Artificial Intelligence III:
Artificial Intelligence and Deep Learning

Lecture 3

Regression & Classification

Dr. Patrick Chan
patrickchan@ieee.org
South China University of Technology, China

Agenda

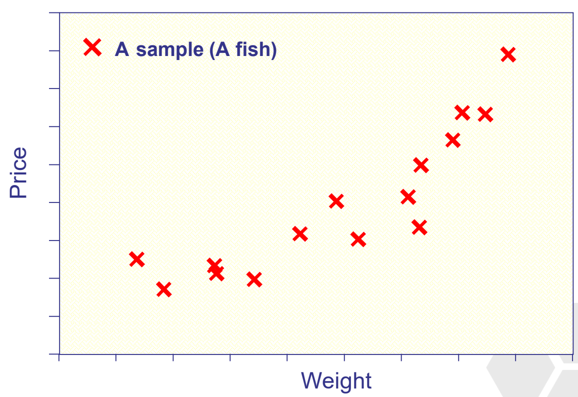
- Supervised Learning
  - Regression
    - Least Mean Square
  - Classification
    - Probability

Regression

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

Weight (kg)	Price ($)
2.2	20
2.6	31
1.2	16.5
0.7	10

⋮



Regression

- The i^th^ training sample: $(x^{(i)}, y^{(i)})$
  - $x^{(i)} = \left[x_1^{(i)}, x_2^{(i)}, \dots, x_d^{(i)}\right]^T \in X$: **Feature vector**
    - $x_j^{(i)}$: the jth feature
    - X : the input space
  - $y^{(i)} \in Y$: **Target Value**
    - Y : the output space

	x Weight (kg)	y Price ($)
$(x^{(1)}, y^{(1)})$	2.2	20
$(x^{(2)}, y^{(2)})$	2.6	31
$(x^{(3)}, y^{(3)})$	1.2	16.5
$(x^{(4)}, y^{(4)})$	0.7	10

⋮

- Training set: $\{(x^{(i)}, y^{(i)}) \mid i = 1 \dots n\}$
  - n is the number of training samples

Regression

- Train a function $h_\theta(x)$ to predict y
  - θ is the parameter vectors of the model
    - Different parameters yields different predictions (different h)
    - Example: weight
  - $h_\theta : X \rightarrow Y$, mapping from X to Y
  - h_θ is called a predictor or hypothesis
- Objective: Build a "good" h_θ
  - 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)
  - May contains other terms besides Error
  - Mean Square Error is a classic measure

$$J(\theta) = \underbrace{\frac{1}{2n}\sum_{i=1}^{n}\big(\overbrace{h_\theta(x^{(i)}) - y^{(i)}}^{\text{square error}}\big)^2}_{\text{mean square error}}$$

  - 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 θ

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

- ML is closely related to Optimization problem
  - Usually, the optimization is quite complicated
    - Since each parameter is a variable
    - Iterative method is used

Gradient Descent

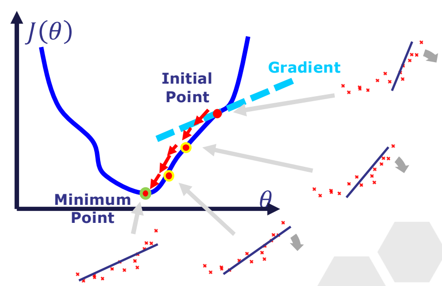
- When h_θ is differentiable, gradient descent can be used to minimize $J(\theta)$

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

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

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

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



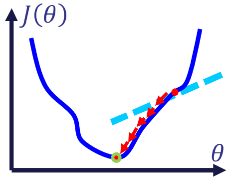
LMS Algorithm

Gradient Descent

- **Algorithm**
  - Start with an arbitrarily chosen weight $\theta^{(1)}$
  - Let t = 0
  - Loop
    - t = t + 1
    - Compute gradient vector $\partial J(\theta^{(t)})/\partial\theta$
    - 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, $\theta = [\theta_1, \theta_2, \dots, \theta_m]$
- Updated Rule for the j^{th} parameter

$$\theta_j^{(t+1)} = \theta_j^{(t)} - \alpha \frac{\partial}{\partial\theta_j} J\left(\theta_j^{(t)}\right)$$

