COMS 4771 Lecture 2

1. Classification problems (review of some probability)
2. Classifiers via generative models
3. Evaluating classifiers
CLASSIFICATION PROBLEMS
Terminology and notation

- Recall: $\mathcal{X}$ is the input space, and $\mathcal{Y}$ is the output space.

- Labeled example: $(x,y) \in \mathcal{X} \times \mathcal{Y}$.
  - Interpretation: $x$ represents the description or measurements of an object; $y$ is the category to which that object belongs.
  - Task: using labeled examples $(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n) \in \mathcal{X} \times \mathcal{Y}$, construct classifier $\hat{f}: \mathcal{X} \rightarrow \mathcal{Y}$ that usually predicts the correct class label.

- Note: possible to see both $(x,1)$ and $(x,2)$ for same input $x$. (Why is this realistic?)
Recall: $\mathcal{X}$ is the **input space**, and $\mathcal{Y}$ is the **output space**.

In classification problems, output space $\mathcal{Y}$ comprised of $K$ possible **classes** (or **categories**). For simplicity, we’ll just call them $\mathcal{Y} := \{1, 2, \ldots, K\}$.

(When $K = 2$, we typically use $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, +1\}$.)
Recall: $\mathcal{X}$ is the **input space**, and $\mathcal{Y}$ is the **output space**.

In classification problems, output space $\mathcal{Y}$ comprised of $K$ possible **classes** (or **categories**). For simplicity, we’ll just call them $\mathcal{Y} := \{1, 2, \ldots, K\}$. (When $K = 2$, we typically use $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, +1\}$.)

**Labeled example**: $(x, y) \in \mathcal{X} \times \mathcal{Y}$.

**Interpretation:**
- $x$ represents the description or measurements of an object;
- $y$ is the category to which that object belongs.
Recall: $\mathcal{X}$ is the input space, and $\mathcal{Y}$ is the output space.

In classification problems, output space $\mathcal{Y}$ comprised of $K$ possible classes (or categories).
For simplicity, we’ll just call them $\mathcal{Y} := \{1, 2, \ldots, K\}$.
(When $K = 2$, we typically use $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, +1\}$.)

Labeled example: $(x, y) \in \mathcal{X} \times \mathcal{Y}$.
Interpretation:
- $x$ represents the description or measurements of an object;
- $y$ is the category to which that object belongs.

Task: using labeled examples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, construct classifier $\hat{f}: \mathcal{X} \rightarrow \mathcal{Y}$ that usually predicts the correct class label.
Recall: \( \mathcal{X} \) is the **input space**, and \( \mathcal{Y} \) is the **output space**.

In classification problems, output space \( \mathcal{Y} \) comprised of \( K \) possible **classes** (or **categories**).

For simplicity, we’ll just call them \( \mathcal{Y} := \{1, 2, \ldots, K\} \).

(When \( K = 2 \), we typically use \( \mathcal{Y} = \{0, 1\} \) or \( \mathcal{Y} = \{-1, +1\} \).)

**Labeled example:** \((x, y) \in \mathcal{X} \times \mathcal{Y}\).

**Interpretation:**

- \( x \) represents the description or measurements of an object;
- \( y \) is the category to which that object belongs.

**Task:** using **labeled examples** \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}\), construct **classifier** \( \hat{f} : \mathcal{X} \to \mathcal{Y} \) that usually predicts the correct class label.

**Note:** possible to see both \((x, 1)\) and \((x, 2)\) for same input \(x\).

(Why is this realistic?)
How do we say how good a classifier is?

- Assume there’s a \textit{distribution} \( P \) over space of labeled examples \( \mathcal{X} \times \mathcal{Y} \).
How do we say how good a classifier is?

- Assume there’s a **distribution** $P$ over space of labeled examples $\mathcal{X} \times \mathcal{Y}$.
- $P$ is *unknown* (e.g., we don’t know its functional form), but it represents the population we care about.
How do we say how good a classifier is?

- Assume there’s a distribution $P$ over space of labeled examples $\mathcal{X} \times \mathcal{Y}$.
- $P$ is unknown (e.g., we don’t know its functional form), but it represents the population we care about.
- For any classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$, we care about its prediction accuracy:
  $$\Pr[f(X) = Y].$$
  where $(X, Y) \sim P$.

[ **Notation:** “$(X, Y) \sim P$” means that inside the argument to $\Pr[\cdot]$, $(X, Y)$ is a ($\mathcal{X} \times \mathcal{Y}$-valued) random variable with distribution $P$. ]
How do we say how good a classifier is?

- Assume there’s a distribution $P$ over space of labeled examples $\mathcal{X} \times \mathcal{Y}$.
- $P$ is unknown (e.g., we don’t know its functional form), but it represents the population we care about.
- For any classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$, we care about its prediction accuracy:
  \[
  \Pr[f(X) = Y].
  \]

  where $(X, Y) \sim P$.

