Discriminant-based Classification

- In classification with \( K \) classes \((C_1, C_2, \ldots, C_k)\):
  - We defined discriminant function \( g_j(x), j=1,2,\ldots,K \)
  - then given any test example \( x \), we chose (predicted) its class label as \( C_i \) if \( g_i(x) \) was the maximum among \( g_1(x), g_2(x), \ldots, g_k(x) \)

- In previous chapters we have:
  - Used \( g_i(x) = \log P(C_i \mid x) \)
  - This is called likelihood classification
  - Where we used maximum likelihood estimate technique for estimate class likelihood \( P(x \mid C_i) \)
Likelihood- vs. Discriminant-based Classification

- **Likelihood-based:** Assume a model for $p(x \mid C_i)$, use Bayes’ rule to calculate $P(C_i \mid x)$
  - $g_i(x) = \log P(C_i \mid x)$
  - This requires estimating class conditional densities $P(x \mid C_i)$
  - For high-dimensional data (many attributes/features), estimating class conditional densities itself is a difficult task

- **Discriminant-based:** Assume a model for $g_i(x \mid \Phi_i)$; no density estimation
  - Parameters $\Phi_i$ describe the class boundary
  - Estimating the class boundary is enough for performing classification
  - No need to accurately estimate the densities inside the boundaries
Linear Discriminant

- Linear discriminant:
  \[ g_i(x \mid w_i, w_{i0}) = w_i^T x + w_{i0} = \sum_{j=1}^{d} w_{ij} x_j + w_{i0} \]

- Advantages:
  - Simple: \( O(d) \) space/computation (\( d \) is the number of features)
  - Knowledge extraction: Weighted sum of attributes; positive/negative weights, magnitudes (credit scoring)
  - Optimal when \( p(x \mid C_i) \) are Gaussian with shared cov matrix; useful when classes are (almost) linearly separable
Generalized Linear Model

- Quadratic discriminant:

\[
g_i(x | W_i, w_i, w_{i0}) = x^T W_i x + w_i^T x + w_{i0}
\]

- Higher-order (product) terms:

\[
z_1 = x_1, \ z_2 = x_2, \ z_3 = x_1^2, \ z_4 = x_2^2, \ z_5 = x_1 x_2
\]

Map from \( x \) to \( z \) using nonlinear basis functions and use a linear discriminant in \( z \)-space

\[
g_i(x) = \sum_{j=1}^{k} w_{ij} \phi_j(x)
\]
Generalized Linear Model

- Example of non-linear basis functions:
  - $\sin(x_1)$
  - $\exp(-(x_1-m)^2/c)$
  - $\exp(-||x-m||^2/c)$
  - $\log(x_2)$
  - $1(x_1>c)$
  - $1(ax_1 + bx_2 > c)$
Two Classes

\[ g(x) = w_1x_1 + w_2x_2 + w_0 = 0 \]

- \( g(x) < 0 \) for \( C_2 \)
- \( g(x) > 0 \) for \( C_1 \)

\[ g(x) = g_1(x) - g_2(x) \]
\[ = (w_1^T x + w_{10}) - (w_2^T x + w_{20}) \]
\[ = (w_1 - w_2)^T x + (w_{10} - w_{20}) \]
\[ = w^T x + w_0 \]

Choose:
\[ \begin{cases} 
C_1 & \text{if } g(x) > 0 \\
C_2 & \text{otherwise}
\end{cases} \]
Geometry

\[ g(x) = 0 \]

\[ |w_0| / |w| \]

\[ |g(x)| / |w| \]
Let the discriminant function is given by \( g(x) = w_1 x_1 + w_2 x_2 + w_0 = w^T x + w_0 \), where \( w = (w_1, w_2)^T \).