- All parameters should be updated at the same time

LMS Algorithm

Gradient Descent

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

$$\theta_j = \theta_j - \alpha \boxed{\frac{\partial}{\partial\theta_j} J(\theta)}$$

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

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

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

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

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

Depend on a model

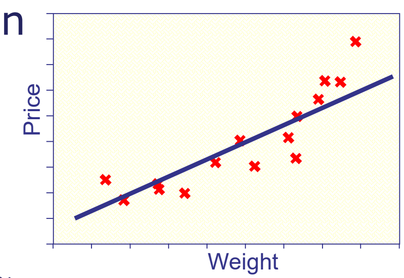
LMS Algorithm

Gradient Descent

- Example: Linear Function

$$h_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_d x_d$$

$$= \sum_{i=1}^{d} \theta_i x_i$$



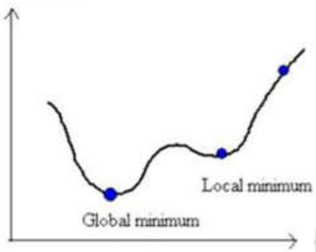
$$\frac{\partial}{\partial\theta_j} J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} 2\left(h_\theta(x^{(i)}) - y^{(i)}\right) \frac{\partial h_\theta(x^{(i)})}{\partial\theta_j}$$

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

$$= \frac{1}{n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$$

LMS Algorithm
Gradient Descent

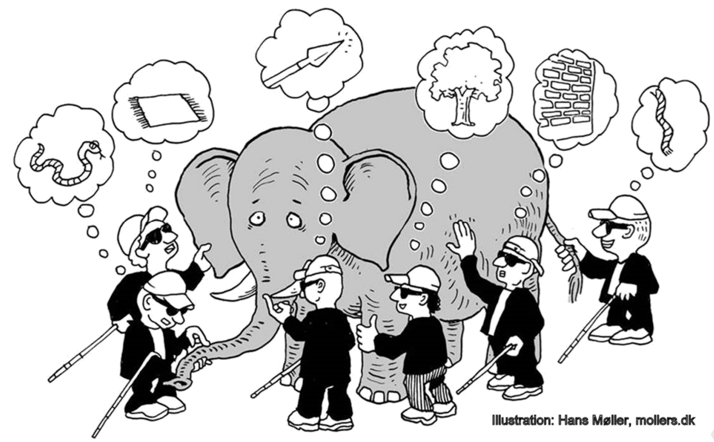
- Related Issues:
  - Size of Learning Rate (α)
    - Too small, convergence is needlessly slow
    - Too large, the correction process will overshoot and cannot even diverge
  - Sub-optimal Solution
    - Trapped by local minimum



  - We will study Gradient Descent again in Artificial Neural Network

Classification

- Objective: output a class based on the features of a sample



Classification
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

    $P(y = \text{healthy}) > P(y = \text{ill})$

Person	Status (y)
A	Ill
B	Healthy
C	Healthy
D	Ill

⋮

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

Classification
Likelihood

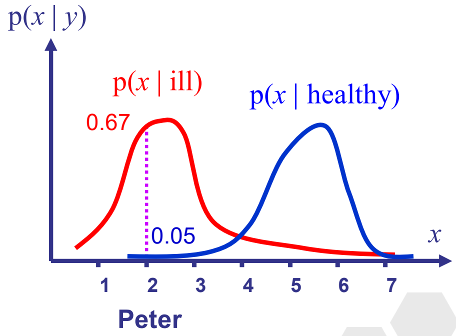
- Physical condition of persons should be considered
  - Quantify the characteristics (features), denoted by $\boldsymbol{x}$
  - E.g. red blood cell #, white blood cell #, temperature
- Assume only "white blood cell #" is measured

Person	White Blood Cell # (x)	Status (y)
A	50	Ill
B	42	Healthy
C	39	Healthy
D	62	Ill

⋮

Likelihood

- 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 | ill) = 0.67
  - p(x=2 | healthy) = 0.05
  - Therefore, Peter is ill
- Should we be satisfied?
  - This decision is based on **Likelihood** $p(x|y)$



Posterior Probability

- Using Prior Probability ($P(y)$) or Likelihood ($p(x|y)$) is not suitable
- Posterior Probability is a better choice
  $P(y|x)$: given x, 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 obtain $P(y|x)$?
  - Obtaining from data is difficult as x is usually a continuous value

Bayes Formula

- **Bayes Formula**

$$\underbrace{P(y|x)}_{\text{posterior}} = \frac{\overbrace{p(x|y)}^{\text{likelihood}}\overbrace{P(y)}^{\text{prior}}}{\underbrace{p(x)}_{\text{evidence}}}$$

- Likelihood and prior probability may be estimated by using a dataset *(Discuss it later in the lecture)*
- How about evidence $p(x)$?