  [ **Notation:** “$(X, Y) \sim P$” means that inside the argument to $\Pr[\cdot]$, $(X, Y)$ is a $(\mathcal{X} \times \mathcal{Y}$-valued) random variable with distribution $P$. ]

- **Prediction error:**
  \[
  \text{err}(f) := \Pr[f(X) \neq Y].
  \]
When is there any hope for finding a classifier with high accuracy?

Key assumption: Data \((X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\) are i.i.d. random labeled examples with distribution \(P\)—i.e., an i.i.d. sample from \(P\).

This assumption is the connection between what we've seen in the past to what we expect to see in the future.

What's next:

- What does an accurate classifier look like?
- How do we exploit the key assumption to construct an accurate classifier?
When is there any hope for finding a classifier with high accuracy?

**Key assumption**: Data \((X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\) are i.i.d. random labeled examples with distribution \(P\)—i.e., an i.i.d. sample from \(P\).

This assumption is the connection between what we’ve seen in the past to what we expect to see in the future.
When is there any hope for finding a classifier with high accuracy?

**Key assumption:** Data $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ are i.i.d. random labeled examples with distribution $P$—i.e., an i.i.d. sample from $P$.

This assumption is the connection between what we’ve seen in the past to what we expect to see in the future.

What’s next:

- What does an accurate classifier look like?
- How do we exploit the key assumption to construct an accurate classifier?
More notation

Let $Z$ be a random variable. What is the meaning of the following statement?

$\mathbb{E}[1\{Z = 1\}]$
Let $Z$ be a random variable. What is the meaning of the following statement?

$$E[\mathbb{1}\{Z = 1\}]$$

- **Indicator function**:

$$\mathbb{1}\{A\} = \begin{cases} 1 & \text{if } A \text{ is true;} \\ 0 & \text{if } A \text{ is false.} \end{cases}$$
Let $Z$ be a random variable. What is the meaning of the following statement?

$$\mathbb{E}[\mathbb{1}\{Z = 1\}]$$

**Indicator function:**

$$\mathbb{1}\{A\} = \begin{cases} 1 & \text{if } A \text{ is true;} \\ 0 & \text{if } A \text{ is false.} \end{cases}$$

**Expectation:** given a random variable $Z$ with distribution $Q$, and a real-valued function $h$,

$$\mathbb{E}[h(Z)] = \text{expected value of } h(Z) = \sum_z \Pr[Z = z] \cdot h(z).$$

[ Note: $h(Z)$ is a real-valued random variable over the same sample space as $Z$. ]
Let $Z$ be a random variable. What is the meaning of the following statement?

$$\mathbb{E}[\mathbb{1}\{Z = 1\}]$$

- **Indicator function:**
  $$\mathbb{1}\{A\} = \begin{cases} 1 & \text{if } A \text{ is true;} \\ 0 & \text{if } A \text{ is false.} \end{cases}$$

- **Expectation:** given a random variable $Z$ with distribution $Q$, and a real-valued function $h$,
  $$\mathbb{E}[h(Z)] = \text{expected value of } h(Z) = \sum_z \Pr[Z = z] \cdot h(z).$$

  [Note: $h(Z)$ is a real-valued random variable over the same sample space as $Z$.]

- **Therefore:**
  $$\mathbb{E}[\mathbb{1}\{Z = 1\}] = \Pr[Z = 1].$$
Suppose $A$ and $B$ are random variables.

- **Question**: What kind of object is $C := \mathbb{E}[A | B]$?
  
  (A real vector, a real number, etc.?)
Suppose $A$ and $B$ are random variables.

- **Question**: What kind of object is $C := \mathbb{E}[A | B]$?  
  (A real vector, a real number, etc.?)

- **Answer**: A random variable!
Suppose $A$ and $B$ are random variables.

- **Question:** What kind of object is $C := \mathbb{E}[A \mid B]$? (A real vector, a real number, etc.?)
- **Answer:** A random variable!

- $h(b) := \mathbb{E}[A \mid B = b]$ is a deterministic function of $b$. 

**Distribution of $C$ is given by:**

$$\Pr[C = h(b)] = \Pr[B = b].$$
Suppose $A$ and $B$ are random variables.

- **Question**: What kind of object is $C := \mathbb{E}[A \mid B]$?
  (A real vector, a real number, etc.?)

- **Answer**: A random variable!

  - $h(b) := \mathbb{E}[A \mid B = b]$ is a deterministic function of $b$.
  - Distribution of $C$ is given by: $\Pr[C = h(b)] = \Pr[B = b]$. 
Suppose $A$ and $B$ are random variables.

- **Question**: What kind of object is $C := \mathbb{E}[A \mid B]$? 
  (A real vector, a real number, etc.)

- **Answer**: A random variable!

- $h(b) := \mathbb{E}[A \mid B = b]$ is a deterministic function of $b$.

- Distribution of $C$ is given by: $\Pr[C = h(b)] = \Pr[B = b]$.

- **Question**: What is the expectation of $C = \mathbb{E}[A \mid B]$?
Suppose $A$ and $B$ are random variables.

- **Question**: What kind of object is $C := \mathbb{E}[A \mid B]$?
  
  (A real vector, a real number, etc.?)

- **Answer**: A random variable!