Take any two points \( x^1, x^2 \), lying on the decision surface (boundary) \( g(x) = 0 \):

\[ g(x^1) = g(x^2) = 0 \]
\[ w^T x^1 + w_0 = w^T x^2 + w_0 \Rightarrow w^T (x^1 - x^2) = 0 \]

Note that \( x^1 - x^2 \) is a vector lying on the decision surface (hyperplane), which means \( w \) is normal to any vector lying on the decision surface.
Any data point $x$ can be written as a sum of two vectors as follows:

- $x = x_p + r(w/||w||)$
  - $x_p$ is normal projection of $x$ on to decision hyperplane ($x_p$ lies on the decision hyperplane)
  - $r$ is distance of $x$ to the hyperplane

$$g(x) = w^T x + w_0 = w^T(x_p + r(w/||w||)) + w_0 = (w^T x_p + w_0) + r(w^T w)/||w|| = 0 + (r||w||^2/||w||) = r||w|| \implies r = g(x)/||w||$$

Similarly if $x=0$, $r$ will denote distance of the hyperplane from the origin

- $g(0) = w_0 = r||w|| \implies r = w_0/||w||$
Multiple Classes

Discriminant function for the $i^{th}$ class is:

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

Choose $C_i$ if

$$g_i(x) = \max_{j=1}^{\kappa} g_j(x)$$

Classes are linearly separable
During testing, given $x$, ideally we should have only one $g_j(x)$, $j=1,2,\ldots,K$ greater than zero and all others should be less than 0.

However, this is not always the case:
- Positive half spaces of the hyperplane's may overlap.
- Or we may have all $g_j(x)<0$.
- These may be taken as “reject” case.

Remembering that $|g_i(x)|/||w_i||$ is the distance from the input point to the decision hyperplane, assuming all $w_i$ have similar length, this assigns point to the class (among all $g_i(x)>0$) to whose decision hyperplane the point is most distant.
It possible that classes are not linearly separable but are pairwise linearly separable.

- We can use \( K(K-1)/2 \) linear discriminants \( g_{ij}(x) \) to classify

\[
g_{ij}(x|w_{ij}, w_{ij0}) = w_{ij}^T x + w_{ij0}
\]

- Parameters are computed during training so as to have

\[
g_{ij}(x) = \begin{cases} 
> 0 & \text{if } x \in C_i \\
\leq 0 & \text{if } x \in C_j \\
\text{don't care} & \text{otherwise}
\end{cases}
\]

- Classification is performed as follows

choose \( C_i \) if

\[
\forall j \neq i, g_{ij}(x) > 0
\]

For an input to be assigned to class \( C_1 \), it should be on the positive side of \( H_{12} \) and \( H_{31} \). We don’t care about the value of \( H_{23} \).
If the class densities are Gaussian, and share a common covariance matrix, the discriminant function is linear, i.e., when 
\[ p(x \mid C_i) \sim N(\mu_i, \Sigma) \]
\[ g_i(x \mid w_i, w_{i0}) = w_i^T x + w_{i0} \]
\[ w_i = \Sigma^{-1} \mu_i \quad w_{i0} = -\frac{1}{2} \mu_i^T \Sigma^{-1} \mu_i + \log P(C_i) \]

For the special case when there are two classes, we define,
\[ y = P(C_1 \mid x) \text{ and } P(C_2 \mid x) = 1 - y \]
\[ y > 0.5 \]
choose \( C_1 \) if \( y / (1 - y) > 1 \) and \( C_2 \) otherwise
\[ \log [y / (1 - y)] > 0 \]

\( \log(y / (1 - y)) \) is known as logit transformation or log odds of \( y \)
In case of two normal classes sharing a common covariance matrix, the log odds is linear

\[
\text{logit}(P(C_1|\mathbf{x})) = \log \frac{P(C_1|\mathbf{x})}{1 - P(C_1|\mathbf{x})} = \log \frac{P(C_1|\mathbf{x})}{P(C_2|\mathbf{x})} \\
= \log \frac{p(\mathbf{x}|C_1)}{p(\mathbf{x}|C_2)} + \log \frac{P(C_1)}{P(C_2)} \\
= \log \frac{(2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left[-(1/2)(\mathbf{x} - \mu_1)^T \Sigma^{-1} (\mathbf{x} - \mu_1)\right]}{(2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left[-(1/2)(\mathbf{x} - \mu_2)^T \Sigma^{-1} (\mathbf{x} - \mu_2)\right]} + \log \frac{P(C_1)}{P(C_2)} \\
= \mathbf{w}^T \mathbf{x} + w_0
\]

where \( \mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2) \) \( w_0 = -\frac{1}{2} (\mu_1 + \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) \)