Bayes Decision Rule

- $p(x)$ is difficult to be obtained relatively
  - $p(x)$ contain all kinds of samples, which is more complicated than $p(x|y)$
  - It can be neglected in decision making
- x is classified as y_1 if

$$\boxed{P(y|x) = \frac{p(x|y)P(y)}{p(x)}}$$

$$p(y_1|x) > p(y_2|x)$$

$$\frac{p(x|y_1)P(y_1)}{p(x)} > \frac{p(x|y_2)P(y_2)}{p(x)}$$

$$p(x|y_1)P(y_1) > p(x|y_2)P(y_2)$$

Classification
Bayes Decision Rule

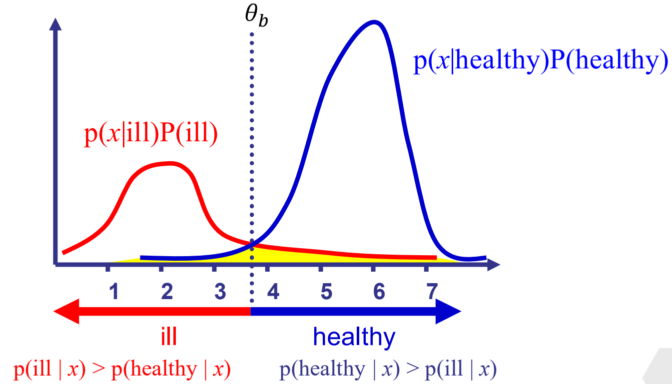
- Given: p(x=2 | ill) = 0.67 p(x=2 | healthy) = 0.05
  P(ill) = 0.15 P(healthy) = 0.85
- Recall, Bayes Decision Rule
  - Decide y_1 if $P(y_1 | x) > P(y_2 | x)$
  - Decide y_2 if $P(y_2 | x) > P(y_1 | x)$
- $P(\text{healthy} \mid x = 2) \propto p(x{=}2 \mid \text{healthy}) \times P(\text{healthy})$
  $= 0.05 \times 0.85 = 0.0425$
- $P(\text{ill} \mid x = 2) \propto p(x{=}2 \mid \text{ill}) \times P(\text{ill})$
  $= 0.67 \times 0.15 = 0.1005$

* 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
Decision Boundary

- Recall, Bayes Decision Rule:
  - if $P(y_1| x) > P(y_2| x)$, decide y_1; otherwise decide y_2
- Its Decision Boundary:



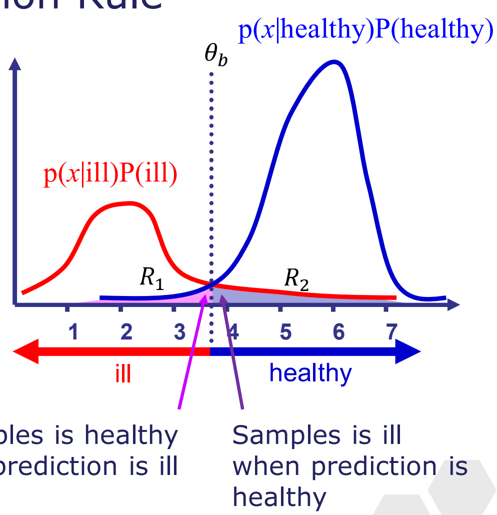
Classification: Bayes Decision Rule
Decision Boundary

- Error of Bayes Decision Rule

Prediction

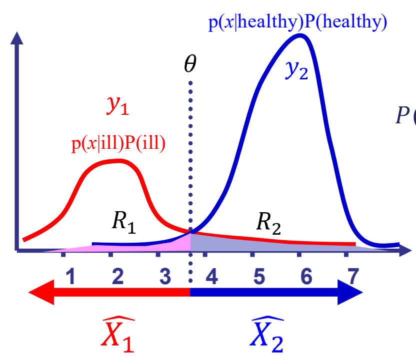
True	$\hat{y}_1$	$\hat{y}_2$
y_1	Predict: y_1 True: y_1	Predict: y_2 True: y_1
y_2	Predict: y_1 True: y_2	Predict: y_2 True: y_2