  - $h(b) := \mathbb{E}[A \mid B = b]$ is a deterministic function of $b$.
  
  - Distribution of $C$ is given by: $\Pr[C = h(b)] = \Pr[B = b]$.

- **Question**: What is the expectation of $C = \mathbb{E}[A \mid B]$?

- **Answer**:

  $$
  \mathbb{E}[C] = \sum_b \Pr[C = h(b)] \cdot h(b)
  $$

  $$
  = \sum_b \Pr[B = b] \cdot \mathbb{E}[A \mid B = b]
  $$

  $$
  = \mathbb{E}[A].
  $$
What is the optimal classifier?

Suppose $(X, Y) \sim P$. 
What is the optimal classifier?

Suppose $(X, Y) \sim P$.

For any classifier $f : \mathcal{X} \to \mathcal{Y}$, its prediction error is

$$\Pr[f(X) \neq Y] = \mathbb{E}[\mathbf{1}\{f(X) \neq Y\}]$$
What is the optimal classifier?

Suppose \((X, Y) \sim P\).

For any classifier \(f : \mathcal{X} \rightarrow \mathcal{Y}\), its prediction error is

\[
\Pr[f(X) \neq Y] = \mathbb{E}[\mathbf{1}\{f(X) \neq Y\}] = \mathbb{E} \left[ \mathbb{E} \left[ \mathbf{1}\{f(X) \neq Y\} \bigg| X \right] \right]. \quad (\dagger)
\]
What is the optimal classifier?

Suppose \((X, Y) \sim P\).

For any classifier \(f: \mathcal{X} \rightarrow \mathcal{Y}\), its prediction error is

\[
\Pr[f(X) \neq Y] = \mathbb{E}[1\{f(X) \neq Y\}] = \mathbb{E} \left[ \mathbb{E} \left[ 1\{f(X) \neq Y\} \mid X \right] \right]. \quad (\dagger)
\]

For each \(x \in \mathcal{X}\),

\[
\mathbb{E} \left[ 1\{f(X) \neq Y\} \mid X = x \right] = \sum_{y \in \mathcal{Y}} \Pr[Y = y \mid X = x] \cdot 1\{f(x) \neq y\}, \quad (\dagger)
\]
**What is the optimal classifier?**

Suppose \((X, Y) \sim P\).

For any classifier \(f : \mathcal{X} \to \mathcal{Y}\), its prediction error is

\[
\Pr[f(X) \neq Y] = \mathbb{E}[\mathbb{1}\{f(X) \neq Y\}] = \mathbb{E}_{X \sim P} \left[ \mathbb{E}_{Y \sim P \mid X} \mathbb{1}\{f(X) \neq Y\} \right]. \quad (\dagger)
\]

For each \(x \in \mathcal{X}\),

\[
\mathbb{E}\left[ \mathbb{1}\{f(X) \neq Y\} \mid X = x \right] = \sum_{y \in \mathcal{Y}} \Pr\left[ Y = y \mid X = x \right] \cdot \mathbb{1}\{f(x) \neq y\}, \quad (\ddagger)
\]

The above quantity (\(\ddagger\)) is minimized (for this \(x \in \mathcal{X}\)) when

\[
f(x) = \arg \max_{y \in \mathcal{Y}} \Pr\left[ Y = y \mid X = x \right]. \quad (\star)
\]
What is the optimal classifier?

Suppose \((X, Y) \sim P\).

For any classifier \(f : \mathcal{X} \to \mathcal{Y}\), its prediction error is

\[
\Pr[f(X) \neq Y] = \mathbb{E}[\mathbb{1}\{f(X) \neq Y\}] = \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}\{f(X) \neq Y\} \mid X\right] \mid X\right].
\]  

(†)

For each \(x \in \mathcal{X}\),

\[
\mathbb{E}\left[\mathbb{1}\{f(X) \neq Y\} \mid X = x\right] = \sum_{y \in \mathcal{Y}} \Pr[Y = y \mid X = x] \cdot \mathbb{1}\{f(x) \neq y\},
\]  

(‡)

The above quantity (‡) is minimized (for this \(x \in \mathcal{X}\)) when

\[
f(x) = \arg\max_{y \in \mathcal{Y}} \Pr[Y = y \mid X = x].
\]  

(⋆)

The classifier \(f\) with property (⋆) for all \(x \in \mathcal{X}\) is called the Bayes classifier, and it has the smallest prediction error (†) among all classifiers.
The Bayes classifier

\[ f^*(x) := \arg \max_{y \in \mathcal{Y}} \Pr[Y = y \mid X = x] \]

divides up the input space \( \mathcal{X} \) into different regions by how it predicts; the boundaries between these regions are called the decision boundaries.
The Bayes classifier

\[ f^*(x) := \arg \max_{y \in \mathcal{Y}} \Pr[Y = y \mid X = x] \]

divides up the input space \( \mathcal{X} \) into different regions by how it predicts; the boundaries between these regions are called the decision boundaries.