The inverse of logit is logistic or sigmoid function

\[
\log \frac{P(C_1|\mathbf{x})}{1 - P(C_1|\mathbf{x})} = \mathbf{w}^T \mathbf{x} + w_0
\]

\[
P(C_1|\mathbf{x}) = \text{sigmoid}(\mathbf{w}^T \mathbf{x} + w_0) = \frac{1}{1 + \exp\left[-(\mathbf{w}^T \mathbf{x} + w_0)\right]}
\]
Calculate $g(x) = w^T x + w_0$ and choose $C_1$ if $g(x) > 0$, or
Calculate $y = \text{sigmoid}(w^T x + w_0)$ and choose $C_1$ if $y > 0.5$
Logistic regression is a classification method where in case of binary classification, the log ratio of $p(C_1 | x)$ and $p(C_2 | x)$ is modeled as a linear function:

$$\text{logit}(P(C_1|x)) = \log \frac{p(C_1 | x)}{p(C_2 | x)} = w^T x + w_0$$

- Since we are modeling ratio of posterior probability directly, there is no need for density estimation i.e. $p(x | C_1)$ and $p(x | C_2)$

- Note that this is slightly different version than what is given in the book but this is the most widely version in practice

- Rearranging, we can write

$$P(C_1|x) = \frac{1}{1 + \exp[-(w^T x + w_0)]} \quad \text{and} \quad P(C_2|x) = \frac{\exp[-(w^T x + w_0)]}{1 + \exp[-(w^T x + w_0)]}$$

- Given $x$, predicted label is $C_1$ when $P(C_1 | x) > P(C_2 | x)$
  - Or alternatively, when $w^T x + w_0 > 0$

- To classify using this model, that we need to know what $w$ and $w_0$ is
  - How do we find $w$ and $w_0$?
Logistic Regression for binary classification

- Given training data \( \mathcal{X} = \{x^t, r^t\}_{t=1}^N \) \( r^t | x^t \) is modeled as Bernoulli distribution

- \( r^t | x^t \sim \text{Bernoulli}(y^t) \) where, \( y^t = P(C_i | x^t) = \frac{1}{1 + \exp[-(w^T x^t + w_0)]} \)

- To estimate \( w \) and \( w_0 \), we can
  - Maximize the likelihood
    \[
    l(w, w_0 | \mathcal{X}) = \prod_t (y^t)^{r^t} (1 - y^t)^{1-r^t}
    \]
  - Or, equivalently maximize the log-likelihood
    \[
    L(w, w_0 | \mathcal{X}) = \sum_t r^t \log y^t + (1 - r^t) \log (1 - y^t)
    \]
  - Or equivalently, minimize negative log-likelihood
    \[
    E(w, w_0 | \mathcal{X}) = -L(w, w_0 | \mathcal{X}) = -\sum_t r^t \log y^t + (1 - r^t) \log (1 - y^t)
    \]
Gradient-Descent

- $E(w | X)$ is error with parameters $w$ on sample $X$
  - $w^* = \arg \min_w E(w | X)$

- **Gradient**
  \[
  \nabla_w E = \left[ \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_d} \right]^T
  \]

- **Gradient-descent**:
  Starts from random $w$ and updates $w$ iteratively in the negative direction of gradient
Gradient-Descent

\[ \Delta w_i = -\eta \frac{\partial E}{\partial w_i}, \forall i \]

\[ w_i = w_i + \Delta w_i \]
Gradient-Descent

- This is a technique for finding the local minimum of a multivariate function.
- Consider a function $f : \mathbb{R}^n \to \mathbb{R}$
  - Suppose for any $x \in \mathbb{R}^n$, the function value $f(x)$ at $x = (x_1, \ldots, x_n)$ is given.
  - In which direction $v$ should we go next from $x$ so that at $x + v$, $f(x + v) \leq f(x)$?
- Consider Taylor series expansion of $f(x)$ around $x$:
  $f(x + v) \approx f(x) + \nabla f(x)^T v$
  - Where, $\nabla f(x) = \left( \frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n} \right)$ is the gradient of $f$ at $x$
  - Then, $f(x) - f(x + v) \approx -\nabla f(x)^T v$
  - Therefore, $f$ will reduce maximally locally at $x$ in the direction $v = -\nabla f(x)$
Gradient-Descent

Used for finding local minimum of a multivariate function

\[ f : \mathbb{R}^n \to \mathbb{R} \]