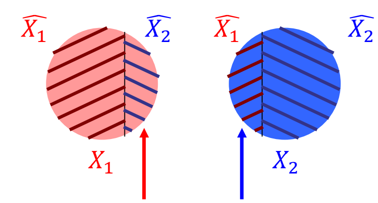
Correct (blue)
Wrong (dark red)



Classification: Bayes Decision Rule
Error

- Error Probability





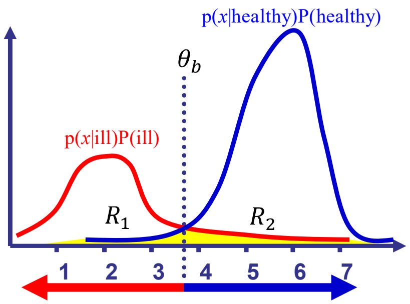
$$P(error) = P(x \in \widehat{X_2}, y_1) + P(x \in \widehat{X_1}, y_2)$$
$$= P(x \in \widehat{X_2}|y_1)P(y_1) + P(x \in \widehat{X_2}|y_2)P(y_2)$$
$$= \int_{\widehat{X_2}} p(x|y_1)P(y_1)dx + \int_{\widehat{X_1}} p(x|y_2)P(y_2)dx$$

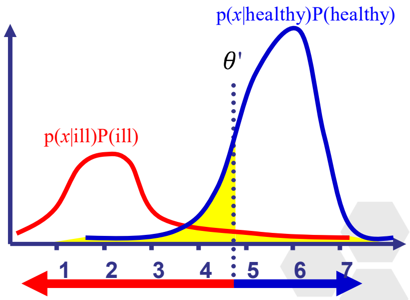
$\widehat{X_1} = \{x \mid x \text{ is classified as } y_1\}$
$\widehat{X_2} = \{x \mid x \text{ is classified as } y_2\}$
Prediction

$X_1 = \{x \mid x \text{ belongs to } y_1\}$
$X_2 = \{x \mid x \text{ belongs to } y_2\}$
True

Error

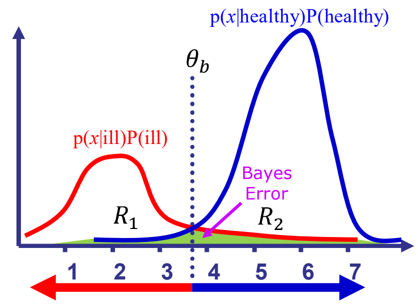
- θ_b or θ' is better?
  - Error of $\theta_b < \theta'$
  - θ_b is better
- Is any boundary better than Bayes Decision Rule (θ_b)?
  - Bayes Rule is optimal (minimal classification error)

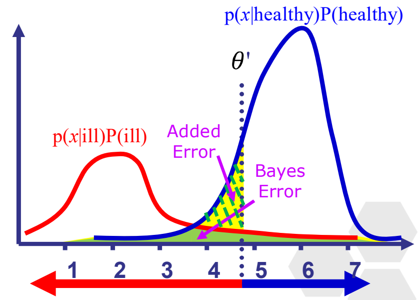




Error

- **Error = Bayes Error + Added Error**
- **Bayes Error**
  Error of Bayes Rule
  - Cannot be reduced
  - Depend on the input space and application
- **Added Error**
  Additional error made by other classifiers
  - Can be reduced by selecting better parameters





Extension to Multi-Class

- Extend to multi-class problem (c classes)

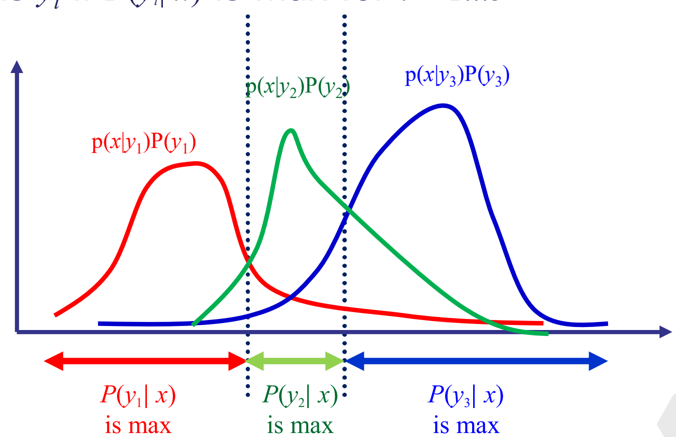
$$y = (y_1, y_2, \ldots, y_c)$$

- Bayes Decision Rule
  - x is y_i if $P(y_i | x)$ is maximum for $i = 1 \ldots c$
- Error for Bayes Decision Rule

