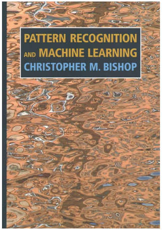
Linear Models for Classification

Bishop PRML Ch. 4 – 4.1

Asahi Tsuruo

June 3, 2020



Outline

Discriminant functions

Example in Hand-written Digit Recognition:

input vector target vector

$\mathbf{x}_i =$  $\mathbf{t}_i = \{0,0,0,0,0,0,0,1,0,0\}$ (4.1*)

- Each input vector $\mathbf{x}_i$ is classified into one of the K discrete classes
  - Denote classes C_k
- Represent input image as a vector $\mathbf{x}_i \in \mathbb{R}^{784}$
- Target vector $\mathbf{t}_i \in \mathbb{R}^{10}$
  - 1-of-K
- Given a training data set $\{(\mathbf{x}_n, \mathbf{t}_n)\}_{n=1}^{n=N}$, learning problem is to construct a "good" function $y(\mathbf{x})$ from these:
  - $y : \mathbb{R}^{784} \to \mathbb{R}^{10}$

Generalized linear models

- Similar to previous chapter on linear models for regression, we'll use a "linear" model for classification:

$$y(\mathbf{x}) = f(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0) \tag{4.3}$$

- This (4.3) is called a generalized linear model
- $f(a)$ is an activate function (a nonlinear function)
  - e.g.

$$f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases} \tag{4.53}$$

- Decision boundary between classes will be linear function of $\mathbf{x}$
- Can also apply non-linearity to $\mathbf{x}$, as in $\phi_i(\mathbf{x})$ for regression

Outline

Discriminant Functions in binary classification

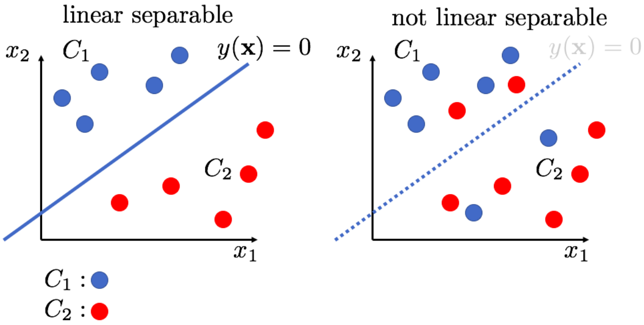
- A simple linear discriminant function:

$$y(\mathbf{x}) = \sum_{j=1}^{D} w_j x_j + w_0$$
$$= \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 \qquad (4.4)$$

$\mathbf{w}$: weight vector, $\mathbf{x}$: input vector, w_0: bias, D: input space dimension

- An input vector $\mathbf{x}$ is assigned to
  - class C_1 if $y(\mathbf{x}) \geq 0$
  - class C_2 otherwise
- Decision boundary:
  - $y(\mathbf{x}) = 0$
  - $(D-1)$ dimensional hyperplane within input space D

Example of lener separable and $D - 1$ hyperplane

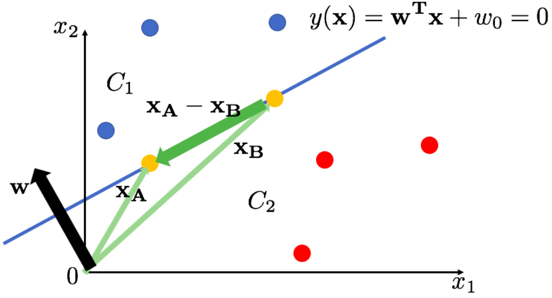


- Input space (planes): $2\ D$
- Decision boundary (lines): $(2-1)\ D = 1\ D$

If input space is $3D$ (solids), decision boundary is $2D$ (planes)

$\mathbf{w}$ determines the orientation of the decision boundary

$\mathbf{x_A}$ and $\mathbf{x_B}$ on the decision boundary

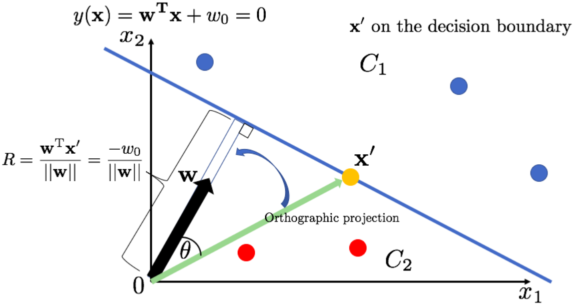


$$y(\mathbf{x_A}) - y(\mathbf{x_B}) = \mathbf{w^T}(\mathbf{x_A} - \mathbf{x_B}) = 0 \quad \text{(A1)}$$

- From an inner product property, $\mathbf{w}$ is orthogonal to every vector on the decision boundary
  - $\mathbf{w}$ determines the orientation of the decision boundary

The distance R from the origin to the decision boundary

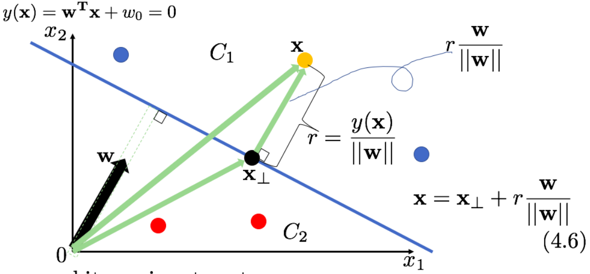
- Bias parameter w_0 determines the location of the decision boundary



$$R = ||\mathbf{e}||||\mathbf{x}'||\cos\theta = \mathbf{e}^{\mathbf{T}} \cdot \mathbf{x}' = \frac{\mathbf{w}^{\mathbf{T}}\mathbf{x}'}{||\mathbf{w}||} = \frac{-w_0}{||\mathbf{w}||} \tag{4.5*}$$

- $\mathbf{w}^{\mathbf{T}}/||\mathbf{w}|| = \mathbf{e}$ is a unit vector, $\mathbf{w}^{\mathbf{T}}\mathbf{x}' + w_0 = 0 \rightarrow \mathbf{w}^{\mathbf{T}}\mathbf{x}' = -w_0$

The perpendicular distance r between a x and the decision boundary



$\mathbf{x}$: arbitrary input vector

$\mathbf{x}_\perp$: orthogonal projection onto the decision boundary

Multiplying both sides of this result (4.6) by $\mathbf{w}^\mathbf{T}$ and adding w_0, and making use of $y(\mathbf{x}) = \mathbf{w}^\mathbf{T}\mathbf{x} + w_0$ and $y(\mathbf{x}_\perp) = \mathbf{w}^\mathbf{T}\mathbf{x}_\perp + w_0 = 0$

$$r = \frac{y(\mathbf{x})}{||\mathbf{w}||} \tag{4.7}$$

Augmented representation for binary classification

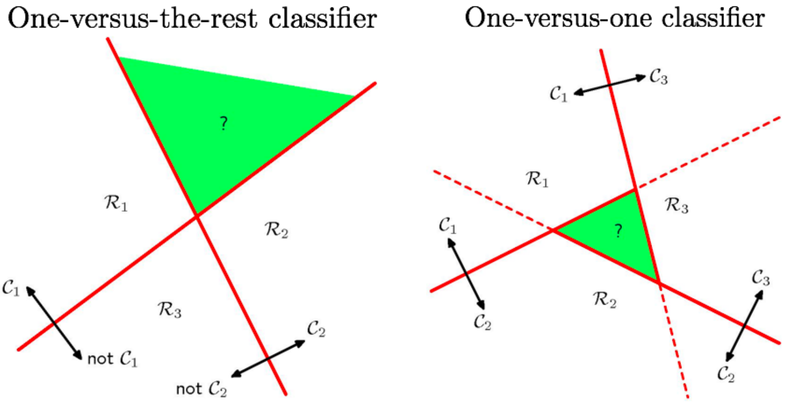
- $x_0 = 1, \hat{\mathbf{w}} = (w_0, \mathbf{w}), \hat{\mathbf{x}} = (x_0, \mathbf{x})$

$$
\begin{aligned}
y(\mathbf{x}) &= \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 && (4.4)\\
&= \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 * 1\\
&= \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 * x_0\\
&= \tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}} && (4.8)
\end{aligned}
$$

- The decision boundaries are D-dimensional hyperplanes passing through the origin of the $D+1$-dimensional expanded input space

Outline

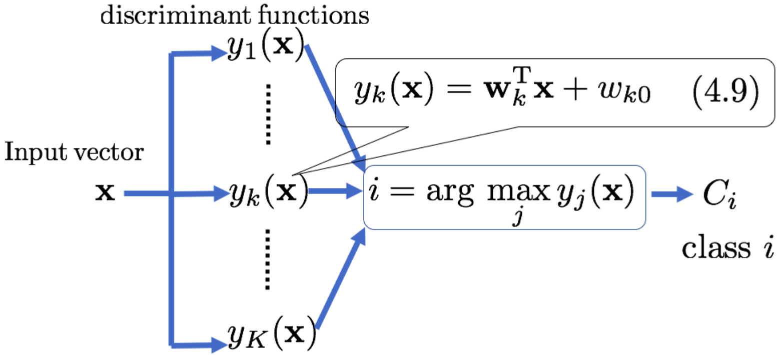
Discriminant Functions in Multiple Classes



