The Bayes classifier

**Theorem**

The classifier \( h^* (x) := \arg \max_k \eta_k (x) \) satisfies

\[
R(h^*) = \min R(h)
\]

where the min is over all possible classifiers.

To calculate the Bayes classifier/Bayes risk, we need to know \( \eta_k (x) = \mathbb{P}[Y = k | X = x] \)

Alternatively, since \( \pi_k f_{X|Y} (x | k) = \eta_k (x) f_X (x) \), to find the maximum \( \eta_k (x) \) it is sufficient to know \( \pi_k f_{X|Y} (x | k) \)
Linear discriminant analysis (LDA)

In linear discriminant analysis (LDA), we make the (strong) assumption that

\[ X \mid Y = k \sim \mathcal{N}(\mu_k, \Sigma) \]

for \( k = 0, \ldots, K - 1 \)

Here \( \mathcal{N}(\mu, \Sigma) \) is the multivariate Gaussian/normal distribution with mean \( \mu \) and covariance matrix \( \Sigma \)

\[
\phi(x; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu) \right\}
\]

Note: Each class has the same covariance matrix \( \Sigma \)
Example

Suppose that $K = 2$

$$d_M^2(x; \hat{\mu}_0, \hat{\Sigma}) - 2 \log \hat{\pi}_0 \begin{cases} 0, & \text{if } 1, \\ 1, & \text{otherwise} \end{cases} = d_M^2(x; \hat{\mu}_1, \hat{\Sigma}) - 2 \log \hat{\pi}_1$$

It turns out that by setting

$$w = \hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0)$$

$$b = \frac{1}{2} \hat{\mu}_0^T \hat{\Sigma}^{-1} \hat{\mu}_0 - \frac{1}{2} \hat{\mu}_1^T \hat{\Sigma}^{-1} \hat{\mu}_1 + \log \frac{\hat{\pi}_1}{\hat{\pi}_0}$$

we can re-write this as

$$w^T x + b \begin{cases} 0, & \text{if } 1, \\ 1, & \text{otherwise} \end{cases} \quad \text{linear classifier}$$
Challenges for LDA

The generative model is rarely valid

Moreover, the number of parameters to be estimated is

- class prior probabilities: $K - 1$
- means: $Kd$
- covariance matrix: $\frac{1}{2}d(d + 1)$

If $d$ is small and $n$ is large, then we can accurately estimate these parameters (provably, using Hoeffding)

If $n$ is small and $d$ is large, then we have more parameters than observations, and will likely obtain very poor estimates

- first apply a dimensionality reduction technique to reduce $d$ (more on this later)
- assume a more structured covariance matrix
Another possible escape

Recall from the very beginning of the lecture that the Bayes classifier can be stated either in terms of maximizing $\pi_k f_{X|Y}(x|k)$ or $\eta_k(x)$.

In LDA, we are estimating $\pi_k f_{X|Y}(x|k)$, which is equivalent to the full joint distribution of $(X, Y)$.

All we really need is to be able to estimate $\eta_k(x)$ - we don’t need to know $f_X(x)$.

LDA commits one of the cardinal sins of machine learning:

*Never solve a more difficult problem as an intermediate step*

Is there a better approach?
Another look at plugin methods

Suppose \( K = 2 \)

Define \( \eta(x) = \eta_1(x) \)

\[ = 1 - \eta_0(x) \]

In this case, another way to express the Bayes classifier is as

\[ h^*(x) = \begin{cases} 1 & \text{if } \eta(x) \geq 1/2 \\ 0 & \text{if } \eta(x) < 1/2 \end{cases} \]

Note that we do not actually need to know the full distribution of \((X, Y)\) to express the Bayes classifier.