**Question**: What can these decision boundaries look like?
By Bayes' rule:

\[ \Pr[Y = y \mid X = x] = \frac{\Pr[Y = y] \cdot \Pr[X = x \mid Y = y]}{\Pr[X = x]} . \]
Structure of the Bayes classifier

By Bayes’ rule:

\[
\Pr[Y = y \mid X = x] = \frac{\Pr[Y = y] \cdot \Pr[X = x \mid Y = y]}{\Pr[X = x]}. 
\]

Since \(\Pr[X = x]\) does not depend on \(y\), the Bayes classifier is

\[
f^*(x) = \arg \max_{y \in \mathcal{Y}} \Pr[Y = y] \cdot \Pr[X = x \mid Y = y].
\]
**Structure of the Bayes classifier**

By Bayes’ rule:

$$\Pr[Y = y | X = x] = \frac{\Pr[Y = y] \cdot \Pr[X = x | Y = y]}{\Pr[X = x]}.$$ 

Since $\Pr[X = x]$ does not depend on $y$, the Bayes classifier is

$$f^*(x) = \arg \max_{y \in \mathcal{Y}} \Pr[Y = y] \cdot \Pr[X = x | Y = y].$$

- $\Pr[Y = \cdot]$ (i.e., the marginal distribution of $Y$) is called the **class prior**.
- $\Pr[X = \cdot | Y = y]$ is called the **class conditional distribution** of $X$ (for class $y$).
By Bayes’ rule:

$$\Pr[Y = y \mid X = x] = \frac{\Pr[Y = y] \cdot \Pr[X = x \mid Y = y]}{\Pr[X = x]}.$$ 

Since $$\Pr[X = x]$$ does not depend on $$y$$, the Bayes classifier is

$$f^*(x) = \arg \max_{y \in Y} \Pr[Y = y] \cdot \Pr[X = x \mid Y = y].$$

- $$\Pr[Y = \cdot]$$ (i.e., the marginal distribution of $$Y$$) is called the class prior.
- $$\Pr[X = \cdot \mid Y = y]$$ is called the class conditional distribution of $$X$$ (for class $$y$$).

If $$X$$ has a probability density (rather than a probability mass function), replace $$\Pr[X = \cdot \mid Y = y]$$ with class conditional density $$p_y(\cdot)$$. 
Example: Gaussian class conditional densities

Suppose $\mathcal{X} = \mathbb{R}$, $\mathcal{Y} = \{0, 1\}$, and the distribution $P$ of $(X, Y)$ is as follows.

- **Class prior:**
  $$\Pr[Y = y] = \pi_y, \quad y \in \{0, 1\}$$
  for some real numbers $\pi_0, \pi_1 \in [0, 1]$ satisfying $\pi_0 + \pi_1 = 1$. 

- **Class conditional density** for class $y \in \{0, 1\}$:
  $$p_y(x) = \frac{1}{\sqrt{2\pi\sigma^2_y}} \exp\left(-\frac{(x - \mu_y)^2}{2\sigma^2_y}\right)$$
  for some $\mu_y \in \mathbb{R}$ and $\sigma^2_y > 0$ (i.e., $N(\mu_y, \sigma^2_y)$).

- **Bayes classifier**:
  $$f^*(x) = \arg \max_{y \in \{0, 1\}} \Pr[Y = y | X = x] = \begin{cases} 
    1 & \text{if } \pi_1 \sigma_1 \exp\left(-\frac{(x - \mu_1)^2}{2\sigma^2_1}\right) > \pi_0 \sigma_0 \exp\left(-\frac{(x - \mu_0)^2}{2\sigma^2_0}\right); \\
    0 & \text{otherwise}
  \end{cases}$$
Example: Gaussian class conditional densities

Suppose $\mathcal{X} = \mathbb{R}$, $\mathcal{Y} = \{0, 1\}$, and the distribution $P$ of $(X, Y)$ is as follows.

- **Class prior:**
  
  $$\Pr[Y = y] = \pi_y, \quad y \in \{0, 1\}$$

  for some real numbers $\pi_0, \pi_1 \in [0, 1]$ satisfying $\pi_0 + \pi_1 = 1$.

- **Class conditional density** for class $y \in \{0, 1\}$:
  
  $$p_y(x) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x - \mu_y)^2}{2\sigma_y^2}\right)$$

  for some $\mu_y \in \mathbb{R}$ and $\sigma_y^2 > 0$ (i.e., $N(\mu_y, \sigma_y^2)$).
Example: Gaussian class conditional densities

Suppose $\mathcal{X} = \mathbb{R}$, $\mathcal{Y} = \{0, 1\}$, and the distribution $P$ of $(X, Y)$ is as follows.

- **Class prior:**
  \[
  \Pr[Y = y] = \pi_y, \quad y \in \{0, 1\}
  \]
  for some real numbers $\pi_0, \pi_1 \in [0, 1]$ satisfying $\pi_0 + \pi_1 = 1$.

- **Class conditional density** for class $y \in \{0, 1\}$:
  \[
  p_y(x) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x - \mu_y)^2}{2\sigma_y^2}\right)
  \]
  for some $\mu_y \in \mathbb{R}$ and $\sigma_y^2 > 0$ (i.e., $N(\mu_y, \sigma_y^2)$).