- A linear discriminant function between two classes separates with a hyperplane
- How can we use this for multiple classes?
  - One-versus-the-rest classifier: build $K-1$ classifiers
  - One-versus-one classifier: build $K(K-1)/2$

Solution for ambiguous regions in Multiple Classes

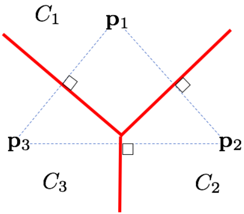
- Select the discriminant function that have a maximum output



- The decision boundary between C_k and C_j
  - Given by $y_k(\mathbf{x}) = y_j(\mathbf{x})$
  - Corresponds to a $D-1$ dimensional hyperplane

$$(\mathbf{w}_k - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{k0} - w_{j0}) = 0 \qquad (4.10)$$

Nearest Neighbour (NN) classifier is a specific case of linear discriminants in selection of max-value



decision boundary
bisecting perpendicular
$\mathbf{p}_k$ prototype of class C_k

decision rule :

$i = \arg\max_j y_j(\mathbf{x}) \Rightarrow \mathbf{x} \in C_k$

$$y_k(\mathbf{x}) = -||\mathbf{x} - \mathbf{p}_k||^2 = -||\mathbf{x}||^2 + 2\mathbf{p}_k^{\mathrm{T}}\mathbf{x} - ||\mathbf{p}_k||^2 \quad (4.4^*)$$

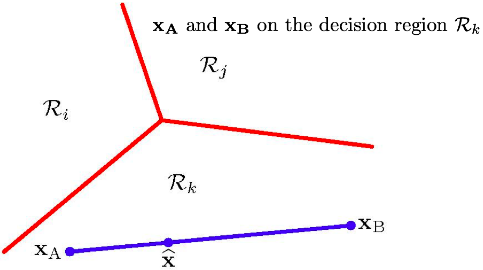
From NN classifier (4.4*), we can obtain the classifier $\mathbf{p}_k^{\mathrm{T}}\mathbf{x} - \frac{1}{2}||\mathbf{p}_k||^2$, thus,

$$y(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0} \tag{4.4}$$

where

$$\mathbf{w}_k = \mathbf{p}_k, \;\; w_0 = -\frac{1}{2}||\mathbf{p}_k||^2 \tag{B1}$$

The decision regions are connected and convex



- $\hat{\mathbf{x}}$: any point that lies on the line connecting $\mathbf{x_A}$ and $\mathbf{x_B}$
- $0 \leq \lambda \leq 1$

$$\hat{\mathbf{x}} = \lambda \mathbf{x}_A + (1 - \lambda)\mathbf{x}_B \tag{4.11}$$

$$y_k(\hat{\mathbf{x}}) = \lambda y_k(\mathbf{x}_A) + (1 - \lambda) y_k(\mathbf{x}_B) \tag{4.12}$$
$$\Rightarrow y_k(\hat{\mathbf{x}}) > y_j(\hat{\mathbf{x}}), \forall j \neq k$$

Outline

Augmented representation for multi-class classification

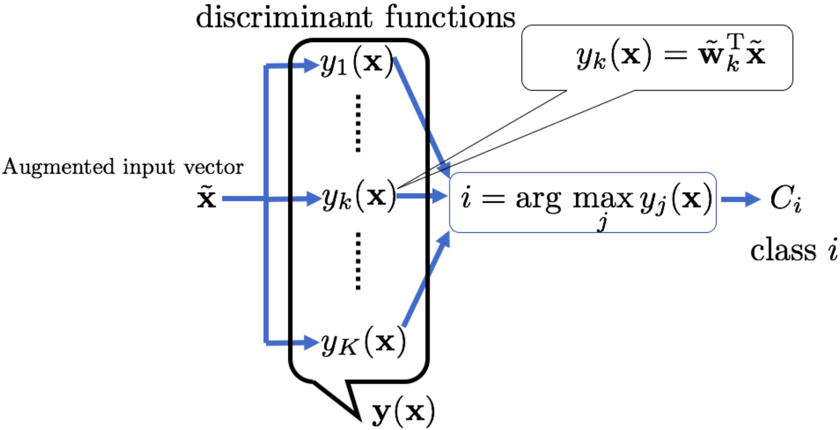
- The discriminant functions for each class C_k

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0} \quad (k = 1, ..., K) \tag{4.13}$$

$$\Downarrow$$

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\tilde{\mathbf{x}} \tag{4.14}$$

- $\widetilde{\mathbf{W}}$ is a matrix whose k^{th} column consists of the $D+1$ dimensional vector $\tilde{\mathbf{w}}_k = (w_{k0}, \mathbf{w}_k^{\mathbf{T}})^{\mathbf{T}}$
- $\tilde{\mathbf{x}}$ is the augmented input vector $(1, \mathbf{x}^{\mathbf{T}})^{\mathbf{T}}$



$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}} \tilde{\mathbf{x}} = \begin{pmatrix} w_{10} & \cdots & w_{1d} & \cdots & w_{1D} \\ \vdots & \ddots & & & \vdots \\ w_{k0} & & \ddots & & w_{kD} \\ \vdots & & & \ddots & \vdots \\ w_{K0} & \cdots & w_{Kd} & \cdots & w_{KD} \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_d \\ \vdots \\ x_D \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{w}}_1^{\mathbf{T}} \tilde{\mathbf{x}} \\ \vdots \\ \tilde{\mathbf{w}}_k^{\mathbf{T}} \tilde{\mathbf{x}} \\ \vdots \\ \tilde{\mathbf{w}}_K^{\mathbf{T}} \tilde{\mathbf{x}} \end{pmatrix} = \begin{pmatrix} y_1(\mathbf{x}) \\ \vdots \\ y_k(\mathbf{x}) \\ \vdots \\ y_K(\mathbf{x}) \end{pmatrix}$$

How do we learn the decision boundaries $(w_0, \mathbf{w})$?

- One approach is to use least squares, similar to regression

$$
\begin{aligned}
E_D(\widetilde{\mathbf{W}}) &= \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}\{y_k(\mathbf{x_n}) - t_{nk}\}^2 \\
&= \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\} \qquad (4.15)
\end{aligned}
$$

- $\widetilde{\mathbf{X}}$ is a matrix whose n^{th} row is $\tilde{\mathbf{x}}_n^{\mathbf{T}}$
- $\mathbf{T}$ is a matrix whose n^{th} row is the vector $\mathbf{t}_n^T$

Let's get used to a matrix representation

$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\tilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\tilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\} \tag{4.15}$$

$$= \frac{1}{2}\mathrm{Tr}\left[\left\{\begin{pmatrix} x_{10} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N0} & \cdots & x_{ND}\end{pmatrix}\begin{pmatrix} w_{10} & \cdots & w_{K0} \\ \vdots & \ddots & \vdots \\ w_{1D} & \cdots & w_{KD}\end{pmatrix} - \begin{pmatrix} t_{11} & \cdots & t_{1K} \\ \vdots & \ddots & \vdots \\ t_{N1} & \cdots & t_{NK}\end{pmatrix}\right\}^{\mathbf{T}}\right.$$

$$\left.\left\{\begin{pmatrix} x_{10} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N0} & \cdots & x_{ND}\end{pmatrix}\begin{pmatrix} w_{10} & \cdots & w_{K0} \\ \vdots & \ddots & \vdots \\ w_{1D} & \cdots & w_{KD}\end{pmatrix} - \begin{pmatrix} t_{11} & \cdots & t_{1K} \\ \vdots & \ddots & \vdots \\ t_{N1} & \cdots & t_{NK}\end{pmatrix}\right\}\right]$$

$$= \frac{1}{2}\mathrm{Tr}\left\{\begin{pmatrix} \tilde{\mathbf{w}}_1^{\mathbf{T}}\tilde{\mathbf{x}}_1 - t_{11} & \cdots & \tilde{\mathbf{w}}_K^{\mathbf{T}}\tilde{\mathbf{x}}_1 - t_{1K} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{w}}_1^{\mathbf{T}}\tilde{\mathbf{x}}_N - t_{N1} & \cdots & \tilde{\mathbf{w}}_K^{\mathbf{T}}\tilde{\mathbf{x}}_N - t_{NN}\end{pmatrix}^{\mathbf{T}}\begin{pmatrix} \tilde{\mathbf{w}}_1^{\mathbf{T}}\tilde{\mathbf{x}}_1 - t_{11} & \cdots & \tilde{\mathbf{w}}_K^{\mathbf{T}}\tilde{\mathbf{x}}_1 - t_{1K} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{w}}_1^{\mathbf{T}}\tilde{\mathbf{x}}_N - t_{N1} & \cdots & \tilde{\mathbf{w}}_K^{\mathbf{T}}\tilde{\mathbf{x}}_N - t_{NN}\end{pmatrix}\right\}$$

$$= \frac{1}{2}\mathrm{Tr}\begin{pmatrix} (\tilde{\mathbf{w}}_1^{\mathbf{T}}\tilde{\mathbf{x}}_1 - t_{11})^2 + \cdots & ,\cdots, & (\tilde{\mathbf{w}}_1^{\mathbf{T}}\tilde{\mathbf{x}}_1 - t_{11})(\tilde{\mathbf{w}}_K^{\mathbf{T}}\tilde{\mathbf{x}}_1 - t_{1K}) + \cdots \\ \vdots & \ddots & \vdots \\ (\tilde{\mathbf{w}}_K^{\mathbf{T}}\tilde{\mathbf{x}}_1 - t_{1K})(\tilde{\mathbf{w}}_1^{\mathbf{T}}\tilde{\mathbf{x}}_1 - t_{11}) + \cdots & ,\cdots, & (\tilde{\mathbf{w}}_K^{\mathbf{T}}\tilde{\mathbf{x}}_1 - t_{1K})^2 + \cdots \end{pmatrix}$$

$$= \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}(y_k(\mathbf{x_n}) - t_{nk})^2$$