All we really need is to decide if \( \eta(x) \geq 1/2 \)
Gaussian case

Suppose that $K = 2$ and that $X | Y = k \sim N(\mu_k, \Sigma)$

$$
\eta(x) = \frac{\pi_1 \phi(x; \mu_1, \Sigma)}{\pi_1 \phi(x; \mu_1, \Sigma) + \pi_0 \phi(x; \mu_0, \Sigma)}
$$

$$
= \frac{\pi_1 e^{-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)}}{\pi_1 e^{-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)} + \pi_0 e^{-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)}}
$$

$$
= \frac{1}{1 + \frac{\pi_0}{\pi_1} e^{\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1) - \frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)}}
$$

$$
= \frac{1}{1 + e^{-(w^T x + b)}}
$$
Logistic regression

This observation gives rise to another class of plugin methods, the most important of which is logistic regression, which implements the following strategy:

1. Assume \( \eta(x) = \frac{1}{1 + e^{-(w^T x + b)}} \) (\( w \in \mathbb{R}^d, b \in \mathbb{R} \))

2. Directly estimate \( w, b \) (somehow) from the data

3. Plug the estimate

\[
\hat{\eta}(x) = \frac{1}{1 + e^{-(\hat{w}^T x + \hat{b})}}
\]

into the formula for the Bayes classifier.
The logistic function

The function $\frac{1}{1+e^{-t}}$ is called a **logistic** function (or a **sigmoid** function in other contexts)
The logistic regression classifier

Denote the logistic regression classifier by

$$\hat{h}(x) = 1 \{ \hat{\eta}(x) \geq 1/2 \}(x)$$

Note that $\hat{h}(x) = 1 \iff \hat{\eta}(x) \geq \frac{1}{2}$

$$\iff \frac{1}{1 + \exp \left( - (\hat{\mathbf{w}}^T x + \hat{b}) \right)} \geq \frac{1}{2}$$

$$\iff \exp \left( - (\hat{\mathbf{w}}^T x + \hat{b}) \right) \leq 1$$

$$\iff (\hat{\mathbf{w}}^T x + \hat{b}) \geq 0$$

So $\hat{h}(x) = \begin{cases} 1 & \text{if } \hat{\mathbf{w}}^T x + b \geq 0 \\ 0 & \text{otherwise} \end{cases}$
Example
Estimating the parameters

**Challenge:** How to estimate the parameters for

\[ \eta(x) = \frac{1}{1 + e^{-(w^T x + b)}} \]

One possibility: \( w = \Sigma^{-1} (\hat{\mu}_1 - \hat{\mu}_0) \)

\[ b = \frac{1}{2} \hat{\mu}_0^T \Sigma^{-1} \hat{\mu}_0 - \frac{1}{2} \hat{\mu}_1^T \Sigma^{-1} \hat{\mu}_1 + \log \frac{\hat{\pi}_1}{\hat{\pi}_0} \]

**Alternative:** *Maximum likelihood estimation*

For convenience, let’s let \( \theta = (b, w) \)

Note that \( \eta(x) \) is really a function of both \( x \) and \( \theta \), so we will use the notation \( \eta(x; \theta) \) to highlight this dependence.
The *a posteriori* probability of our data

Suppose that we knew $\theta$. Then we could compute

$$\mathbb{P}[y_i|x_i; \theta] = \mathbb{P}[Y_i = y_i|X_i = x_i; \theta]$$

$$= \begin{cases} 
\eta(x_i; \theta) & \text{if } y_i = 1 \\
1 - \eta(x_i; \theta) & \text{if } y_i = 0 
\end{cases}$$

$$= \eta(x_i; \theta)^{y_i}(1 - \eta(x_i; \theta))^{1-y_i}$$

Because of independence, we also have that

$$\mathbb{P}[y_1, \ldots, y_n|x_1, \ldots x_n; \theta] = \prod_{i=1}^{n} \mathbb{P}[y_i|x_i; \theta]$$

$$= \prod_{i=1}^{n} \eta(x_i; \theta)^{y_i}(1 - \eta(x_i; \theta))^{1-y_i}$$
Maximum likelihood estimation