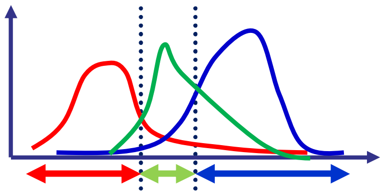
$$P(error \mid x) = 1 - \max[\, P(y_1|x), P(y_2|x), \ldots, P(y_c|x)\,]$$

Extension to Multi-Class

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



Classification: Bayes Decision Rule
Extension to Multi-Class



Error of Bayes Decision Rule:

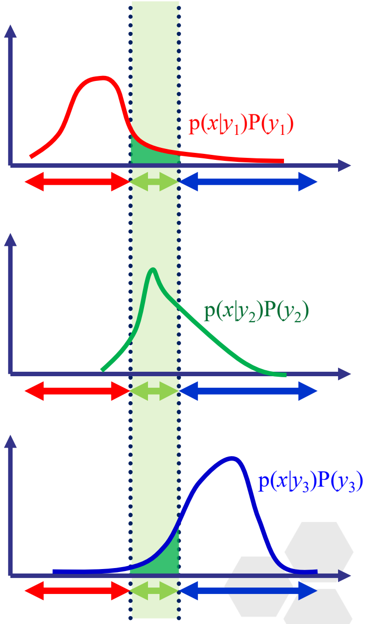
$$P(error \mid x) = 1 - \max[P(y_1|x), P(y_2|x), P(y_3|x)]$$

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

$$P(error \mid x) = P(y_1 \mid x) + P(y_3 \mid x) = 1 - P(y_2 \mid x) = 1 - \max_{i=1,2,3} P(y_i \mid x)$$

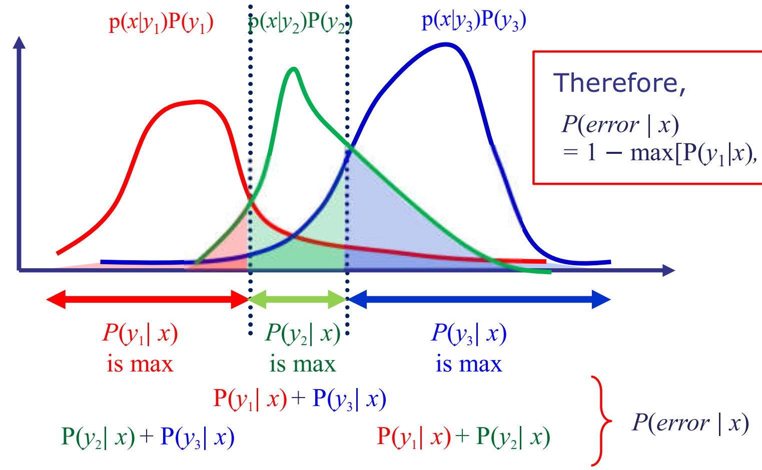
* Posterior probability (figures are not)

* Must not be $P(y_2|x)$



Classification: Bayes Decision Rule
Extension to Multi-Class

- Three-class example
  - Error of Bayes Decision Rule



Therefore,

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

How to apply Bayes Rules?

