Agenda

- Supervised Learning
  - Regression
    - Least Mean Square
  - Classification
    - Probability
Regression

- Given 15 fishes: weight and prices
- Objective: Predict the price of a fish

<table>
<thead>
<tr>
<th>Weight (kg)</th>
<th>Price ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>20</td>
</tr>
<tr>
<td>2.6</td>
<td>31</td>
</tr>
<tr>
<td>1.2</td>
<td>16.5</td>
</tr>
<tr>
<td>0.7</td>
<td>10</td>
</tr>
</tbody>
</table>

The $i$th training sample: $(x^{(i)}, y^{(i)})$

- $x^{(i)} = [x_1^{(i)}, x_2^{(i)}, ..., x_d^{(i)}]^T \in X$: Feature vector
  - $x_j^{(i)}$: the jth feature
- $X$: the input space
- $y^{(i)} \in Y$: Label / Target
  - $Y$: the output space

Training set: $\{(x^{(i)}, y^{(i)}) | i = 1...n\}$

- $n$ is the number of training samples
Regression

- Train a function $h_\theta(x)$ to predict $y$
  - $\theta$ is the parameter vectors (e.g. weight)
  - $h_\theta: X \rightarrow Y$, mapping from $X$ to $Y$
  - $h_\theta$ is called a predictor or hypothesis

- Objective: Build a “good” $h_\theta$
  - What does “good” mean?

Objective Function

- Objective: the predicted value on a training sample closer to the real one
  - Smaller difference between $h_\theta(x^{(i)})$ and $y^{(i)}$

- Cost function (objective function)

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2$$

- Error is a distance measure
- Square avoids the cancellation of positive and negative error
LMS Algorithm

- Least Mean Squares (LMS) aims to minimize $J(\theta)$ by adjusting $\theta$

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2$$

- $\theta$ can be obtained by
  - Pseudoinverse Method ($h_\theta$ is linear)
  - Gradient Descent ($h_\theta$ is differentiable)

LMS Algorithm

Pseudoinverse Method

- When $h_\theta$ is linear, Pseudoinverse can be applied
- A sample and its outputs can be treated as an equation
  - i.e. For the $i$th samples: $\sum_{k=1}^{d} \theta_k x_k^{(i)} = y_i$

- $n$ samples can be represented in matrix notation:

$$
\begin{pmatrix}
x_1^{(1)} & x_2^{(1)} & \cdots & x_d^{(1)} \\
x_1^{(2)} & x_2^{(2)} & \cdots & x_d^{(2)} \\
\vdots & \vdots & \ddots & \vdots \\
x_1^{(n)} & x_2^{(n)} & \cdots & x_d^{(n)} \\
\end{pmatrix}
\begin{pmatrix}
\theta_1 \\
\theta_2 \\
\vdots \\
\theta_d \\
\end{pmatrix}
= 
\begin{pmatrix}
y_1 \\
y_2 \\
\vdots \\
y_n \\
\end{pmatrix}
$$

or

$$X^T \theta = y$$
LMS Algorithm

Pseudoinverse Method

- As $X^T \theta = y$, $\theta$ can be solved by calculating the inverse if $X$ is nonsingular,
  $$\theta = (X^T)^{-1} y$$

- However, $X^T$ is usually rectangular, $(X^T)^{-1}$ is undefined
  - More samples than features
  - More rows than columns
  - More equations than variables
    - $\theta$ is over-determined
    - No exact solution exists

Criterion Function ($J$)

Sum-of-squared-error Function

- $X^T \theta = y$

- Square Error:
  $$J(\theta) = (X^T \theta - y)^2$$
  $$\frac{\partial}{\partial \theta_j} J(\theta) = 2X(X^T \theta - y)$$

- When $\frac{\partial}{\partial \theta_j} J(\theta) = 0$
  $$XX^T \theta = Xy$$
  $$\theta = (XX^T)^{-1} Xy$$