We don’t actually know $\theta$, but we do know $y_1, \ldots, y_n$

Suppose we view $y_1, \ldots, y_n$ to be fixed, and view $\mathbb{P}[y_1, \ldots, y_n|x_1, \ldots x_n; \theta]$ as just a function of $\theta$

When we do this, $\mathcal{L}(\theta) = \mathbb{P}[y_1, \ldots, y_n|x_1, \ldots x_n; \theta]$ is called the likelihood (or likelihood function)

The method of maximum likelihood aims to estimate $\theta$ by finding the $\theta$ that maximizes the likelihood $\mathcal{L}(\theta)$

In practice, it is often more convenient to focus on maximizing the log-likelihood, i.e., $\log \mathcal{L}(\theta)$
The log-likelihood

To see why, note that the likelihood in our case is given by

\[ L(\theta) = \prod_{i=1}^{n} \eta(x_i; \theta)^{y_i} (1 - \eta(x_i; \theta))^{1-y_i} \]

Thus, the log-likelihood is given by

\[ \ell(\theta) = \log L(\theta) \]

\[ = \sum_{i=1}^{n} y_i \log \eta(x_i; \theta) + (1 - y_i) \log (1 - \eta(x_i; \theta)) \]

It is often easier to work with summations instead of products, and since the log is a monotonic transformation, maximizing \( \ell(\theta) \) is equivalent to maximizing \( L(\theta) \)
Simplifying the log-likelihood

Notation
• $\tilde{x} = [1, x(1), \ldots, x(d)]^T$
• $\theta = [b, w(1), \ldots, w(d)]^T$

This means that $w^T x + b = \theta^T \tilde{x}$, which lets us write

$$\eta(x_i; \theta) = \frac{1}{1 + e^{-\theta^T \tilde{x}_i}}$$

Thus, if we let $g(t) = \frac{1}{1 + e^{-t}}$, then we can write

$$\ell(\theta) = \sum_{i=1}^{n} y_i \log g(\theta^T \tilde{x}_i) + (1 - y_i) \log (1 - g(\theta^T \tilde{x}_i))$$
Simplifying the log-likelihood

Facts:

$$\log g(t) = \log \left( \frac{1}{1 + e^{-t}} \right) = -\log(1 + e^{-t})$$

$$\log(1 - g(t)) = \log \left( 1 - \frac{1}{1 + e^{-t}} \right) = \log \left( \frac{e^{-t}}{1 + e^{-t}} \right) = -t - \log(1 + e^{-t})$$

$$= \log \left( \frac{1}{1 + e^t} \right) = -\log(1 + e^t)$$
Simplifying the log-likelihood

**Facts:**

\[
\log g(t) = - \log(1 + e^{-t})
\]

\[
\log(1 - g(t)) = -t - \log(1 + e^{-t}) = - \log(1 + e^t)
\]

Thus

\[
\ell(\theta) = \sum_{i=1}^{n} y_i \log g(\theta^T \tilde{x}_i) + (1 - y_i) \log (1 - g(\theta^T \tilde{x}_i))
\]

\[
= \sum_{i=1}^{n} -y_i \log(1 + e^{-\theta^T \tilde{x}_i}) - \log(1 + e^{\theta^T \tilde{x}_i})
\]

\[
- y_i (-\theta^T \tilde{x}_i - \log(1 + e^{-\theta^T \tilde{x}_i}))
\]

\[
= \sum_{i=1}^{n} y_i \theta^T \tilde{x}_i - \log(1 + e^{\theta^T \tilde{x}_i})
\]
Maximizing the log-likelihood

How can we maximize

\[ \ell(\theta) = \sum_{i=1}^{n} y_i \theta^T \tilde{x}_i - \log(1 + e^{\theta^T \tilde{x}_i}) \]

with respect to \( \theta \)?