- **Bayes classifier:**
  \[
  f^*(x) = \arg \max_{y \in \{0, 1\}} \Pr[Y = y \mid X = x]
  \]
  \[
  = \begin{cases} 
    1 & \text{if } \frac{\pi_1}{\sigma_1} \exp\left(-\frac{(x - \mu_1)^2}{2\sigma_1^2}\right) > \frac{\pi_0}{\sigma_0} \exp\left(-\frac{(x - \mu_0)^2}{2\sigma_0^2}\right) \\
    0 & \text{otherwise.}
  \end{cases}
  \]
Example: Gaussian class conditional densities

\[ \pi_0 = 1/2 \quad \mu_0 = 0 \quad \sigma_0 = 1 \]

\[ \pi_1 = 1/2 \quad \mu_1 = 1 \quad \sigma_1 = 1 \]

1/2 of \( x \)'s from \( N(0, 1) \) (w/ \( y = 0 \))

1/2 of \( x \)'s from \( N(1, 1) \) (w/ \( y = 1 \))
Example: Gaussian class conditional densities

\[ \pi_0 = 1/2 \]
\[ \mu_0 = 0 \]
\[ \sigma_0 = 1 \]

\[ \pi_1 = 1/2 \]
\[ \mu_1 = 1 \]
\[ \sigma_1 = 1 \]

1/2 of \( x \)'s from \( N(0, 1) \) (w/ \( y = 0 \))
1/2 of \( x \)'s from \( N(1, 1) \) (w/ \( y = 1 \))

Bayes classifier:

\[ f^*(x) = \begin{cases} 
1 & \text{if } x > 1/2; \\
0 & \text{otherwise.}
\end{cases} \]
**Example: Gaussian class conditional densities**

\[ \pi_0 = \frac{1}{2}, \quad \mu_0 = 0, \quad \sigma_0 = 1 \]

\[ \pi_1 = \frac{1}{2}, \quad \mu_1 = 1, \quad \sigma_1 = \frac{1}{2} \]

1/2 of \( x \)'s from \( N(0, 1) \) (w/ \( y = 0 \))

1/2 of \( x \)'s from \( N(1, \frac{1}{2}^2) \) (w/ \( y = 1 \))
**Example: Gaussian class conditional densities**

\[
\begin{align*}
\pi_0 &= \frac{1}{2} \\
\mu_0 &= 0 \\
\sigma_0 &= 1 \\
\pi_1 &= \frac{1}{2} \\
\mu_1 &= 1 \\
\sigma_1 &= \frac{1}{2}
\end{align*}
\]

1/2 of x’s from \( N(0, 1) \) (w/ \( y = 0 \))
1/2 of x’s from \( N(1, 1/2^2) \) (w/ \( y = 1 \))

**Bayes classifier:**

\[
f^*(x) = \begin{cases} 
1 & \text{if } x \in [0.38, 2.29]; \\
0 & \text{otherwise.}
\end{cases}
\]
Example: multivariate Gaussians

$\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{0, 1\}$, class conditional densities are Gaussians in $\mathbb{R}^d$ ($d = 2$).

$\Sigma_0 = \Sigma_1$
Bayes classifier:
linear separator

$p(x|\omega_1)$ $p(x|\omega_2)$

$\Sigma_0 \neq \Sigma_1$
Bayes classifier:
quadratic separator
In general, Bayes classifier may be rather complicated!
CLASSIFIERS VIA GENERATIVE MODELS
Plug-in classifiers

Bayes classifier

- Bayes classifier has smallest prediction error among all possible classifiers.
Bayes classifier

- Bayes classifier has smallest prediction error among all possible classifiers.
- But we can’t construct the Bayes classifier without knowing \( \Pr[Y = y|X = x] \) for all \((x, y) \in \mathcal{X} \times \mathcal{Y}\)!

All we have are labeled examples drawn from the distribution.
Bayes classifier

- Bayes classifier has smallest prediction error among all possible classifiers.
- But we can’t construct the Bayes classifier without knowing $\Pr[Y = y | X = x]$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$!

All we have are labeled examples drawn from the distribution.

Plug-in classifiers

Using labeled examples, form an approximation to $\Pr[Y = y | X = x]$, then “plug-in” to the formula for Bayes classifier.
Plug-in classifiers

Bayes classifier

- Bayes classifier has smallest prediction error among all possible classifiers.
- But we can’t construct the Bayes classifier without knowing \( \Pr[Y = y|X = x] \) for all \( (x, y) \in \mathcal{X} \times \mathcal{Y} \)!

All we have are labeled examples drawn from the distribution.

Plug-in classifiers

Using labeled examples, form an approximation to \( \Pr[Y = y|X = x] \), then “plug-in” to the formula for Bayes classifier.

We’ll use “generative” statistical models to estimate \( P \), and then form approximation to \( \Pr[Y = y|X = x] \).
Plug-in classifiers using "generative" statistical models:

1. Use training data (labeled examples) to obtain approximations for each component in Bayes classifier formula:
   \[ f^\star(x) = \arg \max_{y \in Y} \Pr[Y = y] \cdot \Pr[X = x | Y = y] \]
   (i.e., class priors and class conditional distributions).