- $XX^T$ is not always nonsingular
- It should be defined more generally by
  $$\theta = \lim_{\varepsilon \to 0} (XX^T + \varepsilon I)^{-1} Xy$$
**LMS Algorithm**

**Gradient Descent**

- $h_\theta$ is differentiable, gradient descent can be used to minimize $J(\theta)$

\[ J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2 \]

- Influence on $J(\theta)$ by changing the parameter ($\theta$) slightly

\[ \theta^{(t+1)} = \theta^{(t)} - \alpha \frac{\partial}{\partial \theta} J(\theta^{(t)}) \]

  - $\alpha$: the learning rate
  - $\theta^{(t)}$: the parameter at the $t$ time

---

**Algorithm**

- Start with an arbitrarily chosen weight $\theta^{(1)}$
- Let $t = 0$
- Loop
  - $t = t + 1$
  - Compute gradient vector $\frac{\partial J(\theta^{(t)})}{\partial \theta}$
  - Next value $\theta^{(t+1)}$ determined by moving some distance from $\theta^{(t)}$ in the direction of the steepest descent

\[ \theta^{(t+1)} = \theta^{(t)} - \alpha \frac{\partial}{\partial \theta} J(\theta^{(t)}) \]

  - i.e., along the negative of the gradient
- Until Finish Training
LMS Algorithm

Gradient Descent

- Recall, \( \theta = [\theta_1, \theta_2, \ldots, \theta_m] \)
- Updated Rule for the jth parameter

\[
\theta_j^{(t+1)} = \theta_j^{(t)} - \alpha \frac{\partial}{\partial \theta_j} J(\theta_j^{(t)})
\]

- All parameters should be updated at the same time

How to calculate \( \frac{\partial}{\partial \theta_j} J(\theta) \)?

\[
J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2
\]

\[
\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2
\]

\[
= \frac{1}{2n} \sum_{i=1}^{n} \frac{\partial}{\partial \theta_j} (h_\theta(x^{(i)}) - y^{(i)})^2
\]

\[
= \frac{1}{2n} \sum_{i=1}^{n} 2(h_\theta(x^{(i)}) - y^{(i)}) \frac{\partial}{\partial \theta_j} (h_\theta(x^{(i)}) - y^{(i)})
\]

\[
= \frac{1}{2n} \sum_{i=1}^{n} 2(h_\theta(x^{(i)}) - y^{(i)}) \frac{\partial h_\theta(x^{(i)})}{\partial \theta_j}
\]
LMS Algorithm

Gradient Descent

- **Linear Function**
  \[ h_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d \]
  \[ = \sum_{i=1}^{d} \theta_i x_i \]

  \[
  \frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} 2(h_\theta(x^{(i)}) - y^{(i)}) \frac{\partial h_\theta(x^{(i)})}{\partial \theta_j} \\
  = \frac{1}{2n} \sum_{i=1}^{n} 2(h_\theta(x^{(i)}) - y^{(i)}) \frac{\partial}{\partial \theta_j} \left( \sum_{k=1}^{d} \theta_k x_k^{(i)} - y^{(i)} \right) \\
  = \frac{1}{n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}
  \]

**Batch Gradient Descent**

- Calculate the gradient for \( n \) samples, and update 1 time

\[
\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \\
\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}
\]

- Initialize \( \theta_j^{(1)} \), \( j = 1\ldots m \)
- \( t = 0 \)
- do
  - \( t = t + 1 \)
  - for \( j = 1\ldots m \)
    - \( \theta_j^{(t+1)} = \theta_j^{(t)} - \alpha \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \)
  - end for
- end do

All parameters are updated at the same time

\[
\boxed{\text{All samples, one update}}
\]
\[ \theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \]

\[ \frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \]