Find a \( \theta \) such that \( \nabla \ell(\theta) = \begin{bmatrix} \frac{\partial \ell(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial \ell(\theta)}{\partial \theta_{d+1}} \end{bmatrix} = 0 \)

(i.e., compute the partial derivatives and set them to zero)
Computing the gradient

It is not too hard to show that

\[ \nabla \ell(\theta) = \sum_{i=1}^{n} \nabla \left( y_i \theta^T \tilde{x}_i - \log(1 + e^{\theta^T \tilde{x}_i}) \right) \]

\[ = \sum_{i=1}^{n} y_i \tilde{x}_i - \tilde{x}_i e^{\theta^T \tilde{x}_i} \left( 1 + e^{\theta^T \tilde{x}_i} \right)^{-1} \]

\[ = \sum_{i=1}^{n} \tilde{x}_i(y_i - g(\theta^T \tilde{x}_i)) = 0 \]

This gives us \( d + 1 \) equations, but they are nonlinear and have no closed-form solution.
Throughout signal processing and machine learning, we will very often encounter problems of the form

$$\min_{x \in \mathbb{R}^d} f(x)$$

(or minimize $-\ell(\theta)$ for today)

In many (most?) cases, we cannot compute the solution simply by setting $\nabla f(x) = 0$ and solving for $x$

However, there are many powerful algorithms for finding $x$ using a computer
Gradient descent

A simple way to try to find the minimum of our objective function is to iteratively "roll downhill"

From \( x^0 \), take a step in the direction of the negative gradient

\[
x^1 = x^0 - \alpha_0 \nabla f(x) \bigg|_{x=x^0} \quad \alpha_0: "\text{step size}" \\
x^2 = x^1 - \alpha_1 \nabla f(x) \bigg|_{x=x^1} \\
\vdots
\]

\( x^0 \) \hspace{2cm} \( x^1 \)
Convergence of gradient descent

The core iteration of gradient descent is to compute

\[ x^{j+1} = x^j - \alpha_j \nabla f(x) \big|_{x=x^j} \]

Note that if \( \nabla f(x) \big|_{x=x^j} = 0 \), then we have found the minimum and \( x^{j+1} = x^j \), so the algorithm will terminate.

If \( f \) is convex and sufficiently smooth, then gradient descent (with a fixed step size \( \alpha \)) is guaranteed to converge to the global minimum of \( f \).
Step size matters!

Even though gradient descent provably converges, it can potentially take a while.
Step size matters!

Even though gradient descent provably converges, it can potentially take a while.
Newton’s method

Also known as the Newton-Raphson method, this approach can be viewed as simply using the second derivative to automatically select an appropriate step size.

\[ x^{j+1} = x^j - (\nabla^2 f(x))^{-1} \nabla f(x)|_{x=x^j} \]

**Hessian matrix**

\[ \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_d} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_d \partial x_1} & \frac{\partial^2 f}{\partial x_d \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_d^2} \end{bmatrix} \]
Optimization for logistic regression

The negative log-likelihood in logistic regression is a convex function

Both gradient descent and Newton’s method are common strategies for setting the parameters in logistic regression

Newton’s method is much faster when the dimension \( d \) is small, but is impractical when \( d \) is large

Why?

More on this on next week’s homework
Comparison of plugin methods

Naïve Bayes, LDA, and logistic regression are all plugin methods that result in linear classifiers

Naïve Bayes
- plugin method based on density estimation
- scales well to high-dimensions and naturally handles mixture of discrete and continuous features

Linear discriminant analysis
- better if Gaussianity assumptions are valid

Logistic regression
- models only the distribution of $Y|X$, not $(X, Y)$
- valid for a larger class of distributions
- fewer parameters to estimate
Beyond plugin methods

Plugin methods can be useful in practice, but ultimately they are very limited

- There are always distributions where our assumptions are violated
- If our assumptions are wrong, the output is totally unpredictable
- Can be hard to verify whether our assumptions are right
- Require solving a more difficult problem as an intermediate step

For most of the remainder of this course will focus on nonparametric methods that avoid making such strong assumptions about the (unknown) process generating the data