Solution for $\widetilde{\mathbf{W}}$

- Setting the derivative with respect to $\widetilde{\mathbf{W}}$ to zero, and rearranging

$$\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^{\mathrm{T}}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^{\mathrm{T}}\mathbf{T} = \widetilde{\mathbf{X}}^{\dagger}\mathbf{T} \tag{4.16}$$

- $\widetilde{\mathbf{X}}^{\dagger}$ is the pseudo-inverse of the matrix $\widetilde{\mathbf{X}}$

We get the discriminant functions from (4.16)

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\tilde{\mathbf{x}} = \mathbf{T}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}^{\dagger}\right)^{\mathrm{T}}\tilde{\mathbf{x}} \tag{4.17}$$

Solution for $\widetilde{\mathbf{W}}$: Calculation process

- $\mathbf{t}_k$ is a column component of the matrix $\mathbf{T}$
  - re-writing the error function and differentiating with respect to $\mathbf{w}_i$

$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}\{y_k(\mathbf{x_n}) - t_{nk}\}^2 = \frac{1}{2}\sum_{k=1}^{K}||\widetilde{\mathbf{X}}\tilde{\mathbf{w}}_k - \mathbf{t}_k||^2 \qquad (4.15^*)$$

$$\frac{\partial E_D}{\partial \mathbf{w}_i} = \widetilde{\mathbf{X}}^{\mathbf{T}}\left(\widetilde{\mathbf{X}}\tilde{\mathbf{w}}_i - \mathbf{t}_i\right) = 0 \qquad (\text{C1})$$

$$\Downarrow$$

$$\tilde{\mathbf{w}}_i = (\widetilde{\mathbf{X}}^{\mathrm{T}}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^{\mathrm{T}}\mathbf{t}_i = \widetilde{\mathbf{X}}^{\dagger}\mathbf{t}_i \qquad (\text{C2})$$

$$\Downarrow$$

$$\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^{\mathrm{T}}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^{\mathrm{T}}\mathbf{T} = \widetilde{\mathbf{X}}^{\dagger}\mathbf{T} \qquad (4.16)$$

An interesting property of least-squares solutions with multiple target variables

- If every target vector in the training set satisfies some linear constraint for some constants $\mathbf{a}$ and b,

$$\mathbf{a}^{\mathrm{T}}\mathbf{t}_n + b = 0 \tag{4.18}$$

- Then the model prediction for any value of $\mathbf{x}$ will satisfy the same constraint

$$\mathbf{a}^{\mathrm{T}}\mathbf{y}(\mathbf{x}) + b = 0 \tag{4.19}$$

- Thus, if we use 1-of-K coding scheme for K classes, the elements of $\mathbf{y}(\mathbf{x})$ will sum to 1 for any value of $\mathbf{x}$
  - However, this summation constraint alone can not be interpret as probabilities because they aren't considered to lie within the interval (0, 1)

Exercise 4.2

Consider the minimization of a sum-of-squares error function (4.15), and suppose that all of the target vectors in the training set satisfy a linear constraint

$$\mathbf{a}^{\mathrm{T}}\mathbf{t}_n + b = 0 \tag{4.18}$$

where $\mathbf{t}_n$ corresponds to the n^{th} row of the matrix $\mathbf{T}$ in (4.15). Show that as a consequence of this constraint, the elements of the model prediction $\mathbf{y}(\mathbf{x})$ given by the least-squares solution (4.17) also satisfy this constraint, so that

$$\mathbf{a}^{\mathrm{T}}\mathbf{y}(\mathbf{x}) + b = 0 \tag{4.19}$$

To do so, assume that one of the basis functions $\phi_0(\mathbf{x}) = 1$ so that the corresponding parameter $\mathbf{w}_0$ plays the role of a bias.

Decomposition of $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{X}}$

- $\widetilde{\mathbf{W}} = \begin{pmatrix} \mathbf{w}_0^{\mathrm{T}} \\ \mathbf{W} \end{pmatrix}$: $\mathbf{w}_0$ is the column vector of the bias weight
- $\widetilde{\mathbf{X}} = \begin{pmatrix} \mathbf{1} & \mathbf{X} \end{pmatrix}$: $\mathbf{1}$ is a column vector of N length

By using the definition above, the error function (4.15) is re-written,

$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\} \tag{4.15}$$

$$= \frac{1}{2}\mathrm{Tr}\left\{(\mathbf{X}\mathbf{W} + \mathbf{1}\mathbf{w}_0^{\mathrm{T}} - \mathbf{T})^{\mathrm{T}}(\mathbf{X}\mathbf{W} + \mathbf{1}\mathbf{w}_0^{\mathrm{T}} - \mathbf{T})\right\} \tag{C3}$$

Solution for $\mathbf{w}_0$

Setting the derivative of (C3) with respect to $\mathbf{w}_0$ to zero,

$$
\begin{aligned}
\frac{\partial E_D}{\partial \mathbf{w}_0} &= \frac{1}{2}\frac{\partial}{\partial \mathbf{w}_0}\mathrm{Tr}\left\{\left\{(\mathbf{XW}+\mathbf{1}\mathbf{w}_0^{\mathrm{T}}-\mathbf{T})^{\mathrm{T}}(\mathbf{XW}+\mathbf{1}\mathbf{w}_0^{\mathrm{T}}-\mathbf{T})\right\}\right\} \\
&= \frac{1}{2}\frac{\partial}{\partial \mathbf{w}_0}\{\mathrm{Tr}\{(\mathbf{XW})^{\mathrm{T}}\mathbf{XW}+(\mathbf{XW})^{\mathrm{T}}\mathbf{1}\mathbf{w}_0^{\mathrm{T}}-(\mathbf{XW})^{\mathrm{T}}\mathbf{T} \\
&+(\mathbf{1}\mathbf{w}_0^{\mathrm{T}})^{\mathrm{T}}\mathbf{XW}+(\mathbf{1}\mathbf{w}_0^{\mathrm{T}})^{\mathrm{T}}\mathbf{1}\mathbf{w}_0^{\mathrm{T}}-(\mathbf{1}\mathbf{w}_0^{\mathrm{T}})^{\mathrm{T}}\mathbf{T}-\mathbf{T}^{\mathrm{T}}\mathbf{XW}-\mathbf{T}^{\mathrm{T}}\mathbf{1}\mathbf{w}_0^{\mathrm{T}}+\mathbf{T}^{\mathrm{T}}\mathbf{T}\}\} \\
&= \frac{1}{2}\frac{\partial}{\partial \mathbf{w}_0}\mathrm{Tr}\{\mathbf{w}_0\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{w}_0^{\mathrm{T}}+2\mathbf{w}_0(\mathbf{1}^{\mathrm{T}}\mathbf{XW}-\mathbf{1}^{\mathrm{T}}\mathbf{T})+(\mathbf{XW}-\mathbf{T})^{\mathrm{T}}(\mathbf{XW}-\mathbf{T})\} \\
&= \frac{1}{2}\frac{\partial}{\partial \mathbf{w}_0}\mathrm{Tr}\{N\mathbf{w}_0\mathbf{w}_0^{\mathrm{T}}+2\mathbf{w}_0\mathbf{1}^{\mathrm{T}}(\mathbf{XW}-\mathbf{T})+\mathrm{const}\} \\
&= \frac{N}{2}\frac{\partial}{\partial \mathbf{w}_0}\mathbf{w}_0^{\mathrm{T}}\mathbf{w}_0+\frac{2}{2}\frac{\partial}{\partial \mathbf{w}_0}\mathbf{w}_0\{\mathbf{1}^{\mathrm{T}}(\mathbf{XW}-\mathbf{T})\}^{\mathrm{T}}+0 \\
&= N\mathbf{w}_0+(\mathbf{XW}-\mathbf{T})^{\mathrm{T}}\mathbf{1}=0 \qquad \text{(C4)}
\end{aligned}
$$

$$N\mathbf{w}_0 + (\mathbf{XW} - \mathbf{T})^{\mathrm{T}}\mathbf{1} = 0 \tag{C4}$$

$$\Downarrow$$

$$\mathbf{w}_0 = \frac{1}{N}\mathbf{T}^{\mathrm{T}}\mathbf{1} - \frac{1}{N}\mathbf{W}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{1} = \bar{\mathbf{t}} - \mathbf{W}^{\mathrm{T}}\bar{\mathbf{x}} \tag{C5}$$