**Stochastic Gradient Descent (Incremental Gradient Descent)**
- Update by 1 sample each time
- Initialize \( \theta_j^{(1)} \), \( j = 1 \ldots m \)
- \( t = 0 \)
- do
  - \( t = t + 1 \)
  - \( i = \text{mod}(i, n) + 1 \)
  - for \( j = 1 \ldots m \)
    - \( \theta_j^{(t+1)} = \theta_j^{(t)} - \alpha (h_\theta^{(t)}(x^{(i)}) - y^{(i)}) x_j^{(i)} \)
  - end for
- until The range of \( i = 1 \ldots n \)

One samples, one update

All parameters are updated at the same time

17 Dr. Patrick Chan @ SCUT

**Batch Gradient Descent**
- 1 update: \( n \) samples
- Time complexity of learning a dataset once is lower (1 update)
- One update is costly when \( n \) is large
- Mathematic proof on convergence but takes time

**Stochastic Gradient Descent**
- 1 update: 1 sample
- Time complexity of learning a dataset once is higher (\( n \) updates)
- Make a little progress every time
- May never converge but close to the minimum faster usually (good enough in practical)
**Related Issues:**

- **Size of Learning Rate ($\eta$)**
  - Too small, convergence is needlessly slow
  - Too large, the correction process will overshoot and cannot even diverge

- **Sub-optimal Solution**
  - Trapped by local minimum

**Problems of Pseudoinverse:**

- Problem of singularity
- Working with large matrices
- Only error can be considered

**Problems of Gradient Descent**

- Long training time
- Local minimum
Objective: output a class based on the features of a sample

Prior Probability

Peter went to body check to see if he is ok
\[ y = \text{(ill, healthy)} \]

According to the previous records, the doctor concluded
- 85% of people was healthy
  \[ P(y = \text{healthy}) = 0.85 \]
- 15% of people was ill
  \[ P(y = \text{ill}) = 0.15 \]
- Therefore, Peter was healthy
  \[ P(y = \text{healthy}) > P(y = \text{ill}) \]

Should Peter be satisfied with this diagnosis?
- This decision is based on Prior Probability \( P(y) \)

<table>
<thead>
<tr>
<th>Person</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Ill</td>
</tr>
<tr>
<td>B</td>
<td>Healthy</td>
</tr>
<tr>
<td>C</td>
<td>Healthy</td>
</tr>
<tr>
<td>D</td>
<td>Ill</td>
</tr>
<tr>
<td>\vdots</td>
<td></td>
</tr>
</tbody>
</table>

Illustration: Hans Maller, mollers.dk
Physical condition of persons should be considered
- Quantify the characteristics (features), denoted by $x$
- E.g. red blood cell #, white blood cell #, temperature

Assume only white blood cell # is measured

<table>
<thead>
<tr>
<th>Person</th>
<th>White Blood Cell #</th>
<th>Status ($y$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>50</td>
<td>Ill</td>
</tr>
<tr>
<td>B</td>
<td>42</td>
<td>Healthy</td>
</tr>
<tr>
<td>C</td>
<td>39</td>
<td>Healthy</td>
</tr>
<tr>
<td>D</td>
<td>62</td>
<td>Ill</td>
</tr>
</tbody>
</table>

Assume the white blood cell # ($x$) of Peter is 2

A probability density function (pdf) of persons is considered

The Doctor said
- $p(x=2 \mid \text{ill}) = 0.67$
- $p(x=2 \mid \text{healthy}) = 0.05$
- Therefore, Peter is ill

Should we be satisfied?
- This decision is based on Likelihood $P(x \mid y)$
Classification

Posterior Probability

- Using Prior Probability (P(y)) or Likelihood (p(x|y)) is not suitable

- Posterior Probability is a better choice
  \[ P( y | x ) : \text{given } x, \text{the probability of } y \]

- Bayes Decision Rule (Bayes Classifier)
  - When \( P(y_1 | x) > P(y_2 | x) \), x is \( y_1 \)
  - When \( P(y_2 | x) > P(y_1 | x) \), x is \( y_2 \)
  - When \( P(y_1 | x) = P(y_2 | x) \), no decision