2. Plug-in approximations to formula to form classifier \( \hat{f} \).

   - Estimating class priors is easy: parameterized by \( \pi_y : y \in Y \) where \( \pi_y \geq 0 \) for all \( y \in Y \) and \( \sum_{y \in Y} \pi_y = 1 \). MLE is straightforward.

   - Estimating class conditional distributions hard in general. Usually just use simple parametric models.
Plug-in classifiers using “generative” statistical models:

1. Use training data (labeled examples) to obtain approximations for each component in Bayes classifier formula:

\[ f^*(x) = \arg \max_{y \in \mathcal{Y}} \Pr[Y = y] \cdot \Pr[X = x \mid Y = y] \]

(i.e., class priors and class conditional distributions).
Plug-in classifiers using “generative” statistical models:

1. Use training data (labeled examples) to obtain approximations for each component in Bayes classifier formula:

\[ f^*(x) = \arg \max_{y \in Y} \Pr[Y = y] \cdot \Pr[X = x | Y = y] \]

(i.e., class priors and class conditional distributions).

2. Plug-in approximations to formula to form classifier \( \hat{f} \).
Plug-in classifiers using “generative” statistical models:

1. Use **training data** (labeled examples) to obtain approximations for each component in Bayes classifier formula:

   \[ f^*(x) = \arg \max_{y \in \mathcal{Y}} \Pr[Y = y] \cdot \Pr[X = x \mid Y = y] \]

   (i.e., class priors and class conditional distributions).

2. **Plug-in** approximations to formula to form classifier \( \hat{f} \).

- **Estimating class priors is easy**: parameterized by \( (\pi_y : y \in \mathcal{Y}) \) where \( \pi_y \geq 0 \) for all \( y \in \mathcal{Y} \) and \( \sum_{y \in \mathcal{Y}} \pi_y = 1 \). MLE is straightforward.
Plug-in classifiers using “generative” statistical models:

1. Use **training data** (labeled examples) to obtain approximations for each component in Bayes classifier formula:

\[
f^*(x) = \arg \max_{y \in \mathcal{Y}} \Pr[Y = y] \cdot \Pr[X = x \mid Y = y]
\]

(i.e., **class priors** and **class conditional distributions**).

2. **Plug-in** approximations to formula to form classifier \( \hat{f} \).

- **Estimating class priors is easy**: parameterized by \((\pi_y : y \in \mathcal{Y})\) where \(\pi_y \geq 0\) for all \(y \in \mathcal{Y}\) and \(\sum_{y \in \mathcal{Y}} \pi_y = 1\). MLE is straightforward.
- **Estimating class conditional distributions** hard in general.
Plug-in classifiers using “generative” statistical models:

1. Use training data (labeled examples) to obtain approximations for each component in Bayes classifier formula:

   \[ f^*(x) = \arg \max_{y \in \mathcal{Y}} \Pr[Y = y] \cdot \Pr[X = x | Y = y] \]

   (i.e., class priors and class conditional distributions).

2. Plug-in approximations to formula to form classifier \( \hat{f} \).

   - Estimating class priors is easy: parameterized by \((\pi_y : y \in \mathcal{Y})\) where \(\pi_y \geq 0\) for all \(y \in \mathcal{Y}\) and \(\sum_{y \in \mathcal{Y}} \pi_y = 1\). MLE is straightforward.

   - Estimating class conditional distributions hard in general.

   Usually just use simple parametric models.
**Example: Gaussian class conditional densities**

\( \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{1, 2, \ldots, K\} \)

- **Class priors**: MLE estimate of \( \pi_y \) is

\[
\hat{\pi}_y := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{y_i = y\}.
\]

**Class conditional density** \( \mathcal{N}(\mu_y, \Sigma_y) \): MLE estimate of \( (\mu_y, \Sigma_y) \) is

\[
\hat{\mu}_y := \frac{1}{n\hat{\pi}_y} \sum_{i=1}^{n} \mathbb{1}\{y_i = y\} \mathbf{x}_i, \\
\hat{\Sigma}_y := \frac{1}{n\hat{\pi}_y} \sum_{i=1}^{n} \mathbb{1}\{y_i = y\} (\mathbf{x}_i - \hat{\mu}_y)(\mathbf{x}_i - \hat{\mu}_y)^\top.
\]

**Plug-in classifier**: \( \hat{f}(\mathbf{x}) = \arg \max_{y \in \mathcal{Y}} \hat{\pi}_y \det(\hat{\Sigma}_y)^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \hat{\mu}_y)(\mathbf{x} - \hat{\mu}_y)^\top \right) \).