- Bayes Rules:

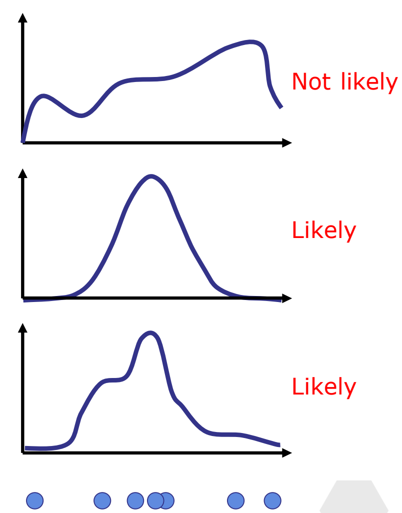
  If $p(x|y_1)P(y_1) > p(x|y_2)P(y_2)$, x is classified as y_1
  Otherwise, it is y_2

- How to learn $p(x \mid y_i)$ and $P(y_i)$ in an application?
  - $P(y_i)$: Ratio of the class i
    - y_i is discrete, easy to estimate
  - $p(x \mid y_i)$: Distribution of samples in the class i
    - x is usually continues and has many dimensions, different to estimate

How to apply Bayes Rules?
$p(x \mid y_i)$ and $P(y_i)$ Estimation

- $p(x \mid y_i)$ means
  $p(x)$ and x is from y_i

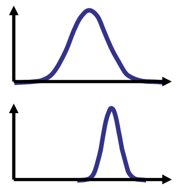
- Given samples of a class (x is from y_i), how can we know its real distribution $p(x)$?
  - By estimation



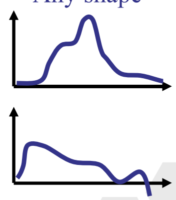
$p(x \mid y_i)$ Estimation
Type of Learning

- How to estimate $P(x \mid y_i)$?
  - **Parametric Methods** (Briefly Introduce here)
    - Model-based Method *Assume form of sample distribution (pdf) is known*
    - Estimate distribution parameters
    - Bias (Great if the assumptions are correct)
  - **Non-Parametric Methods** (Part 2, NN)
    - Model Free Method *No assumption on pdf*
    - A proper form for discriminant function is assumed
    - Usually sub-optimal, but good results generally

Shape is fixed



Any shape



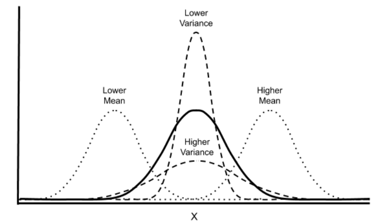
Parametric Methods
Normal Distribution

- Assume samples in each class follow normal distribution (Gaussian distribution),

$$D \sim \mathrm{N}(\mu, \sigma^2)$$

- 1-dimension: $p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$
- d-dimension: $p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^t\Sigma^{-1}(x-\mu)\right]$

$x = (x_1, x_2, \ldots, x_d)^t$: sample vector
$\mu = (\mu_1, \mu_2, \ldots, \mu_d)^t$: mean vector
$\Sigma = d \times d$: covariance matrix
σ : variance (d=1 of covariance matrix)
$|\Sigma|$ and Σ^{-1} : determinant and inverse
t : transpose



Discriminant Functions for the Normal Density

- To simplify calculations by transforming multiplication into addition

$$p(x|y)P(y) \quad \text{and} \quad p(x|y) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^t\Sigma^{-1}(x-\mu)\right]$$

- The natural log function is a monotonic increasing function

$$p(x|y)P(y) \propto \ln\big(p(x|y)P(y)\big) = \ln\big(p(x|y)\big) + \ln\big(P(y)\big)$$

- g(x) is used for comparison

$$g(x) = \ln\big(p(x|y)\big) + \ln\big(P(y)\big)$$

Discriminant Functions for the Normal Density

- Substitute $p(x|y)$ to $g(x)$

$$g(x) = \ln\big(p(x|y)\big) + \ln\big(P(y)\big)$$

$$p(x|y) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^t\Sigma^{-1}(x-\mu)\right]$$

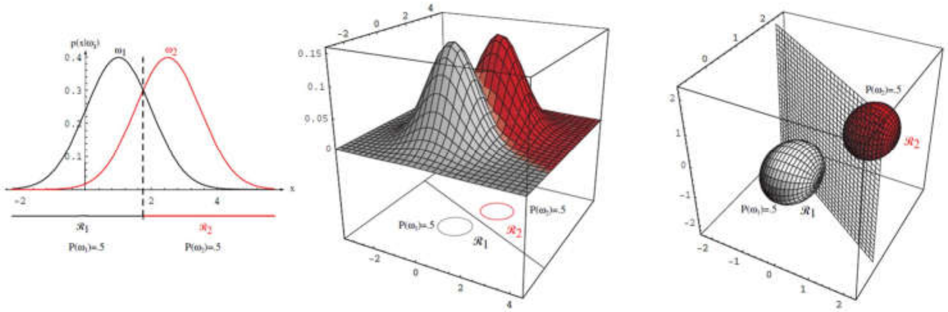
- Therefore:

$$\boxed{g(x) = -\frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(x-\mu)^t\Sigma^{-1}(x-\mu) + \ln P(y)}$$