- How to calculate \( P( y | x ) \)?
  - It is difficult as \( x \) is usually a continuous value

Bayes Formula

- Bayes Formula
  \[
  P(y|x) = \frac{p(x|y)P(y)}{p(x)}
  \]
  
  - Likelihood and prior probability may be estimated by using a dataset:
    \[
    p(x=2 \mid \text{ill}) = 0.67 \quad p(x=2 \mid \text{healthy}) = 0.05 \\
    P(\text{ill}) = 0.15 \quad P(\text{healthy}) = 0.85
    \]

- How about evidence \( p(x) \)?
Classification

Bayes Decision Rule

- \( p(x) \) is difficult to obtain
  - Fortunately, it can be neglected since it will not affect the decision

- \( x \) is classified as \( y_1 \) if
  \[
p(y_1|x) > p(y_2|x)
  \]
  \[
  \frac{p(x|y_1)p(y_1)}{p(x)} > \frac{p(x|y_2)p(y_2)}{p(x)}
  \]
  \[
p(x|y_1)p(y_1) > p(x|y_2)p(y_2)
  \]

Recall:
- \( p(x=2 | \text{ill}) = 0.67 \quad p(x=2 | \text{healthy}) = 0.05 \)
- \( P(\text{ill}) = 0.15 \quad P(\text{healthy}) = 0.85 \)

According to Bayes Decision Rule
- Decide \( y_1 \) if \( P(y_1|x) > P(y_2|x) \)
- Decide \( y_2 \) if \( P(y_2|x) > P(y_1|x) \)

- \( P(\text{healthy} | x = 2) \propto p(x=2 | \text{healthy}) \times P(\text{healthy}) \)
  \[
  = 0.05 \times 0.85 = 0.0425
  \]

- \( P(\text{ill} | x = 2) \propto p(x=2 | \text{ill}) \times P(\text{ill}) \)
  \[
  = 0.67 \times 0.15 = 0.1005
  \]
  * Note that if \( p(x) \) is considered, then \( P(y_1|x) + P(y_2|x) = 1 \).

- \( 0.1005 > 0.0425 \), therefore, **Peter is ill**
Classification: Bayes Decision Rule

Maximum Likelihood Estimation

- When \( P(y_1) = P(y_2) \), the decision is based entirely on the likelihood ( \( p(x \mid y_j) \) )

\[
P(y \mid x) = \frac{p(x \mid y) P(y)}{p(x)}
\]

- Decide \( y_1 \) if \( p(x \mid y_1) > p(x \mid y_2) \)
- Decide \( y_2 \) if \( p(x \mid y_2) > p(x \mid y_1) \)

- MLE is a special case of Bayes Rule

Classification: Bayes Decision Rule

Decision Boundary

- Error is usually unavoidable
  - Samples with the same value may be ill or healthy
    - e.g. \( x = 3 \)
  - A classifier only can classify a sample to ONE class based on its value
    - A function

- But always can be minimized
  - Classify to the class with higher posterior probability
Classification: Bayes Decision Rule

Error Probability

- There are two possible errors:
  - Error Probability ($P(\text{error} \mid x)$) is
    - $P(y_1 \mid x)$ if $y_2$ is chosen
    - $P(y_2 \mid x)$ if $y_1$ is chosen

Recall the Bayes Decision Rule:
- if $P(y_1 \mid x) > P(y_2 \mid x)$, decide $y_1$
- Otherwise decide $y_2$

Error: $P(\text{error} \mid x) = \min \left[ P(y_1 \mid x), P(y_2 \mid x) \right]$
Classification: Bayes Decision Rule: Error Probability

Bayes & Added Error

\[ P(error) = P(x \in R_2, y_1) + P(x \in R_1, y_2) \]
\[ = P(x \in R_2 | y_1)P(y_1) + P(x \in R_1 | y_2)P(y_2) \]
\[ = \int_{R_2} p(x|y_1)P(y_1)dx + \int_{R_1} p(x|y_2)P(y_2)dx \]