**Caveat**: \( \hat{\Sigma}_y \) could be singular!
Example: Gaussian class conditional densities

\( \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{1, 2, \ldots, K\} \)

- **Class priors**: MLE estimate of \( \pi_y \) is

\[
\hat{\pi}_y := \frac{1}{n} \sum_{i=1}^{n} 1\{y_i = y\}.
\]

Class conditional density \( N(\mu_y, \Sigma_y) \): MLE estimate of \( (\mu_y, \Sigma_y) \) is

\[
\hat{\mu}_y := \frac{1}{n \hat{\pi}_y} \sum_{i=1}^{n} 1\{y_i = y\} \mathbf{x}_i,
\]

\[
\hat{\Sigma}_y := \frac{1}{n \hat{\pi}_y} \sum_{i=1}^{n} 1\{y_i = y\} (\mathbf{x}_i - \hat{\mu}_y) (\mathbf{x}_i - \hat{\mu}_y)^\top.
\]

- **Plug-in classifier**:

\[
\hat{f}(\mathbf{x}) = \arg \max_{y \in \mathcal{Y}} \frac{\hat{\pi}_y}{\det(\hat{\Sigma}_y)^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \hat{\mu}_y)^\top \hat{\Sigma}_y^{-1} (\mathbf{x} - \hat{\mu}_y)\right).
\]

Caveat: \( \hat{\Sigma}_y \) could be singular!
Example: Gaussian class conditional densities

\[ \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{1, 2, \ldots, K\} \]

▶ Class priors: MLE estimate of \( \pi_y \) is

\[
\hat{\pi}_y := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{y_i = y\}.
\]

Class conditional density \( \mathcal{N}(\mu_y, \Sigma_y) \): MLE estimate of \( (\mu_y, \Sigma_y) \) is

\[
\hat{\mu}_y := \frac{1}{n\hat{\pi}_y} \sum_{i=1}^{n} \mathbb{1}\{y_i = y\} x_i,
\]

\[
\hat{\Sigma}_y := \frac{1}{n\hat{\pi}_y} \sum_{i=1}^{n} \mathbb{1}\{y_i = y\} (x_i - \hat{\mu}_y)(x_i - \hat{\mu}_y)^\top.
\]

▶ Plug-in classifier:

\[
\hat{f}(x) = \arg \max_{y \in \mathcal{Y}} \frac{\hat{\pi}_y}{\det(\hat{\Sigma}_y)^{1/2}} \exp\left(-\frac{1}{2} (x - \hat{\mu}_y)^\top \hat{\Sigma}_y^{-1} (x - \hat{\mu}_y)\right).
\]

Caveat: \( \hat{\Sigma}_y \) could be singular!
Advantages:

1. Very simple recipe, just need some domain expertise to pick the class conditional distributions.

2. Surprising effective in many applications.
Advantages:

1. Very simple recipe, just need some domain expertise to pick the class conditional distributions.
2. Surprising effective in many applications.

Disadvantages:

1. Hard to justify unless class conditional distributions are close to the truth.
2. Effort modeling $P$ away from decision boundary between classes is not necessary for good classification.
Classifiers via generative models: recap

Advantages:

1. Very simple recipe, just need some domain expertise to pick the class conditional distributions.
2. Surprising effective in many applications.

Disadvantages:

1. Hard to justify unless class conditional distributions are close to the truth.
2. Effort modeling $P$ away from decision boundary between classes is not necessary for good classification.

Next time: methods for modeling decision boundary directly.
EVALUATING CLASSIFIERS
Let $S := ((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ be the training data used to construct a classifier $\hat{f}$ (say, via the plug-in method).
Let $S := ((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ be the training data used to construct a classifier $\hat{f}$ (say, via the plug-in method).

**Question:** Can we tell if $\hat{f}$ is any good?

- **True error:**

  \[ \text{err}(\hat{f}) := \Pr[\hat{f}(X) \neq Y] \]

  where $(X, Y) \sim P$.

- **Training error:**

  \[ \text{err}(\hat{f}, S) := \frac{1}{|S|} \sum_{(x, y) \in S} 1\{\hat{f}(x) \neq y\} \]

  This often under-estimates the true error $\text{err}(\hat{f})$. 
Let $S := ((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ be the training data used to construct a classifier $\hat{f}$ (say, via the plug-in method).

**Question:** Can we tell if $\hat{f}$ is any good?

- **True error:**
  \[
  \text{err}(\hat{f}) := \Pr[\hat{f}(X) \neq Y]
  \]
  where $(X, Y) \sim P$.

- **Training error:**
  \[
  \text{err}(\hat{f}, S) := \frac{1}{|S|} \sum_{(x, y) \in S} 1\{\hat{f}(x) \neq y\}.
  \]
Let \( S := ((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)) \) be the training data used to construct a classifier \( \hat{f} \) (say, via the plug-in method).

**Question:** Can we tell if \( \hat{f} \) is any good?

- **True error:**

  \[
  \text{err}(\hat{f}) := \Pr[\hat{f}(X) \neq Y]
  \]

  where \((X, Y) \sim P\).

  Unfortunately, we don’t know \( P \) so this can’t be computed.

- **Training error:**

  \[
  \text{err}(\hat{f}, S) := \frac{1}{|S|} \sum_{(x,y) \in S} \mathbb{1}\{\hat{f}(x) \neq y\}.
  \]
Let \( S := ((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)) \) be the training data used to construct a classifier \( \hat{f} \) (say, via the plug-in method).