**Input:** stepsize \( \eta > 0 \), tolerance parameter \( \varepsilon > 0 \)

1. Start at some point \( x_0 \in \mathbb{R}^n \)
2. For \( i \geq 1 \), repeat until \( \|x_{i+1} - x_i\| \leq \varepsilon \)
   1. \( x_{i+1} = x_i - \eta \nabla f(x_i) \)
Training: Gradient-Descent

\[ E(w, w_0 | \mathcal{X}) = -\sum_t r^t \log y^t + (1 - r^t) \log (1 - y^t) \]

If \( y = \text{sigmoid}(a) \)

\[ \frac{dy}{da} = y(1 - y) \]

\[ \Delta w_j = -\eta \frac{\partial E}{\partial w_j} = \eta \sum_t \left( \frac{r^t}{y^t} - \frac{1 - r^t}{1 - y^t} \right) y^t (1 - y^t) x_j^t \]

\[ = \eta \sum_t (r^t - y^t) x_j^t, j = 1, \ldots, d \]

\[ \Delta w_0 = -\eta \frac{\partial E}{\partial w_0} = \eta \sum_t (r^t - y^t) \]
For $j = 0, \ldots, d$

$$w_j \leftarrow \text{rand}(-0.01, 0.01)$$

Repeat

For $j = 0, \ldots, d$

$$\Delta w_j \leftarrow 0$$

For $t = 1, \ldots, N$

$$o \leftarrow 0$$

For $j = 0, \ldots, d$

$$o \leftarrow o + w_j x^t_j$$

$$y \leftarrow \text{sigmoid}(o)$$

$$\Delta w_j \leftarrow \Delta w_j + (r^t - y) x^t_j$$

For $j = 0, \ldots, d$

$$w_j \leftarrow w_j + \eta \Delta w_j$$

Until convergence
Logistic Regression for K classes ($K > 2$)

- Given training data $\mathcal{X} = \{x^t, r^t\}_{t=1}^{N_t}$ $x^t$ is modeled as Multinomial distribution
  
  $r^t|x^t \sim \text{Mult}_K (1, y')$ where, $y'_i = P(C_i|x) = \frac{\exp[w^T_i x^t + w_{i0}]}{\sum_{j=1}^K \exp[w^T_j x^t + w_{j0}]}$, $i = 1, ..., K$

- To estimate $w_1, w_2, ..., w_K$ and $w_{10}, w_{20}, ..., w_{K0}$ we can
  
  - Maximize the likelihood
    
    \[ l \left( \{w_i, w_{i0}\}_{i=1}^K \mid \mathcal{X} \right) = \prod_t \prod_i (y'_i)^{r'_i} \]

  - Or equivalently, minimize negative log-likelihood
    
    \[ E \left( \{w_i, w_{i0}\}_{i=1}^K \mid \mathcal{X} \right) = -\sum_t r'_i \log y'_i \]

  - The gradient can computed using simple formula
    
    \[ \Delta w_j = \eta \sum_t (r'_j - y'_j)x^t \quad \Delta w_{j0} = \eta \sum_t (r'_j - y'_j) \]

- Using gradient descent, we can have simple algorithm for logistic regression for K class classification problem
For $i = 1, \ldots, K$, For $j = 0, \ldots, d$, $w_{ij} \leftarrow \text{rand}(-0.01, 0.01)$

Repeat

For $i = 1, \ldots, K$, For $j = 0, \ldots, d$, $\Delta w_{ij} \leftarrow 0$

For $t = 1, \ldots, N$

For $i = 1, \ldots, K$

\[ o_i \leftarrow 0 \]

For $j = 0, \ldots, d$

\[ o_i \leftarrow o_i + w_{ij}x^t_j \]

For $i = 1, \ldots, K$

\[ y_i \leftarrow \exp(o_i) / \sum_k \exp(o_k) \]

For $i = 1, \ldots, K$

For $j = 0, \ldots, d$

\[ \Delta w_{ij} \leftarrow \Delta w_{ij} + (r^t_i - y_i)x^t_j \]

For $i = 1, \ldots, K$

For $j = 0, \ldots, d$

\[ w_{ij} \leftarrow w_{ij} + \eta \Delta w_{ij} \]

Until convergence