Error

\[ = \text{Bayes Error} + \text{Added Error} \]

Cannot be reduced for this input space
Can be reduced by choosing better parameters

Extension to Multi-Class

- Extend to multi-class problem \((c\ classes)\)

\[ y = (y_1, y_2, \ldots, y_c) \]

- Bayes Decision Rule

  \[ x \text{ is } y_i \text{ if } P(y_i | x) \text{ is maximum for } i = 1...c \]

- Error for Bayes Decision Rule

\[ P(error \mid x) = 1 - \max[ P(y_1 | x), P(y_2 | x), \ldots, P(y_c | x) ] \]
A three-class example:

- Bayes Decision Rule
  - \( x \) is \( y_i \) if \( P(y_i | x) \) is max for \( i = 1 \ldots 3 \)

\[
P(y_1 | x) \text{ is max} \\
P(y_2 | x) \text{ is max} \\
P(y_3 | x) \text{ is max}
\]

Error of Bayes Decision Rule:

\[
P(error | x) = 1 - \max[P(y_1|x), P(y_2|x), P(y_3|x)]
\]

For example, in the green region, 
(x is classified as \( y_2 \) based on Bayes Rule)

\[
P(error | x) = P(y_1 | x) + P(y_3 | x)
= 1 - P(y_2 | x)
= 1 - \max_{i=1,2,3} P(y_i | x) \quad * \text{Must not be } P(y_2|x)
\]
**Classification: Bayes Decision Rule**

**Action Cost**

- Considering Posterior Probability only minimizes the error rate but **not the cost of different error**

- In some applications, the cost of taking each action when a sample belongs to different classes should be considered

- E.g.
  - Send a healthy person to a hospital
  - Send a sick person to home

Let $\lambda_{ij} = \lambda(a_i | y_j)$ be the cost of taking the action $a_i$ when the class is $y_j$

- $\lambda_{ii} > \lambda_{ij}$ for any $i$ and $j$

<table>
<thead>
<tr>
<th>Action</th>
<th>Class</th>
<th>Action</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send to Hospital ($a_1$)</td>
<td>ill ($y_1$)</td>
<td>Send a patient to hospital</td>
<td>$\lambda_{11} = \lambda(a_1</td>
</tr>
<tr>
<td>Send to home ($a_2$)</td>
<td>healthy ($y_2$)</td>
<td>Send a healthy person to hospital</td>
<td>$\lambda_{12} = \lambda(a_1</td>
</tr>
</tbody>
</table>
Action Cost

- **Expected loss** for an action $a_i$ risk on $x$:

$$R(a_i|x) = \sum_{j=1}^{c} \lambda(a_i|y_j)P(y_j|x)$$

  - Loss of taking action $a_i$
  - Probability of classifying as $y_j$ given $x$

- **Overall risk** $R$ (expected loss)

$$R = \int R(a(x)|x)p(x)dx$$

- For example, a two-class problem:

  $$R(a_1|x) = \lambda_{11}P(y_1|x) + \lambda_{12}P(y_2|x)$$
  $$R(a_2|x) = \lambda_{21}P(y_1|x) + \lambda_{22}P(y_2|x)$$

Minimum Risk Decision Rule

- **Select** $a_i$ with the minimum $R(a_i|x)$

  $$\min R(a_i|x) \text{ for } i = 1, \ldots, c$$

- For a two-class problem:

  - Decide $a_1$ if $R(a_1|x) < R(a_2|x)$
  - otherwise decide $a_2$
Example:

- \( \lambda_{11} = 1 \quad \lambda_{21} = 10 \)
- \( \lambda_{12} = 5 \quad \lambda_{22} = 3 \)

When \( P(y_1 | x) = 0.1 \) and \( P(y_2 | x) = 0.9 \):