- $\bar{\mathbf{t}} = \frac{1}{N}\mathbf{T}^{\mathrm{T}}\mathbf{1}$ is a mean vector of all target vectors
- $\bar{\mathbf{x}}$ is a mean vector of all feature vectors

Solution for $\mathbf{W}$

Substituting (C5) into (C3),

$$
\begin{aligned}
E_D(\widetilde{\mathbf{W}}) &= \frac{1}{2}\mathrm{Tr}\left\{(\mathbf{X}\mathbf{W} + \mathbf{1}\mathbf{w}_0^{\mathrm{T}} - \mathbf{T})^{\mathrm{T}}(\mathbf{X}\mathbf{W} + \mathbf{1}\mathbf{w}_0^{\mathrm{T}} - \mathbf{T})\right\} \\
&= \frac{1}{2}\mathrm{Tr}\left\{(\mathbf{X}\mathbf{W} + \mathbf{1}(\bar{\mathbf{t}} - \mathbf{W}^{\mathrm{T}}\overline{\mathbf{x}})^{\mathrm{T}} - \mathbf{T})^{\mathrm{T}}(\mathbf{X}\mathbf{W} + \mathbf{1}(\bar{\mathbf{t}} - \mathbf{W}^{\mathrm{T}}\overline{\mathbf{x}})^{\mathrm{T}} - \mathbf{T})\right\} \\
&= \frac{1}{2}\mathrm{Tr}\left\{(\mathbf{X}\mathbf{W} + \mathbf{1}\bar{\mathbf{t}}^{\mathrm{T}} - \mathbf{1}\overline{\mathbf{x}}^{\mathrm{T}}\mathbf{W} - \mathbf{T})^{\mathrm{T}}(\mathbf{X}\mathbf{W} + \mathbf{1}\bar{\mathbf{t}}^{\mathrm{T}} - \mathbf{1}\overline{\mathbf{x}}^{\mathrm{T}}\mathbf{W} - \mathbf{T})\right\} \\
&= \frac{1}{2}\mathrm{Tr}\left\{(\mathbf{X}\mathbf{W} + \overline{\mathbf{T}} - \overline{\mathbf{X}}\mathbf{W} - \mathbf{T})^{\mathrm{T}}(\mathbf{X}\mathbf{W} + \overline{\mathbf{T}} - \overline{\mathbf{X}}\mathbf{W} - \mathbf{T})\right\} \\
&= \frac{1}{2}\mathrm{Tr}\left\{\left\{(\mathbf{X} - \overline{\mathbf{X}})\mathbf{W} - (\mathbf{T} - \overline{\mathbf{T}})\right\}^{\mathrm{T}}\left\{(\mathbf{X} - \overline{\mathbf{X}})\mathbf{W} - (\mathbf{T} - \overline{\mathbf{T}})\right\}\right\} \qquad \text{(C6)}
\end{aligned}
$$

- $\overline{\mathbf{T}} = \mathbf{1}\bar{\mathbf{t}}^{\mathrm{T}}$ is the matrix whose row is $\bar{\mathbf{t}}^{\mathrm{T}}$
- $\overline{\mathbf{X}} = \mathbf{1}\overline{\mathbf{x}}^{\mathrm{T}}$ is the matrix whose row is $\overline{\mathbf{x}}^{\mathrm{T}}$

Setting the derivative of this (C6) with respect to $\mathbf{W}$ to zero,

$$\begin{aligned}\frac{\partial E_D}{\partial \mathbf{W}} &= \frac{\partial}{\partial \mathbf{W}}\left\{\frac{1}{2}\mathrm{Tr}\left\{\{(\mathbf{X}-\overline{\mathbf{X}})\mathbf{W}-(\mathbf{T}-\overline{\mathbf{T}})\}^{\mathrm{T}}\{(\mathbf{X}-\overline{\mathbf{X}})\mathbf{W}-(\mathbf{T}-\overline{\mathbf{T}})\}\right\}\right\} \\ &= (\mathbf{X}-\overline{\mathbf{X}})^{\mathrm{T}}\{(\mathbf{X}-\overline{\mathbf{X}})\mathbf{W}-(\mathbf{T}-\overline{\mathbf{T}})\} = 0 \end{aligned} \tag{C7}$$

- $\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}\left\{(\mathbf{AXB}+\mathbf{C})(\mathbf{AXB}+\mathbf{C})^{\mathrm{T}}\right\} = 2\mathbf{A}^{\mathrm{T}}(\mathbf{AXB}+\mathbf{C})\mathbf{B}^{\mathrm{T}}$

$$\begin{aligned}\mathbf{W} &= \{(\mathbf{X}-\overline{\mathbf{X}})^{\mathrm{T}}(\mathbf{X}-\overline{\mathbf{X}})\}^{-1}(\mathbf{X}-\overline{\mathbf{X}})(\mathbf{T}-\overline{\mathbf{T}}) = \{\widehat{\mathbf{X}}^{\mathrm{T}}\widehat{\mathbf{X}}\}^{-1}\widehat{\mathbf{X}}\widehat{\mathbf{T}} \\ &= \widehat{\mathbf{X}}^{\dagger}\widehat{\mathbf{T}} \end{aligned} \tag{C8}$$

- $\widehat{\mathbf{X}} = \mathbf{X}-\overline{\mathbf{X}}$
- $\widehat{\mathbf{T}} = \mathbf{T}-\overline{\mathbf{T}}$
- $\widehat{\mathbf{X}}^{\dagger} = \{\widehat{\mathbf{X}}^{\mathrm{T}}\widehat{\mathbf{X}}\}^{-1}\widehat{\mathbf{X}}$: $\widehat{\mathbf{X}}^{\dagger}$ is the is the pseudo-inverse of the matrix $\widehat{\mathbf{X}}$

Verification using new input

Now, we suppose that $\mathbf{x}^*$ is a new input,

$$\begin{aligned}\mathbf{y}(\mathbf{x}^*) &= \mathbf{W}^{\mathrm{T}}\mathbf{x}^* + \mathbf{w}_0 = \mathbf{W}^{\mathrm{T}}\mathbf{x}^* + \overline{\mathbf{t}} - \mathbf{W}^{\mathrm{T}}\overline{\mathbf{x}} = \mathbf{W}^{\mathrm{T}}(\mathbf{x}^* - \overline{\mathbf{x}}) + \overline{\mathbf{t}} \\ &= (\widehat{\mathbf{X}}^{\dagger}\widehat{\mathbf{T}})^{\mathrm{T}}(\mathbf{x}^* - \overline{\mathbf{x}}) + \overline{\mathbf{t}} \\ &= \overline{\mathbf{t}} + \widehat{\mathbf{T}}^{\mathrm{T}}(\widehat{\mathbf{X}}^{\dagger})^{\mathrm{T}}(\mathbf{x}^* - \overline{\mathbf{x}}) \end{aligned} \tag{C9}$$

If we apply (4.18) to $\overline{\mathbf{t}}$, we get

$$\mathbf{a}^{\mathrm{T}}\overline{\mathbf{t}} = \frac{1}{N}\mathbf{a}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\mathbf{1} = -b \tag{C10}$$

Therefore, applying (C9) to (4.19), we obtain

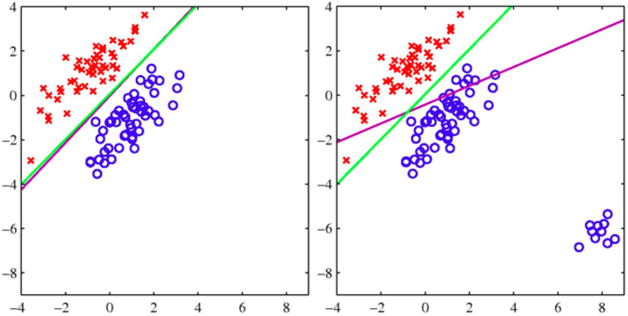
$$\begin{aligned}\mathbf{a}^{\mathrm{T}}\mathbf{y}(\mathbf{x}^*) &= \mathbf{a}^{\mathrm{T}}\{\overline{\mathbf{t}} + \widehat{\mathbf{T}}^{\mathrm{T}}(\widehat{\mathbf{X}}^{\dagger})^{\mathrm{T}}(\mathbf{x}^* - \overline{\mathbf{x}})\} \\ &= \mathbf{a}^{\mathrm{T}}\overline{\mathbf{t}} + \mathbf{a}^{\mathrm{T}}\widehat{\mathbf{T}}^{\mathrm{T}}(\widehat{\mathbf{X}}^{\dagger})^{\mathrm{T}}(\mathbf{x}^* - \overline{\mathbf{x}}) \\ &= \mathbf{a}^{\mathrm{T}}\overline{\mathbf{t}} = -b \end{aligned} \tag{C11}$$

