{C}_k = \arg\max_{1\le j\le K}\mathbf{y}_j(\mathbf{x}). \tag{13}$$

Example: Suppose $\mathbf{y}(\mathbf{x}) = [0, 0.2, 0.9, 0.3]^T$. We decide that $\mathbf{x}$ is from class $\mathcal{C}_3$

The LS linear discriminant function suffers from some severe problems including robustness to outliers as illustrated in Fig. 1. Here we see that the additional data points in the right-hand side figure produce significant change in the location of the decision boundary, even though these points would be correctly classified by the original boundary in the left-hand figure.



(a) (b)

Figure 1:

4.1.4 Fisher's Linear Discriminant

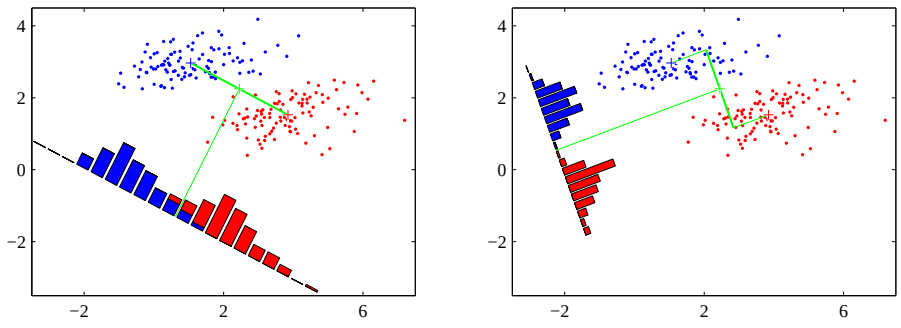
One way to view linear classification is in terms of *dimensionality reduction* or DR. Consider first the case of two classes and suppose we take the D-dimensional input $\mathbf{x}$ and project it down to one dimension using

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

If we place a threshold on y and classify $y \geq -w_0$ as class $\mathcal{C}_1$ and otherwise class $\mathcal{C}_2$, then we obtain our standard linear classifier. Recall that if $\mathbf{x}$ is on the decision surface then $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = 0$ and thus if $\mathbf{x} \in \mathcal{C}_1$, then $\mathbf{w}^T\mathbf{x} \geq -w_0$.

In general, the projection onto one dimension leads to a considerable loss of information, and classes that are well separated in the original D-dimensional space may become strongly overlapping in the 1-D space. However, by adjusting $\mathbf{w}$, we can select a projection that maximizes the class *separation*.

The idea proposed by Fisher is to 1) maximize a function that will give a large separation between the *projected class means* while also giving 2) a small variance within each class, thereby *minimizing the class overlap*. This is illustrated in Fig. 2



(a) (b)

Figure 2:

To begin consider a two-class problem in which there are N_1 points of class $\mathcal{C}_1$ and N_2 points of class $\mathcal{C}_2$ and

the mean vectors of the two classes are given by

$$\begin{aligned} \mathbf{m}_1 &= \frac{1}{N_1}\sum_{n\in\mathcal{C}_1}\mathbf{x}_n \\ \mathbf{m}_2 &= \frac{1}{N_2}\sum_{n\in\mathcal{C}_2}\mathbf{x}_n. \end{aligned} \tag{15}$$

The simplest measure of the separation of classes, when projected onto $\mathbf{w}$ is the separation of the projected class means, m_1 and m_2. This suggests we might choose $\mathbf{w}$ so as to maximize

$$(m_2 - m_1)^2 = \left[\mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1)\right]^2 \tag{16}$$

where $m_k = \mathbf{w}^T\mathbf{m}_k$ is the mean of the projected data from class $\mathcal{C}_k$, i.e. projected class mean.

The within-class variance of the transformed (projected) data from class $\mathcal{C}_k$ is

$$s_k^2 = \sum_{n\in\mathcal{C}_k}(y_n - m_k)^2 \tag{17}$$

where $y_n = \mathbf{w}^T\mathbf{x}_n$. We can define the total within-class variance for the whole data set to be simply $s_1^2 + s_2^2$.

The Fisher criterion is defined to be the ratio of the between-class variance (which we want to maximize) to the within-class variance (which we want to minimize) and is given by

$$\begin{aligned} J(\mathbf{w}) &= \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \\ &= \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}} \end{aligned} \tag{18}$$

where the *between-class* covariance matrix is given by

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T \tag{19}$$

and the total *within-class* covariance matrix is given by

$$\mathbf{S}_W = \sum_{n\in\mathcal{C}_1}(\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n\in\mathcal{C}_2}(\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T. \tag{20}$$

Differentiating (18) with respect to $\mathbf{w}$ we find that $J(\mathbf{w})$ is maximized when

$$(\mathbf{w}^T\mathbf{S}_B\mathbf{w})\mathbf{S}_W\mathbf{w} = (\mathbf{w}^T\mathbf{S}_W\mathbf{w})\mathbf{S}_B\mathbf{w}. \tag{21}$$