- \( R(a_1 | x) = 1 \times 0.1 + 5 \times 0.9 = 4.6 \)
- \( R(a_2 | x) = 10 \times 0.1 + 3 \times 0.9 = 3.7 \)
- Action 2 \( (a_2) \) is selected

When \( P(y_1 | x) = 0.8 \) and \( P(y_2 | x) = 0.2 \):

- \( R(a_1 | x) = 1 \times 0.8 + 5 \times 0.2 = 1.8 \)
- \( R(a_2 | x) = 10 \times 0.8 + 3 \times 0.2 = 8.6 \)
- Action 1 \( (a_1) \) is selected

\[ R(a_i | x) = \sum_{j=1}^{c} \lambda(a_i | y_j) P(y_j | x) \]

Minimize the risk \( R(a_i | x) \) requires maximizing the posterior probability \( P(y_i | x) \)

For example, a 2-class problem, if we take action 1,

\[
R(a_1 | x) = \lambda_{11} P(y_1 | x) + \lambda_{12} P(y_2 | x) \\
= \lambda_{11} P(y_1 | x) + \lambda_{12} (1 - P(y_1 | x)) \\
= (\lambda_{11} - \lambda_{12}) P(y_1 | x) + \lambda_{12}
\]

- \( \lambda_{11} \) and \( \lambda_{12} \) are given, and \( \lambda_{11} < \lambda_{12} \) \((\lambda_{11} - \lambda_{12})\) is negative.
- \( R(a_1 | x) \) is minimal if \( P(y_1 | x) \) is maximal (i.e. \( P(y_2 | x) \) is minimal)
Classification: Bayes Decision Rule: Action Cost
Minimum Risk Decision Rule

- **Special case: Zero-one loss function**
  - No loss if taking $a_i$ for $x_i$ in $y_i$
  - Otherwise Loss is 1

$$\lambda_{ij} = \begin{cases} 
0 & i = j \\
1 & i \neq j 
\end{cases}$$

$$R(a_i | x) = \sum_{j=1}^{c} \lambda(a_i | y_j) P(y_j | x)$$

$$= \sum_{j \neq i} P(y_j | x)$$

$$= 1 - P(y_i | x)$$

- Equivalent to Bayes Rule

- **Example: 2-class problem**
  - $a_1$ is chosen if $R(a_1 | x) < R(a_2 | x)$
  - Otherwise, $a_2$ is chosen

$$R(a_1 | x) < R(a_2 | x)$$

$$\lambda_{11}P(y_1 | x) + \lambda_{12}P(y_2 | x) < \lambda_{21}P(y_1 | x) + \lambda_{22}P(y_2 | x)$$

$$\lambda_{12} - \lambda_{22} < \frac{P(x | y_1)P(y_1)}{P(x | y_2)P(y_2)}$$

Rely on $\lambda$

Rely on the distribution
Classification: Bayes Decision Rule: Action Cost

Minimum Risk Decision Rule

\[ \theta_\lambda = \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} < \frac{p(x|y_1)P(y_1)}{p(x|y_2)P(y_2)} \]

Zero-one loss function (Bayes Rule)

\[ \lambda_0 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \theta_{\lambda_0} = 1 \]

Another loss function

\[ \lambda_\alpha = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} \quad \theta_{\lambda_\alpha} = 2 \]

Loss(Taking \( a_2 \) when \( y_1 \)) >
Loss(Taking \( a_1 \) when \( y_2 \))

Loss(Classify as \( y_2 \) when \( y_1 \)) >
Loss(Classify as \( y_1 \) when \( y_2 \))

Reduce to chance of predicting \( y_1 \)

Learning Type

- Two types of learning method:
  - **Parametric Methods**
    - Assume the form of sample distribution (pdf) is known, E.g. Gaussian distribution
    - Estimate parameters of the distribution
    - Bias (Good if the assumption is correct)
  - **Non-Parametric Methods**
    - No assumption on pdf
    - Instead, the proper form for discriminant function is assumed
    - Usually sub-optimal, but good results generally