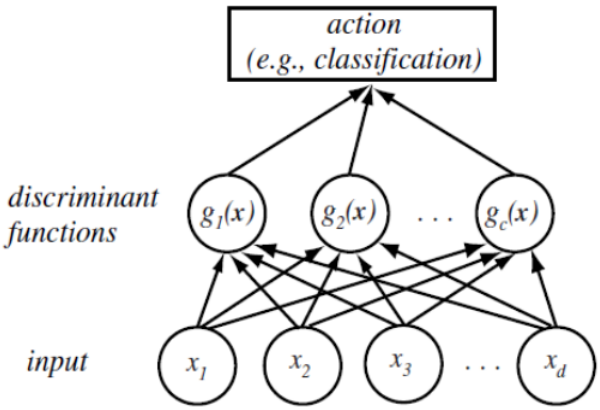
Chapter 5: Linear Discriminant Functions

5.1 Discriminant Functions

Discriminant Functions:

$$
\begin{aligned}
&g_i : \mathbb{R}^d \to \mathbb{R} \quad (i \le i \le c) \\
&\text{Decide } w_i \\
&\quad \text{if } g_i(x) > g_j(x) \quad \text{for all } i \neq j
\end{aligned}
$$

- Useful way to represent classifier
- One function per category



Minimum risk: $g_i(x) = -R(\alpha_i|x) \quad (1 \le i \le c)$

Minimum-error-rate: $g_i(x) = P(w_i|x) \quad (1 \le i \le c)$

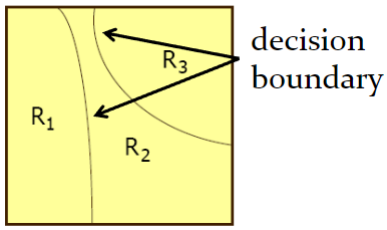
Decision region

$$
\begin{aligned}
&c \text{ discriminant functions} \\
&g_i(\cdot) \quad (1 \le i \le c)
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
&c \text{ decision regions} \\
&\mathcal{R}_i \subset R^d \quad (1 \le i \le c)
\end{aligned}
$$

$$
R_i = \{x | x \in R^d : \; g_i(x) > g_j(x) \quad \forall j \neq i\}
$$

$$
\text{where } R_i \cap R_j = \emptyset \quad (i \neq j) \text{ and } \bigcup_{i=1}^{c} R_i = R^d
$$

Decision Boundary



surface in feature space where ties occur among several largest discriminant functions

5.2 Linear Discriminant Functions

$$g_i(x) = w_i^T x + w_{i0}$$

- w_i: **weight vector** (d-dimensional)
- w_{i0}: **bias/threshold** (scalar)

5.2.1 The two-category case

$$\begin{aligned} g_1(x) &= w_1^T x + w_{10} \\ g_2(x) &= w_2^T x + w_{20} \end{aligned} \xrightarrow{g(x)=g_1(x)-g_2(x)} \begin{aligned} &\text{Decide } w_1 \text{ if } g(x) > 0 \\ &\text{Decide } w_2 \text{ otherwise} \end{aligned}$$

Hence we have:

$$\begin{aligned} g(x) = g_1(x) - g_2(x) &= (w_1^T x + w_{10}) - (w_2^T x + w_{20}) \\ &= (w_1 - w_2)^T x + (w_{10} - w_{20}) \end{aligned}$$

$$\text{let } \begin{aligned} w &= w_1 - w_2 \\ b &= w_{10} - w_{20} \end{aligned} \quad \Rightarrow \quad g(x) = w^T x + b$$

- It suffice to consider only $d+1$ parameters (w and b) instead of $2(d+1)$ parameters under two-category case

---

Training set

$$D^* = \{(x_i, w_i) | i = 1, 2, \ldots, n\} \quad (x_i \in R^d, w_i \in \{-1, +1\})$$

The task

Determine $g(x) = w^T x + b$ which can classify all training examples in $\mathcal{D}^*$ correctly

$$\begin{aligned} g(x_i) &= w^T x_i + b > 0 \quad \text{if } w_i = +1 \\ g(x_i) &= w^T x_i + b < 0 \quad \text{if } w_i = -1 \end{aligned} \quad \Longrightarrow \quad w_i \cdot (w^T x_i + b) > 0 \quad (i = 1, 2, \ldots, n)$$

Solution to (w,b):

Minimize a **criterion/objective function** (准则函数) $J(w, b)$ based on the training examples $\{(x_i, w_i) | i = 1, 2, \ldots, n\}$

- $J(w, b) = -\sum_{i=1}^{n} \text{sign}[w_i \cdot g(x_i)]$

- $J(w,b)=-\sum_{i=1}^{n}w_i\cdot g(x_i)$
- $J(w,b)=\sum_{i=1}^{n}\big(g(x_i)-w_i\big)^2$

Taylor Expansion:

$$f(x+\Delta x)=f(x)+\nabla f(x)^T\Delta x+O(\Delta x^T\Delta x)$$

We set Δx to be **negatively proportional** to the gradient at x, i.e. $\Delta x=-\eta\cdot\nabla f(x)$ (η being a small positive scalar)

Hence we have:

$$f(x+\Delta x)=f(x)-\eta\cdot\nabla f(x)^T\nabla f(x)+O(\Delta x^T\Delta x)\leq f(x)$$

- $\nabla f(x)^T\nabla f(x)$: being non-negative
- $(\Delta x^T\Delta x)$: ignored when $(\Delta x^T\Delta x)$ is small

---

Gradient Descent:

To minimize some d-variate function $f(\cdot)$, the general gradient descent techiniques work in the following **iterative way**:

1. Set **learning rate** $\eta>0$ and a small **threshold** $\epsilon>0$
2. Randomly initialize $x_0\in\mathcal{R}^d$ as the **starting point** and set $k=0$
3. **do** $k=k+1, x_k=x_{k-1}-\eta\cdot\nabla f(x_{k-1})$, **until** $|f(x_k)-f(x_{k-1})|<\epsilon$
    - $x_k=x_{k-1}-\eta\cdot\nabla f(x_{k-1})$: **gradient descent step**
4. Return x_k and $f(x_k)$

For the two-category case, we are ought to **choose certain criterion function** $J(w,b)$ defined over the training set $\mathcal{D}^*$, and then **invoke the standard gradient descent prodcedure** on the $(d+1)$-variate function $J(\cdot,\cdot)$ to determine (w,b)

1. $J(w,b)=-\sum_{i=1}^{n}w_i\cdot g(x_i)\Rightarrow\nabla J(w,b)=-\sum_{i=1}^{n}w_i\cdot\begin{bmatrix}x_i\\1\end{bmatrix}$
2. $J(w,b)=\sum_{i=1}^{n}(g(x_i)-w_i)^2\Rightarrow\nabla J(w,b)=2\cdot\sum_{i=1}^{n}(w^tx_i+b-w_i)\cdot\begin{bmatrix}x_i\\1\end{bmatrix}$