Supplement for (C11)

$$
\begin{aligned}
\mathbf{a}^{\mathrm{T}}\widehat{\mathbf{T}}^{\mathrm{T}} = \mathbf{a}^{\mathrm{T}}(\mathbf{T} - \overline{\mathbf{T}})^{\mathrm{T}} &= \mathbf{a}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}} - \mathbf{a}^{\mathrm{T}}\overline{\mathbf{T}}^{\mathrm{T}} \\
&= \mathbf{a}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}} - \mathbf{a}^{\mathrm{T}}(\mathbf{1}\bar{\mathbf{t}}^{\mathrm{T}})^{\mathrm{T}} \\
&= \mathbf{a}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}} - \mathbf{a}^{\mathrm{T}}\bar{\mathbf{t}}\mathbf{1}^{\mathrm{T}} \\
&= \mathbf{a}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}} - \frac{1}{N}\mathbf{a}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\mathbf{1}\mathbf{1}^{\mathrm{T}} \\
&= \mathbf{a}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}} + b\mathbf{1}^{\mathrm{T}} \\
&= -b\mathbf{1}^{\mathrm{T}} + b\mathbf{1}^{\mathrm{T}} \\
&= b(\mathbf{1}^{\mathrm{T}} - \mathbf{1}^{\mathrm{T}}) \\
&= \mathbf{0}^{\mathrm{T}} \qquad \text{(C12)}
\end{aligned}
$$

Least Squares is highly sensitive to outliers

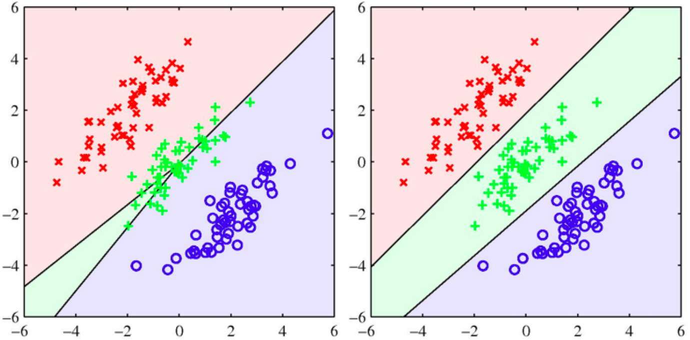
- Two-class classification between red and blue points:
  - green line: decision boundary from logistic regression
  - magenda line: decision boundary from least square



- Output range:
  - logistic regression: 0 -1
  - discriminant function in this chapter with least square: ?

Problems with Least Squares against assumption of a Gaussian distribution

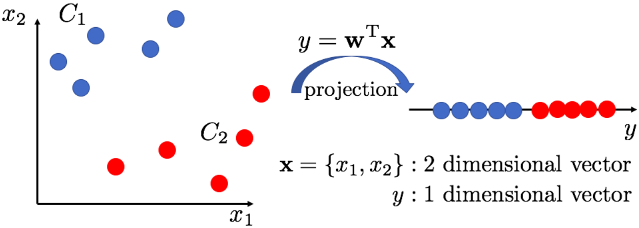
- Multiple-class classification among red, green, and blue points
  - left figure: decision boundary from least squares
  - right figure: decision boundary from logistic regression



Apparently, distributions of samples aren't Gaussian distribution

Outline

The two-class linear discriminant acts as a projection



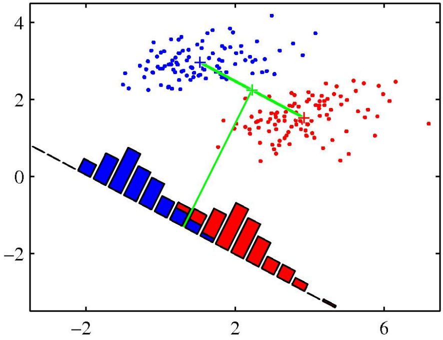
$$y = \mathbf{w}^{\mathrm{T}}\mathbf{x} \tag{4.20}$$

If we place a threshold on y, this is a linear classifier

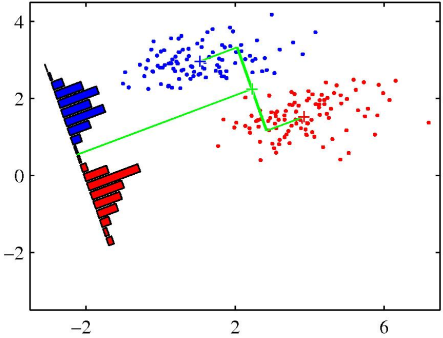
- classify $y \geq -w_0$ as C_1
- otherwise C_2

What $\mathbf{w}$ is the best?

- A natural idea would be to project in the direction of the line connecting class means
- However, problematic if classes have variance in this direction

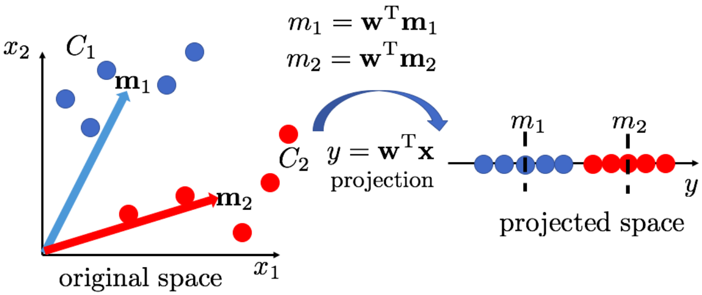


Minimize the class overlap in the projected space



- A large separation between the projected class means
- A small variance within each class

Mean (vector) definition in each space



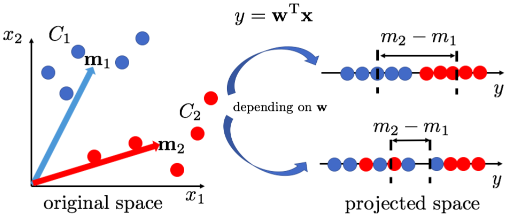
- Mean vector of each class in a original space

$$\mathbf{m}_k = \frac{1}{N_k}\sum_{n \in C_k} \mathbf{x}_n \tag{4.21*}$$

- Mean of each class in a projected space

$$m_k = \mathbf{w}^{\mathrm{T}}\mathbf{m}_k \tag{4.23}$$

The ratio of mean separation in projected space



- The ratio of separation in projected space is distance between projected class means
  - Distance $m_2 - m_1$ depends on the $\mathbf{w}$

$$m_2 - m_1 = \mathbf{w}^{\mathrm{T}}(\mathbf{m}_2 - \mathbf{m}_1) \tag{4.22}$$

- Now, distance $m_2 - m_1$ can have a large value without limitation by increasing of value $\mathbf{w}$

$\Rightarrow$ Constraint of $\mathbf{w}$: unit length $||\mathbf{w}|| = 1$

Exercise 4.4: A Lagrange multiplier to perform the constrained maximization

- Maximize $m_2 - m_1 = \mathbf{w^T(m_2 - m_1)}$
- Subject to $||\mathbf{w}||^2 - 1 = 0$ ($||\mathbf{w}||^2 = 1$)
  - λ is Lagrange multiplier

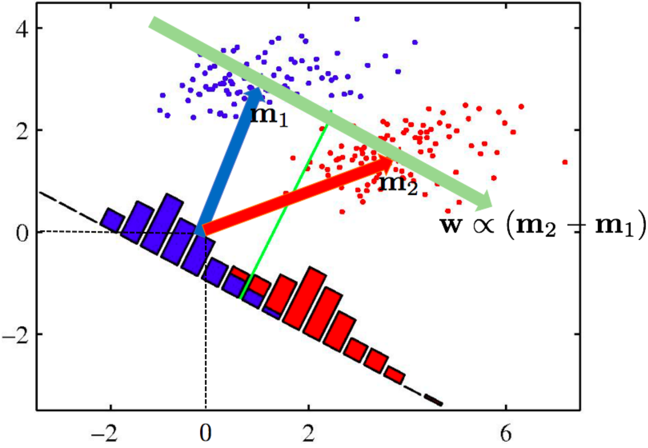
$$L(\mathbf{w}, \lambda) = \mathbf{w^T(m_2 - m_1)} - \lambda(||\mathbf{w}||^2 - 1) \tag{D1}$$

- Setting the derivative of (D1) with respect to $\mathbf{w}$ to zero

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{m_2 - m_1} - 2\lambda\mathbf{w} = 0 \tag{D2}$$

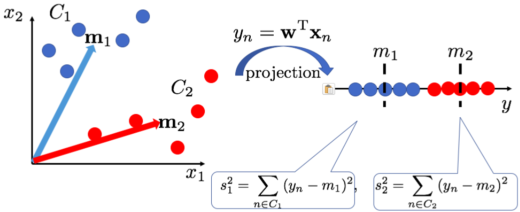
$$\mathbf{w} = \frac{1}{2\lambda}(\mathbf{m}_2 - \mathbf{m}_1) \propto (\mathbf{m}_2 - \mathbf{m}_1) \tag{D3}$$

There is considerable class overlap in the projected space



$\Rightarrow$We should take variance into consideration

The within-class variance



- The within-class variance of the transformed (projected) data from class C_k

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2 \tag{4.24}$$

$$y_n = \mathbf{w}^{\mathbf{T}}\mathbf{x}_n \tag{4.20*}$$

- This variance (4.24) doesn't have $1/N$, taking the number of samples into consideration

Fisher criterion

$$J(\mathbf{w}) = \frac{\text{maximization : between-class variance}}{\text{minimization : within-class variance}}$$

maximization : Fisher criterion

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \tag{4.25}$$

- Ratio of the between-class variance $(m_2 - m_1)^2$ to the within-class variance $(s_1^2 + s_2^2)$

Deformation for maximization of $J(\mathbf{w})$

- Rewriting Fisher criterion:

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}} \tag{4.26}$$

- Between-class covariance matrix:

$$\mathbf{S}_{\mathrm{B}} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}} \tag{4.27}$$

- Total within-class covariance matrix:

$$\mathbf{S}_{\mathrm{W}} = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^{\mathrm{T}} + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^{\mathrm{T}} \tag{4.28}$$

Deformation for maximization of $J(\mathbf{w})$: Supplement

$$\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w} = \mathbf{w}^{\mathrm{T}}(\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}}\mathbf{w} = (m_2 - m_1)^2 \tag{D4}$$

$$\begin{aligned}
\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w} &= \mathbf{w}^{\mathrm{T}}\sum_{n\in C_1}(\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^{\mathrm{T}}\mathbf{w} \\
&\quad + \mathbf{w}^{\mathrm{T}}\sum_{n\in C_2}(\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^{\mathrm{T}}\mathbf{w} \\
&= \sum_{n\in C_1}(y_n - m_1)^2 + \sum_{n\in C_2}(y_n - m_2)^2 \\
&= s_1^2 + s_2^2
\end{aligned} \tag{D5}$$

Maximization of $J(\mathbf{w})$

- Maximization of $J(\mathbf{w})$: Setting the derivative with respect $\mathbf{w}$ to 0

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}}\left(\frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}\right) = \frac{2(\mathbf{S_B}\mathbf{w})(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}) - 2(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w})(\mathbf{S}_{\mathrm{W}}\mathbf{w})}{(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w})^2} = 0 \tag{D6}$$

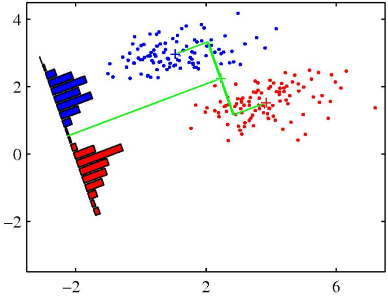
$$(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w})\mathbf{S}_{\mathrm{W}}\mathbf{w} = (\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w})\mathbf{S}_{\mathrm{B}}\mathbf{w} \tag{4.29}$$

In term of weight vector $\mathbf{w}$,

- Ignore scalar factors $\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}$ and $\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}$ because direction of weight vector $\mathbf{w}$ only is important
- Use $\mathbf{S}_{\mathrm{B}}\mathbf{w} = (\mathbf{m}_2 - \mathbf{m}_1)((\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}}\mathbf{w}) \propto (\mathbf{m}_2 - \mathbf{m}_1)$

$$\Downarrow$$

Fisher's linear discriminant



$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}\mathbf{S}_{\mathrm{B}}\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \tag{4.30}$$

- (4.30) is a selector of direction down to a one dimension, although (4,30) is called "discriminant"
- However: we can use (4.30) to construct a classifier setting a threshold

Outline

Fisher criterion can be obtained as a special case of least squares especially in Two classes

- Least squares
  - Making the model predictions as close as possible to a set of target values
- Fisher criterion
  - Making maximum class separation in the projected space
- Target values in this proof (Actually not limited to this definition, Mika, "Kernel Fisher Discriminants")
  - $C_1 : \frac{N}{N_1}$
  - $C_2 : -\frac{N}{N_2}$

Minimization of sum-of-squares error

- The sum-of-squares error function

$$E = \frac{1}{2} \sum_{n=1}^{N} (\mathbf{w}^{\mathrm{T}} \mathbf{x}_n + w_0 - t_n)^2 \tag{4.31}$$

- Setting the derivatives of E with respect to $\mathbf{w}$ and w_0 to zero

$$\frac{\partial E}{\partial \mathbf{w}} = \sum_{n=1}^{N} (\mathbf{w}^{\mathrm{T}} \mathbf{x}_n + w_0 - t_n) = 0 \tag{4.32}$$

$$\frac{\partial E}{\partial w_0} = \sum_{n=1}^{N} (\mathbf{w}^{\mathrm{T}} \mathbf{x}_n + w_0 - t_n) \mathbf{x}_n = 0 \tag{4.33}$$

- Bias w_0 expression from (4.32):

$$\frac{\partial E}{\partial \mathbf{w}} = \sum_{n=1}^{N} (\mathbf{w}^{\mathrm{T}} \mathbf{x}_n + w_0 - t_n) = \sum_{n=1}^{N} \mathbf{w}^{\mathrm{T}} \mathbf{x}_n + \sum_{n=1}^{N} w_0 - \sum_{n=1}^{N} t_n$$

$$= N \mathbf{w}^{\mathrm{T}} \mathbf{m} + N w_0 - \sum_{n=1}^{N} t_n = 0 \tag{E1}$$

$$w_0 = -\frac{N}{N} \mathbf{w}^{\mathrm{T}} \mathbf{m} + \frac{1}{N} \sum_{n=1}^{N} t_n = -\mathbf{w}^{\mathrm{T}} \mathbf{m} \tag{4.34}$$

Using following equations

$$\sum_{n=1}^{N} t_n = N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2} = 0 \tag{4.35}$$

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n = \frac{1}{N} (N_1 \mathbf{m}_1 + N_2 \mathbf{m}_2) \tag{4.36}$$

- $\frac{\partial E}{\partial w_0} = 0$ (4.33) is represented by using target values that is used in this chapter

$$\left(\mathbf{S}_{\mathrm{W}} + \frac{N_1 N_2}{N}\mathbf{S}_{\mathrm{B}}\right)\mathbf{w} = N(\mathbf{m}_1 - \mathbf{m}_2) \tag{4.37}$$

$$(4.37) \Rightarrow \mathbf{S}_{\mathrm{W}}\mathbf{w} + \frac{N_1 N_2}{N}\mathbf{S_B}\mathbf{w} = N(\mathbf{m}_1 - \mathbf{m}_2)$$

In term of $\mathbf{w}$,

- Use $\mathbf{S}_{\mathrm{B}}\mathbf{w} = (\mathbf{m}_2 - \mathbf{m}_1)((\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}}\mathbf{w}) \propto (\mathbf{m}_2 - \mathbf{m}_1)$
- Ignore scalar factors

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \tag{4.38}$$

$$\Downarrow$$

- The discriminant function: $y(\mathbf{x}) \geq 0 \rightarrow C_1$, otherwise C_2

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = \mathbf{w}^{\mathrm{T}}\mathbf{x} - \mathbf{w}^{\mathbf{T}}\mathbf{m} = \mathbf{w}^{\mathbf{T}}(\mathbf{x} - \mathbf{m}) \tag{E2}$$

Verification from (4.33) to (4.37)

$$\frac{\partial E}{\partial w_0} = \sum_{n=1}^{N}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 - t_n)\mathbf{x}_n = \sum_{n=1}^{N}\mathbf{w}^{\mathrm{T}}\mathbf{x}_n\mathbf{x}_n + \sum_{n=1}^{N} w_0\mathbf{x}_n - \sum_{n=1}^{N} t_n\mathbf{x}_n$$

$$= \sum_{n=1}^{N}(\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}})\mathbf{w} - N(\mathbf{m}\mathbf{m}^{\mathbf{T}})\mathbf{w} - N(\mathbf{m_1} - \mathbf{m}_2) = 0 \tag{E3}$$

$$\left(\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}} - N\mathbf{m}\mathbf{m}^{\mathbf{T}}\right)\mathbf{w} = N(\mathbf{m_1} - \mathbf{m}_2) \tag{E4}$$

To calculate (E4), using following equations,

$$\sum_{n=1}^{N} w_0\mathbf{x}_n = N w_0\mathbf{m} = -N(-\mathbf{w}^{\mathbf{T}}\mathbf{m})\mathbf{m} = -N(\mathbf{m}\mathbf{m}^{\mathbf{T}})\mathbf{w} \tag{E5}$$

$$\sum_{n=1}^{N} t_n\mathbf{x}_n = \sum_{n\in C_1} t_n\mathbf{x}_n + \sum_{n\in C_2} t_n\mathbf{x}_n = N_1\frac{N}{N_1}\mathbf{m}_1 - N_2\frac{N}{N_2}\mathbf{m}_2$$

$$= N(\mathbf{m}_1 - \mathbf{m}_2) \tag{E6}$$

Verification from (4.33) to (4.37)

- We need to prove: comparing (4.37) with (E4)

$$\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} - N\mathbf{m}\mathbf{m}^{\mathbf{T}} = \mathbf{S}_{\mathrm{W}} + \frac{N_1 N_2}{N}\mathbf{S}_{\mathrm{B}} \tag{E7}$$

- Expanding total within-class covariance matrix

$$\mathbf{S}_{\mathrm{W}} = \sum_{n\in C_1}(\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^{\mathrm{T}} + \sum_{n\in C_2}(\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^{\mathrm{T}} \tag{4.28}$$

$$= \sum_{n\in C_1} \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} - 2\sum_{n\in C_1} \mathbf{x}_n \mathbf{m}_1^{\mathrm{T}} + \sum_{n\in C_1} \mathbf{m}_1 \mathbf{m}_1^{\mathrm{T}}$$

$$+ \sum_{n\in C_2} \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} - 2\sum_{n\in C_2} \mathbf{x}_n \mathbf{m}_2^{\mathrm{T}} + \sum_{n\in C_2} \mathbf{m}_2 \mathbf{m}_2^{\mathrm{T}}$$

$$= \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} - N_1 \mathbf{m}_1 \mathbf{m}_1^{\mathrm{T}} - N_2 \mathbf{m}_2 \mathbf{m}_2^{\mathrm{T}} \tag{E8}$$

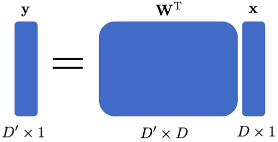
- Expanding the left side of the equation (E7)

$$
\begin{aligned}
\text{left} &= \mathbf{S_W} + N_1\mathbf{m}_1\mathbf{m}_1^{\mathrm{T}} + N_2\mathbf{m}_2\mathbf{m}_2^{\mathrm{T}} - N\mathbf{m}\mathbf{m}^{\mathbf{T}} \\
&= \mathbf{S_W} + N_1\mathbf{m}_1\mathbf{m}_1^{\mathrm{T}} + N_2\mathbf{m}_2\mathbf{m}_2^{\mathrm{T}} \\
&- N(\frac{N_1^2}{N^2}\mathbf{m}_1\mathbf{m}_1^{\mathrm{T}} + 2\frac{N_1N_2}{N^2}\mathbf{m}_1\mathbf{m}_2^{\mathrm{T}} + \frac{N_2^2}{N^2}\mathbf{m}_2\mathbf{m}_2^{\mathrm{T}}) \\
&= \mathbf{S_W} \\
&+ \frac{N_1N_2}{N}\left(\frac{N}{N_2}\mathbf{m}_1\mathbf{m}_1^{\mathrm{T}} + \frac{N}{N_1}\mathbf{m}_2\mathbf{m}_2^{\mathrm{T}} - \frac{N_1}{N_2}\mathbf{m}_1\mathbf{m}_1^{\mathrm{T}} - \mathbf{2}\mathbf{m}_1\mathbf{m}_2^{\mathrm{T}} - \frac{N_2}{N_1}\mathbf{m}_2\mathbf{m}_2^{\mathrm{T}}\right) \\
&= \mathbf{S_W} + \frac{N_1N_2}{N}\left(\frac{N-N_1}{N_2}\mathbf{m}_1\mathbf{m}_1^{\mathrm{T}} - \mathbf{2}\mathbf{m}_1\mathbf{m}_2^{\mathrm{T}} + \frac{N-N_2}{N_1}\mathbf{m}_2\mathbf{m}_2^{\mathrm{T}}\right) \\
&= \mathbf{S_W} + \frac{N_1N_2}{N}\left(\mathbf{m}_1\mathbf{m}_1^{\mathrm{T}} - \mathbf{2}\mathbf{m}_1\mathbf{m}_2^{\mathrm{T}} + \mathbf{m}_2\mathbf{m}_2^{\mathrm{T}}\right) \\
&= \mathbf{S_W} + \frac{N_1N_2}{N}(\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}} \\
&= \mathbf{S}_{\mathrm{W}} + \frac{N_1N_2}{N}\mathbf{S}_{\mathrm{B}}
\end{aligned}
$$

Outline

Fisher's discriminant for multiple classes

- The number of classes: $K > 2$
- Input space: $D > K$
- Linear features: $y_k = \mathbf{w}_k^T \mathbf{x}$, $D' > 1, k = 1, \ldots, D'$
- Projection matrix $\mathbf{W}$: $\{\mathbf{w}_k\} = [\mathbf{w}_1 \cdots \mathbf{w}_k \cdots \mathbf{w}_{D'}]$ are the columns



- Relationship between vector $\mathbf{y}$ and matrix $\mathbf{W}$:

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x} \tag{4.39}$$

- (4.39) transforms an original feature vector $\mathbf{x}$ into a new feature vector $\mathbf{y}$, using projection matrix $\mathbf{W}$

This (4.39) doesn't include bias.

Definition of covariance matrix in original space

- The generalization of the total within-class covariance matrix:

$$\mathbf{S}_{\mathrm{W}} = \sum_{k=1}^{K} \mathbf{S}_k \tag{4.40}$$

- the within-class covariance matrix:

$$\mathbf{S}_k = \sum_{n \in C_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^{\mathrm{T}} \tag{4.41}$$

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{n \in C_k} \mathbf{x}_n \tag{4.42}$$

- Total covariance matrix:

$$\mathbf{S}_{\mathrm{T}} = \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^{\mathrm{T}} \tag{4.43}$$

where $\mathbf{m}$ is the mean of the total data set

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n = \frac{1}{N} \sum_{k=1}^{K} N_k \mathbf{m}_k \tag{4.44}$$

- Decomposition of total covariance matrix (4.43):

$$\mathbf{S}_{\mathrm{T}} = \mathbf{S}_{\mathrm{W}} + \mathbf{S}_{\mathrm{B}} \tag{4.45}$$

where

$$\mathbf{S}_{\mathrm{B}} = \sum_{k=1}^{K} N_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^{\mathrm{T}} \tag{4.46}$$

Definition of covariance matrix in projected space

- We can define similar matrices in the projected D'-dimensional $\mathbf{y}$-space

$$\mathbf{s}_{\mathrm{W}} = \sum_{k=1}^{K} \sum_{n \in C_k} (\mathbf{y}_n - \mu_k)(\mathbf{y}_n - \mu_k)^{\mathrm{T}} \tag{4.47}$$

$$\mathbf{s}_{\mathrm{B}} = \sum_{k=1}^{K} N_k (\mu_k - \mu)(\mu_k - \mu)^{\mathrm{T}} \tag{4.48}$$

where

$$\mu_k = \frac{1}{N_k} \sum_{n \in C_k} \mathbf{y}_n, \quad \mu = \frac{1}{N} \sum_{k=1}^{K} N_k \mu_k \tag{4.49}$$

Construction of a scalar criterion

- Construction of a scalar criterion
  - that is large when the between-class covariance is large and when the within-class covariance is small

$$J(\mathbf{W}) = \mathrm{Tr}\left\{\mathbf{s}_{\mathrm{W}}^{-1}\mathbf{s}_{\mathrm{B}}\right\} \tag{4.50}$$

- This criterion can be rewritten as an explicit function of the projection matrix $\mathbf{W}$

$$J(\mathbf{W}) = \mathrm{Tr}\left\{(\mathbf{W}\mathbf{S}_{\mathrm{W}}\mathbf{W}^{\mathrm{T}})^{-1}(\mathbf{W}\mathbf{S}_{\mathrm{B}}\mathbf{W}^{\mathrm{T}})\right\} \tag{4.51}$$

- The projection matrix $\mathbf{W}$ is determined by the eigenvectors of $\mathbf{S}_{\mathbf{W}}^{-1}\mathbf{S}_{\mathbf{B}}$ that correspond to the D' largest eigenvalues

$\mathbf{W}$ is determined by the eigenvectors of $\mathbf{S_W^{-1}S_B}$ that correspond to the D' largest eigenvalues

Taking the derivative of (4.51) with respect to $\mathbf{W}$,

$$\begin{aligned}\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} &= \frac{\partial}{\partial \mathbf{W}}\left\{\mathrm{Tr}\left\{\mathbf{s}_{\mathrm{W}}^{-1}\mathbf{s}_{\mathrm{B}}\right\}\right\} = \frac{\partial}{\partial \mathbf{W}}\left\{\mathrm{Tr}\left\{(\mathbf{W}\mathbf{S}_{\mathrm{W}}\mathbf{W}^{\mathrm{T}})^{-1}(\mathbf{W}\mathbf{S}_{\mathrm{B}}\mathbf{W}^{\mathrm{T}})\right\}\right\}\\ &= -2\mathbf{S_W}\mathbf{W}(\mathbf{W^T}\mathbf{S_W}\mathbf{W})^{-1}(\mathbf{W^T}\mathbf{S_B}\mathbf{W})(\mathbf{W^T}\mathbf{S_W}\mathbf{W})^{-1}\\ &+ 2\mathbf{S_B}\mathbf{W}(\mathbf{W^T}\mathbf{S_W}\mathbf{W})^{-1}\\ &= -2\mathbf{S_W}\mathbf{W}\mathbf{s_W}^{-1}\mathbf{s_B}\mathbf{s_W}^{-1} + 2\mathbf{S_B}\mathbf{W}\mathbf{s_W}^{-1} \end{aligned} \qquad \text{(F1)}$$

Setting equation (F1) to zero,

$$\begin{aligned}\mathbf{S_B}\mathbf{W}\mathbf{s_W}^{-1} &= \mathbf{S_W}\mathbf{W}\mathbf{s_W}^{-1}\mathbf{s_B}\mathbf{s_W}^{-1}\\ \mathbf{S_B}\mathbf{W} &= \mathbf{S_W}\mathbf{W}\mathbf{s_W}^{-1}\mathbf{s_B}\\ (\mathbf{S_W}^{-1}\mathbf{S_B})\mathbf{W} &= \mathbf{W}(\mathbf{s_W}^{-1}\mathbf{s_B})\end{aligned} \qquad \text{(F2)}$$

Preparation for proof

- Two matrices $\mathbf{s_B}$ and $\mathbf{s_W}$ can be simultaneously diagonalized to $\mathbf{P}$ and $\mathbf{Q}$ by a linear transformation $\mathbf{z} = \mathbf{B}^{\mathrm{T}}\mathbf{y}$

$$\mathbf{B^T s_B B} = \mathbf{P_b},\ \mathbf{B^T s_W B} = \mathbf{Q_w} \tag{F3}$$

where $\mathbf{B}$ is a $D' \times D'$ regular matrix and $\mathbf{B}^{-1}$ exists

- The criterion value is invariant under this regular mapping from $\mathbf{y}$ to $\mathbf{z}$

$$\begin{aligned}\mathrm{Tr}\left\{\mathbf{Q_w}^{-1}\mathbf{P_b}\right\} &= \mathrm{Tr}\left\{(\mathbf{B^T s_W B})^{-1}(\mathbf{B^T s_B B})\right\} = \mathrm{Tr}\left\{\mathbf{B^{-1} s_W^{-1} B^{T-1} B^T s_B B}\right\} \\ &= \mathrm{Tr}\left\{\mathbf{B^{-1} s_W^{-1} s_B B}\right\} = \mathrm{Tr}\left\{\mathbf{s_W^{-1} s_B}\right\}\end{aligned} \tag{F4}$$

- "Similarity invariance": The traces are invariant with respect to similarity , i.e. if P is a regular matrix, $\mathrm{Tr}(P^{-1}XP) = \mathrm{Tr}(X)$

Condition that satisfy (F2), to maximize the criterion (4.50)

$$(\mathbf{S_W}^{-1}\mathbf{S_B})\mathbf{W} = \mathbf{W}(\mathbf{s_W}^{-1}\mathbf{s_B}) \tag{F2}$$

(F2) is rewritten by using (F3)

$$\begin{aligned}
(\mathbf{S_W}^{-1}\mathbf{S_B})\mathbf{W} &= \mathbf{W}(\mathbf{B}\mathbf{Q_w}^{-1}\mathbf{B}^{-1}\mathbf{B}\mathbf{P_b}\mathbf{B}^{-1}) \\
(\mathbf{S_W}^{-1}\mathbf{S_B})\mathbf{W} &= \mathbf{W}(\mathbf{B}\mathbf{Q_w}^{-1}\mathbf{P_b}\mathbf{B}^{-1}) \\
(\mathbf{S_W}^{-1}\mathbf{S_B})\mathbf{W}\mathbf{B} &= \mathbf{W}\mathbf{B}\mathbf{Q_w}^{-1}\mathbf{P_b} \\
(\mathbf{S_W}^{-1}\mathbf{S_B})\mathbf{W}\mathbf{B} &= (\mathbf{W}\mathbf{B})\mathbf{\Lambda} \qquad \text{(F5)}
\end{aligned}$$

- The components of $\mathbf{\Lambda}$ and the column vectors of $\mathbf{WB}$ are the D' eigenvalues and eigenvectors of $\mathbf{S_W}^{-1}\mathbf{S_B}$ respectively

The trace of a matrix is the summation of the eigenvalues

- μ and λ are eigenvalues of $\left\{\mathbf{S}_{\mathrm{W}}^{-1}\mathbf{S}_{\mathrm{B}}\right\}$ and $\left\{\mathbf{s}_{\mathrm{W}}^{-1}\mathbf{s}_{\mathrm{B}}\right\}$ respectively

$$\mathrm{Tr}\left\{\mathbf{S}_{\mathrm{W}}^{-1}\mathbf{S}_{\mathrm{B}}\right\} = \mu_1 + \cdots + \mu_{D'} + \cdots + \mu_D \tag{F6}$$

$$J(\mathbf{W}) = \mathrm{Tr}\left\{\mathbf{s}_{\mathrm{W}}^{-1}\mathbf{s}_{\mathrm{B}}\right\} = \lambda_1 + \cdots + \lambda_{D'} \tag{F7}$$

- The eigenvalues λ of (F7) are also the eigenvalues of (F6), from (F5)

Therefore, the projection matrix $\mathbf{W}$ is determined by eigenvectors of $\mathbf{S}_{\mathbf{W}}^{-1}\mathbf{S}_{\mathbf{B}}$ that correspond to the D' largest eigenvalues

$K - 1$ or more future can not be found by this means

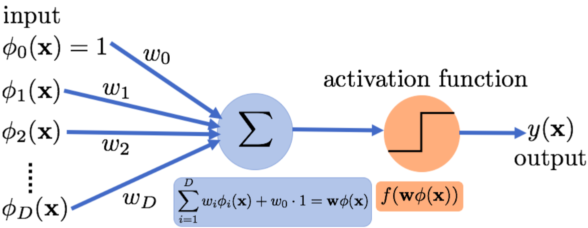
- $\mathbf{S_B}$ is composed of the sum of K matrices
  - Each of K matrices from an outer product of two vectors is rank 1
  - Only $(K - 1)$ of these matrices are independent as a result of constraint (4.44) that is linear combination in equation of mean
- Thus, $\mathbf{S_B}$ has rank at most equal to $(K - 1)$ and has nonzero eivenvalues at most $(K - 1)$

$$\Downarrow$$

This shows that the projection onto the $(K - 1)$-dimensional subspace spanned by the eigenvectors of $\mathbf{S_B}$ does not alter the value of $J(\mathbf{W})$

Outline

Perceptron: Two class model



- A generalized linear model:

$$y(\mathbf{x}) = f(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})) \tag{4.52}$$

- The nonlinear activation function:

$$f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases} \tag{4.53}$$

- The main difference compared to the methods we've seen so far is the learning algorithm

Perceptron criterion

- Target values: $t = 1$ for class C_1 and $t = -1$ for class C_2
- Minimization of the number of mis-classified examples
  - An example is mis-classified if $\mathbf{w}^{\mathbf{T}}\phi(\mathbf{x}_n)t_n < 0$
  - Perceptron criterion:

$$E_p(\mathbf{w}) = -\sum_{n\in\mathcal{M}} \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)t_n \qquad (4.54)$$

Sum over mis-classified examples only

Perceptron Learning Algorithm

- Minimize the error function using stochastic gradient descent (gradient descent per example):

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_{\mathrm{P}}(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta \phi(\mathbf{x}_n) t_n \tag{4.55}$$

- Iterate over all training examples, only change **w** if the example is mis-classified
- Guarantee to converge only if data are linearly separable
- May take many iterations
- Sensitive to initialization

The effect of a single update in the perceptron learning algorithm

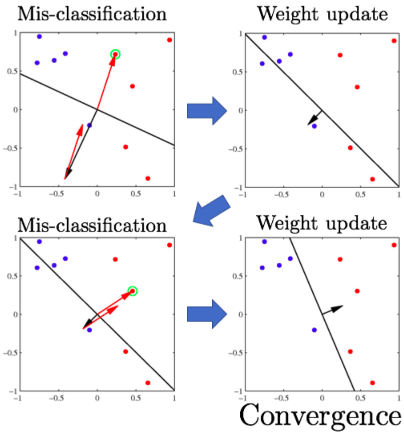
$$-\mathbf{w}^{(\tau+1)\mathrm{T}}\phi(\mathbf{x}_n)t_n = -\mathbf{w}^{(\tau)\mathrm{T}}\phi(\mathbf{x}_n)t_n - (\phi(\mathbf{x}_n)t_n)^{\mathrm{T}}\phi(\mathbf{x}_n)t_n < -\mathbf{w}^{(\tau)\mathrm{T}}\phi(\mathbf{x}_n)t_n \tag{4.56}$$

- The contribution to the error from a mis-classified pattern decrease
- The contribution to the error from the other mis-classified patterns may not decrease

$$\Downarrow$$

The perceptron learning rule is not guaranteed to reduce the total error function at each stage

The convergence of the perceptron learning algorithm



- Note that there are many hyperplanes with 0 error
  - Supoort vector machine is a nice way of choosing one