Parametric Methods

Normal Distribution ($\Sigma_i = \sigma^2 I$)

- Assume the covariance matrices are the identity matrix, the distributions are spherical

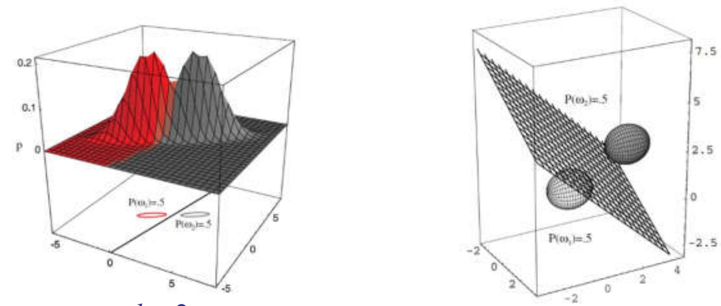


$d = 1$ $d = 2$ $d = 3$

Parametric Methods

Normal Distribution ($\Sigma_i = \sigma^2 I$)

- Assume each class has the same covariance matric, the distributions is in ellipse sharp

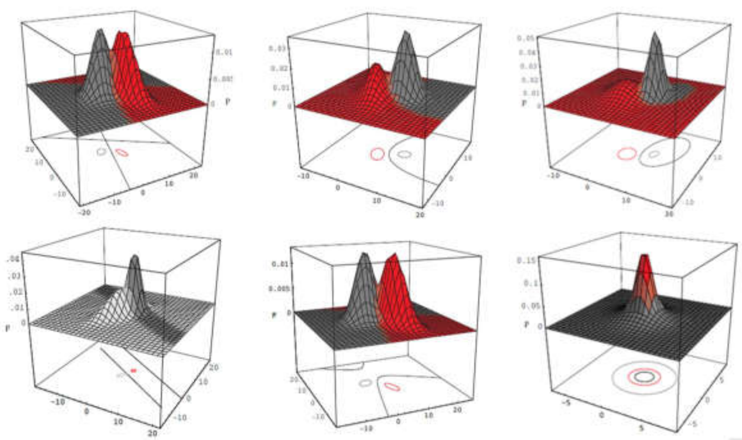


$d = 2$ $d = 3$

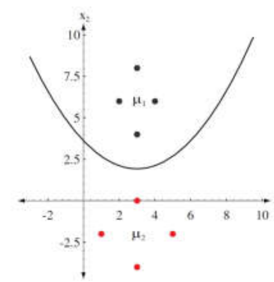
Parametric Methods

Normal Distribution ($\Sigma_i = \sigma^2 I$)

- The covariance matric can be anything



Discriminant Functions for Multivariate Normal Density



- Given $\mu_1 = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$ $\mu_2 = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$
  $\Sigma_1 = \begin{pmatrix} 1/2 & 0 \\ 0 & 2 \end{pmatrix}$ $\Sigma_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$
- Inverse $\Sigma_1^{-1} = \begin{pmatrix} 2 & 0 \\ 0 & 1/2 \end{pmatrix}$ $\Sigma_2^{-1} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$
- Assume $P(y_1) = P(y_2) = 0.5$
- Decision Boundary $g_1(x) = -(x_1 - 3)^2 - \frac{1}{4}(x_2 - 6)^2 - ln(4\pi)$

$$g_2(x) = -\frac{1}{4}(x_1 - 3)^2 - \frac{1}{4}(x_2 + 2)^2 - ln(8\pi)$$

$$g_1(x) - g_2(x) = -\frac{3}{4}(x_1 - 3)^2 - \frac{1}{2}(x_2 - 2)^2 - l\,n(2) = 0$$

$$\boxed{g_i(x) = -\frac{1}{2}(x - \mu_i)^t \Sigma_i^{-1}(x - \mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)}$$

Normal Distribution

- Multi-class problem
  - Even with small number of classes, the shapes of the boundary regions is complex