**Question:** Can we tell if \( \hat{f} \) is any good?

- **True error:**
  \[
  \text{err}(\hat{f}) := \Pr[\hat{f}(X) \neq Y]
  \]
  where \((X, Y) \sim P\).

  Unfortunately, we don’t know \( P \) so this can’t be computed.

- **Training error:**
  \[
  \text{err}(\hat{f}, S) := \frac{1}{|S|} \sum_{(x, y) \in S} 1\{\hat{f}(x) \neq y\}.
  \]

  This often under-estimates the true error \( \text{err}(\hat{f}) \).
Test error

General methodology
Given your pile of labeled examples, \((\text{randomly})\) split into two disjoint groups:

1. **training data** \(S\);
2. **test data** \(T\).
Test error

General methodology
Given your pile of labeled examples, (randomly) split into two disjoint groups:

1. training data $S$;
2. test data $T$.

Assuming you only use $S$ to build the classifier $\hat{f}$, the test error $\text{err}(\hat{f}, T) := \frac{1}{|T|} \sum_{(x,y) \in T} 1\{\hat{f}(x) \neq y\}$

is a good estimate of the true error of $\hat{f}$. 
**Test error**

**General methodology**
Given your pile of labeled examples, (randomly) split into two disjoint groups:

1. **training data** $S$;
2. **test data** $T$.

Assuming you only use $S$ to build the classifier $\hat{f}$, the **test error**

$$\text{err}(\hat{f}, T) := \frac{1}{|T|} \sum_{(x,y) \in T} \mathbb{1}\{\hat{f}(x) \neq y\}$$

is a good estimate of the true error of $\hat{f}$.

**Assuming $T$ is an i.i.d. sample from $P$:**

- The test error is an **unbiased estimate of** $\text{err}(\hat{f})$: $\mathbb{E}[\text{err}(\hat{f}, T) \mid S] = \text{err}(\hat{f})$.
  [ Expectation is over random draw of $T$. ]

- The standard deviation of $\text{err}(\hat{f}, T)$ is $\sqrt{\frac{\text{err}(\hat{f})(1 - \text{err}(\hat{f}))}{|T|}}$. 
Example: OCR

- **Task**: classify images of handwritten digits
Task: classify images of handwritten digits

Data: 30000 grayscale 28 × 28 images (treated as vectors in \( \mathbb{R}^{784} \)), with labels indicating the digit they represent.

An example from each class:

```
0 1 2 3 4 5 6 7 8 9
```
Example: OCR

- **Task:** classify images of handwritten digits
- **Data:** 30000 grayscale $28 \times 28$ images (treated as vectors in $\mathbb{R}^{784}$), with labels indicating the digit they represent.

An example from each class:

```
0 1 2 3 4 5 6 7 8 9
```

- Split into training data $S$ and test data $T$ using random $(2/3, 1/3)$-split.
Example: OCR

- **Task:** classify images of handwritten digits
- **Data:** 30000 grayscale $28 \times 28$ images (treated as vectors in $\mathbb{R}^{784}$), with labels indicating the digit they represent.
  
  An example from each class:

  0 1 2 3 4 5 6 7 8 9

- Split into training data $S$ and test data $T$ using random $(2/3, 1/3)$-split.
- Using $S$, trained plug-in classifier $\hat{f}$ with Gaussian class conditional densities (plus one other trick).

  MLE $\hat{\mu}_y$ for $y \in \mathcal{Y}$:
Example: OCR

- **Task**: classify images of handwritten digits
- **Data**: 30000 grayscale $28 \times 28$ images (treated as vectors in $\mathbb{R}^{784}$), with labels indicating the digit they represent.

An example from each class:

![Example Images](image)

- Split into training data $S$ and test data $T$ using random $(2/3, 1/3)$-split.
- Using $S$, trained plug-in classifier $\hat{f}$ with Gaussian class conditional densities (plus one other trick).

MLE $\hat{\mu}_y$ for $y \in \mathcal{Y}$:

![MLE Images](image)

- **Training error**: $\text{err}(\hat{f}, S) = 0.0346$
- **Test error**: $\text{err}(\hat{f}, T) = 0.0415$
Example: OCR

- **Task**: classify images of handwritten digits

- **Data**: 30000 grayscale $28 \times 28$ images (treated as vectors in $\mathbb{R}^{784}$), with labels indicating the digit they represent.

  An example from each class:

  0 1 2 3 4 5 6 7 8 9

- Split into training data $S$ and test data $T$ using random $(2/3, 1/3)$-split.

- Using $S$, trained plug-in classifier $\hat{f}$ with Gaussian class conditional densities (plus one other trick).

  MLE $\hat{\mu}_y$ for $y \in \mathcal{Y}$:

  0 1 2 3 4 5 6 7 8 9

- **Training error**: $err(\hat{f}, S) = 0.0346$

- **Test error**: $err(\hat{f}, T) = 0.0415$

- **True error?** Unknown, but perhaps $0.039 \leq err(\hat{f}) \leq 0.044$ or so.