This leads us to (ignoring scale factors)

$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1). \tag{22}$$

This result is known as *Fisher's linear discriminant*, although strictly it is not a discriminant but rather a specific choice of direction of projection of the data down to one dimension. However, the projected data can subsequently be used to construct a discriminant, by choosing a threshold y_0 so that

$$\begin{cases} \mathbf{w}^T\mathbf{x} \geq y_0, & \mathcal{C}_1 \\ \mathbf{w}^T\mathbf{x} < y_0, & \mathcal{C}_2 \end{cases} \tag{23}$$

Fisher's discriminant for multiple classes is covered in Section 4.1.6.

4.1.7 Perceptron Algorithm

The perceptron corresponds to a two-class model in which the input vector $\mathbf{x}$ is first transformed using a fixed nonlinear transformation to give $\boldsymbol{\phi}(\mathbf{x})$, and then this is used to construct the generalized linear model of the form

$$y(\mathbf{x}) = f\left(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})\right) \tag{24}$$

where the nonlinear activation function $f(\cdot)$ is given by a step function of the form

$$f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases} \tag{25}$$

For the perceptron it is more convenient to use target values $t = +1$ for class $\mathcal{C}_1$ and $t = -1$ for class $\mathcal{C}_2$ which matches the choice of the activation function.

We now consider an error function known as the *perceptron criterion*. To derive this, we note that we are seeking a weight vector $\mathbf{w}$ such that data $\mathbf{x}_n$ in class $\mathcal{C}_1$ will have $\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) > 0$ whereas data $\mathbf{x}_n$ in class $\mathcal{C}_2$ will have $\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) < 0$. Using the $t \in \{-1, +1\}$ target coding scheme it follows that we want $\mathbf{w}$ such that all data satisfies $\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)t_n > 0$, i.e.

$$\mathcal{C}_1 : \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) > 0, \ t = +1 \tag{26}$$

$$\mathcal{C}_2 : \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) < 0, \ t = -1 \tag{27}$$

The perceptron criterion, which we seek to minimize, is therefore given by

$$E_P(\mathbf{w}) = -\sum_{n\in\mathcal{M}} \mathbf{w}^T\boldsymbol{\phi}_n t_n \tag{28}$$

where $\boldsymbol{\phi}_n = \boldsymbol{\phi}(\mathbf{x}_n)$ and $\mathcal{M}$ denotes the set of all misclassified patterns. Note that the argument in the summation is negative since $\boldsymbol{\phi}_n$ is misclassified, i.e. an error.

We utilize a stochastic gradient descent algorithm to "search" for $\mathbf{w}$:

$$\begin{aligned} \mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} - \eta\nabla E_P(\mathbf{w}) \\ &= \mathbf{w}^{(\tau)} + \eta\boldsymbol{\phi}_n t_n \end{aligned} \tag{29}$$

where η is the learning rate parameter. The perceptron learning algorithm is illustrated in Fig. 3. See p. 194 for details.

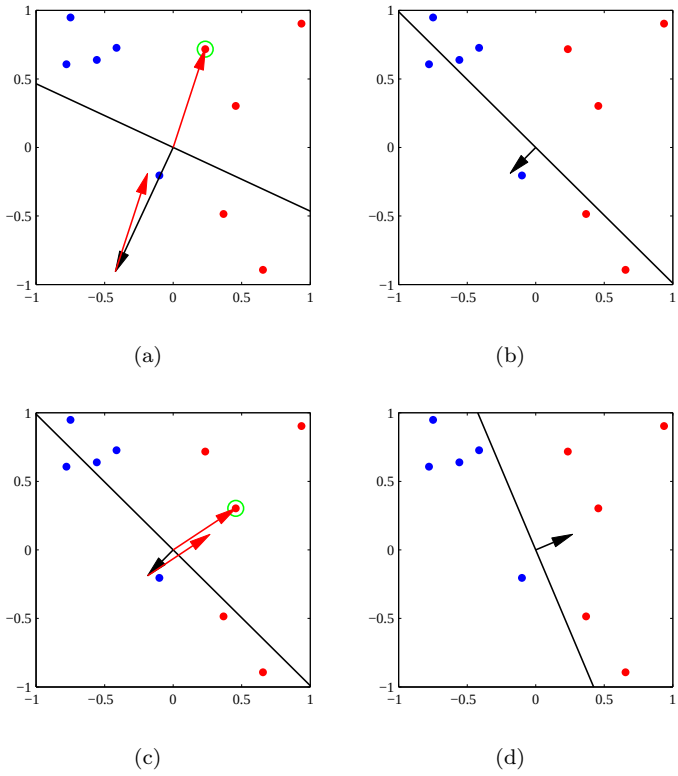


Figure 3: