UVA CS 6316/4501 – Fall 2016 Machine Learning

Lecture 12: Generative Bayes Classifiers

Dr. Yanjun Qi

University of Virginia

Department of Computer Science

Where are we ? ➔
Five major sections of this course

- ❑ Regression (supervised)
- ❑ Classification (supervised)
- ❑ Unsupervised models
- ❑ Learning theory
- ❑ Graphical models

Where are we ? ➔

Three major sections for classification

- We can divide the large variety of classification approaches into roughly three major types

1. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., support vector machine, decision tree

2. Generative:
   - build a generative statistical model
   - e.g., naïve bayes classifier, Bayesian networks

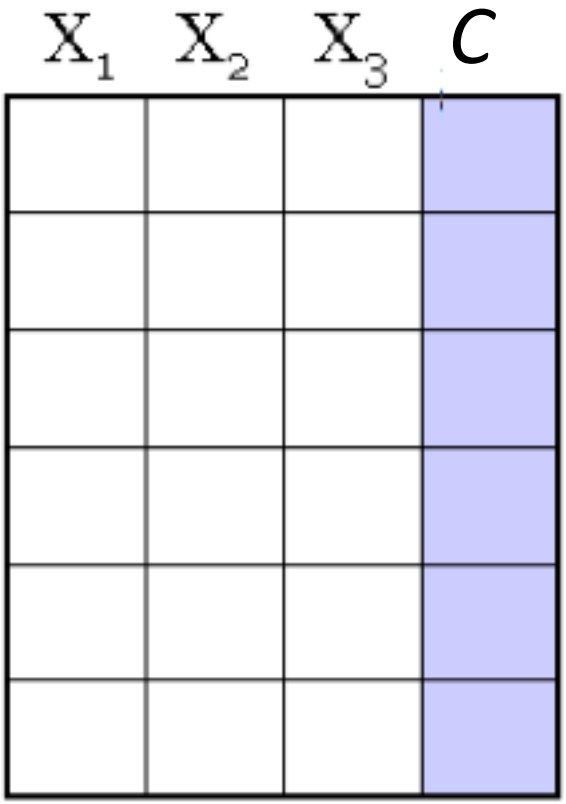
3. Instance based classifiers
   - Use observation directly (no models)
   - e.g. K nearest neighbors

Last Lecture Recap: Probability Review

- The big picture : data <-> probabilistic model
- Sample space, Events and Event spaces
- Random variables
- Joint probability, Marginal probability, conditional probability,
- Chain rule, Bayes Rule, Law of total probability, etc.
- Structural properties
  - Independence, conditional independence

Today : Generative Bayes Classifiers

- ✓ Bayes Classifier
  - MAP classification rule
  - Generative Bayes Classifier
- ✓ Naïve Bayes Classifier
- ✓ Gaussian Bayes Classifiers
  - Gaussian distribution
  - Gaussian NBC
  - LDA, QDA



A Dataset for classification

$$f : X \longrightarrow C$$

Output as Discrete Class Label
$C_1, C_2, \ldots, C_L$

$$P(C \mid \mathbf{X})$$

$(X_1, X_2, \ldots, X_p)$

- **Data**/*points/instances/examples/samples/records*: [rows]
- **Features**/*attributes/dimensions/independent variables/covariates/ predictors/regressors*: [columns, except the last]
- **Target**/*outcome/response/label/dependent variable*: special column to be predicted [last column]

Bayes classifiers

- Treat each feature attribute and the class label as random variables.

Bayes classifiers

- Treat each feature attribute and the class label as random variables.

- Given a sample **x** with attributes ($x_1, x_2, \ldots, x_p$):
  - Goal is to predict its class *C*.
  - Specifically, we want to find the value of C_i that maximizes $p(C_i \mid x_1, x_2, \ldots, x_p)$.

Bayes classifiers

- Treat each feature attribute and the class label as random variables.

- Given a sample **x** with attributes ($x_1, x_2, \ldots, x_p$):
  - Goal is to predict its class C. → $\{c_1, c_2, \ldots, C_L\}$
  - Specifically, we want to find the value of C_i that maximizes $p(C_i \mid x_1, x_2, \ldots, x_p)$.

- Can we estimate $p(C_i \mid \mathbf{x}) = p(C_i \mid x_1, x_2, \ldots, x_p)$ directly from data?

Bayes classifiers
➔ MAP classification rule

- Establishing a probabilistic model for classification

➔ **MAP** classification rule

- **MAP**: **M**aximum **A** **P**osterior

Bayes classifiers
➔ MAP classification rule

- Establishing a probabilistic model for classification

➔ **MAP** classification rule

- **MAP**: **M**aximum **A** **P**osterior
- Assign x to c^* if

$$c^* = \underset{c_i \in \{c_1, c_2, \cdots, c_L\}}{\operatorname{argmax}} P(c_i|x)$$

$$P(C = c^* | \mathbf{X} = \mathbf{x}) > P(C = c | \mathbf{X} = \mathbf{x})$$

$$\text{for } c \neq c^*, \; c = c_1, \cdots, c_L$$

Adapt from Prof. Ke Chen NB slides

Bayes classifiers
➔ MAP classification rule

- Establishing a probabilistic model for classification

➔ **MAP** classification rule

- **MAP**: **M**aximum **A** **P**osterior
- Assign x to c^* if

$$P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x}) \quad c \neq c^*,\ c = c_1, \cdots, c_L$$

$$\left\{ \begin{array}{l} P(C=c_1 \mid x) \\ P(C=c_2 \mid x) \\ P(C=c_3 \mid x) \end{array} \right\} \max \Rightarrow c_i$$

Adapt from Prof. Ke Chen NB slides

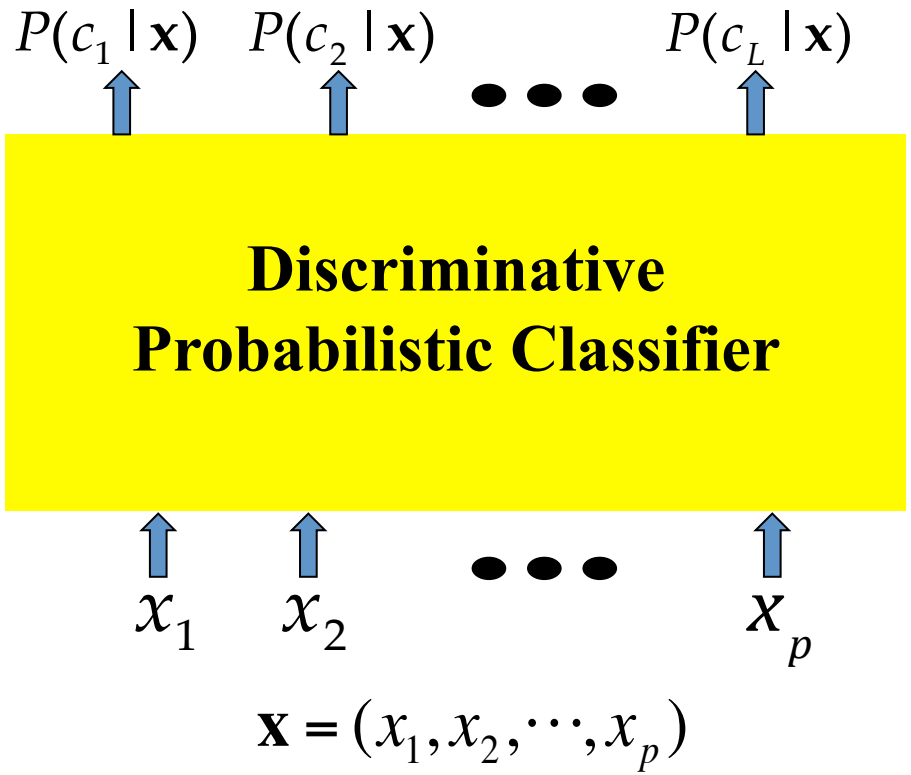
Bayes classifiers
➔ MAP classification rule

- Establishing a probabilistic model for classification
  - **(1) Discriminative**
  - **(2) Generative**

(1) Discriminative

$$P(C|\mathbf{X})$$

$$C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_p)$$



logistic regression

Adapt from Prof. Ke Chen NB slides

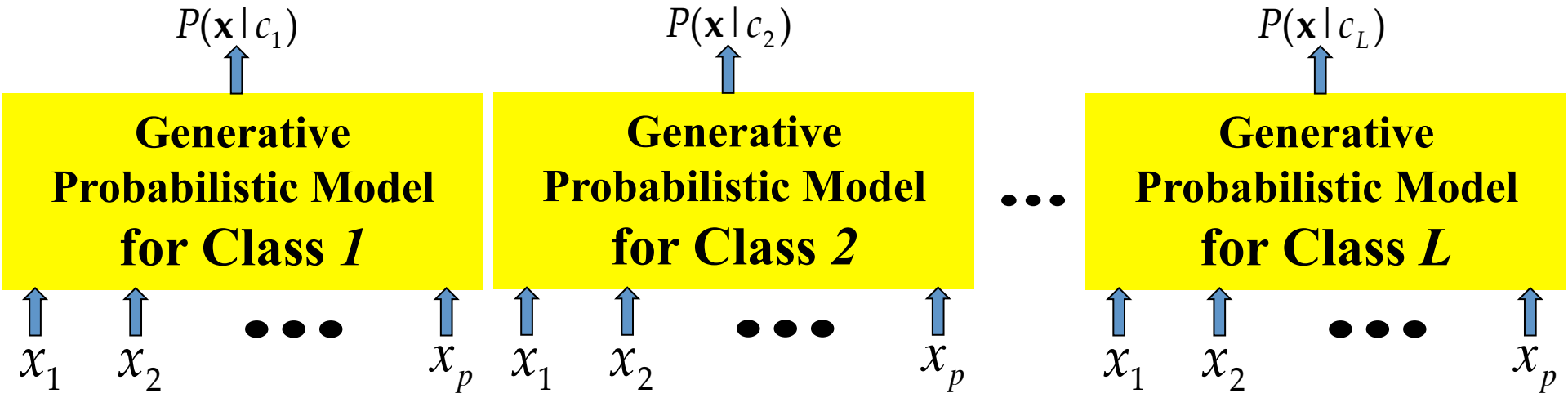
(2) Generative

$p(x|c_1), p(x|c_2), \cdots, p(x|c_L)$

$P(\mathbf{X}|C)$,

$C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_p)$

$\Rightarrow p(c|x)$ MAP rule



$\mathbf{x} = (x_1, x_2, \cdots, x_p)$

Adapt from Prof. Ke Chen NB slides

(2) Generative

Bayes Rule

generative model

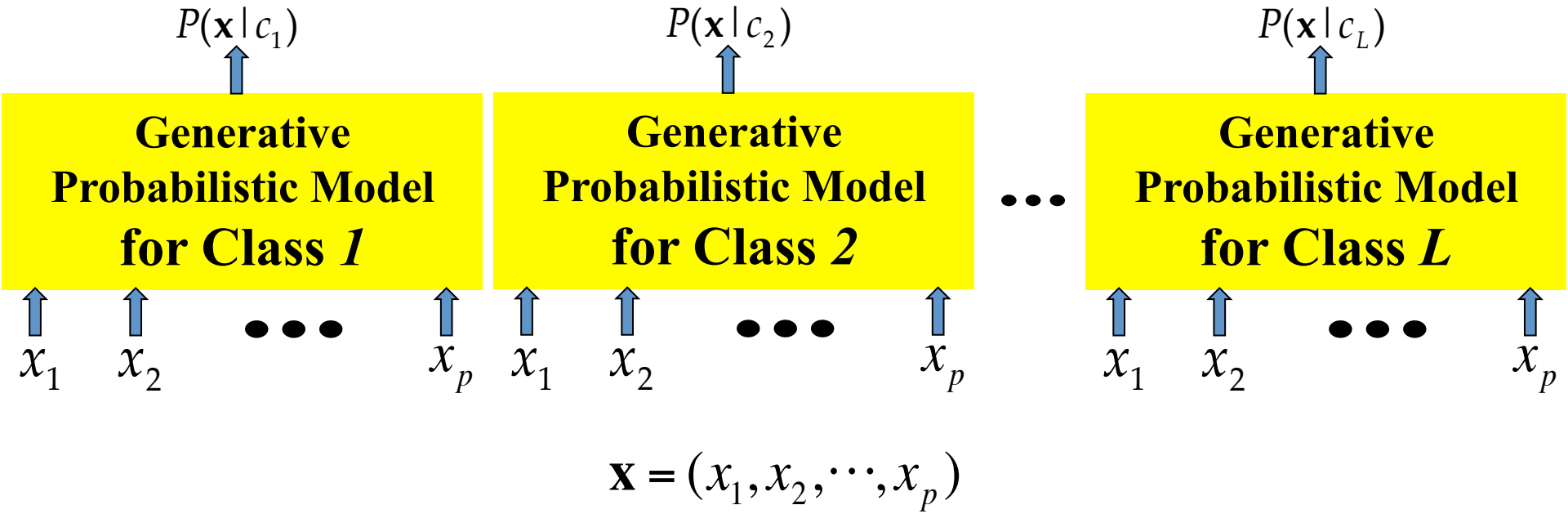
$$p(c|\mathbf{x}) = \frac{p(\mathbf{x}|c)\,p(c)}{p(\mathbf{x})}$$

$$P(\mathbf{X}|C),$$

$$C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_p)$$

$$p(X = x_i \,|\, C = c_i)$$

$$(x_1, x_2, \cdots, x_p)$$



$$\mathbf{x} = (x_1, x_2, \cdots, x_p)$$

Adapt from Prof. Ke Chen NB slides

Review : Bayes' Rule – for Generative Bayes Classifiers

$$P(C,X) = P(C \mid X)P(X) = P(X \mid C)P(C)$$

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}$$

Review : Bayes' Rule – for Generative Bayes Classifiers

$$P(C,X) = P(C \mid X)P(X) = P(X \mid C)P(C)$$

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}$$

$P(C_1), P(C_2), \ldots, P(C_L)$

max ⇒ C^*

$P(C_1|x), P(C_2|x), \ldots, P(C_L|x)$

$$P(C_i \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C_i)P(C_i)}{P(\mathbf{X})}$$

Review : Bayes' Rule – for Generative Bayes Classifiers

$$P(C,X) = P(C \mid X)P(X) = P(X \mid C)P(C)$$

[MAP rule]

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}$$

Posterior: $P(C_1|x), P(C_2|x), \ldots, P(C_L|x)$

Prior: $P(C_1), P(C_2), \ldots, P(C_L)$

max ⇒ C^*

$$P(C \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})}$$

Summary:
Generative classification with the MAP rule

- MAP classification rule
  - **MAP**: **M**aximum **A** **P**osterior
  - Assign x to c^* if

$$P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x}) \quad c \neq c^*, \; c = c_1, \cdots, c_L$$

- Generative classification with the MAP rule

Summary:
Generative classification with the MAP rule

$$P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x}) \quad c \neq c^*,\ c = c_1, \cdots, c_L$$

- Generative classification with the MAP rule
  - Apply Bayes rule to convert them into posterior probabilities

$$P(C = c_i \mid \mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X} = \mathbf{x} \mid C = c_i) P(C = c_i)}{P(\mathbf{X} = \mathbf{x})}$$
$$\propto P(\mathbf{X} = \mathbf{x} \mid C = c_i) P(C = c_i)$$
$$\text{for } i = 1, 2, \cdots, L$$

$P(C=c_i)$

$P(X=x \mid C=c_i)$

$\text{argmax}_{i=1,2,\cdots,L} \{ P(c_i \mid x) = \frac{P(x \mid c_i) P(c_i)}{P(x)} \}$

$= \text{argmax}_{i=1,2,\cdots,L} \{ P(x \mid c_i) P(c_i) \}$

  - Then apply the MAP rule

Adapt from Prof. Ke Chen NB slides

Summary: Generative Bayes Classifier with the MAP rule

Task: Classify a new instance *X* based on a tuple of attribute values $X = \langle X_1, X_2, \ldots, X_p \rangle$ into one of the classes

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_p)$$

$$= \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_p \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_p)}$$

$$= \underset{c_j \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_p \mid c_j) P(c_j)$$

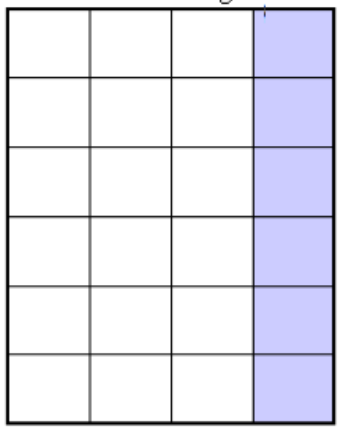
$j = 1, 2, \ldots, L$

MAP = Maximum A Posteriori

Adapt From Carols' prob tutorial

An Example

- Example: Play Tennis



PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$p(C=\text{Yes} \mid X_1, X_2, X_3, X_4)$

$p(C=\text{No} \mid X_1, X_2, X_3, X_4)$

→ $p(C_1=\text{Yes}) = 9/14$

$p(C_2=\text{No}) = 5/14$

→ $p(X_1, X_2, X_3, X_4 \mid C_i)$

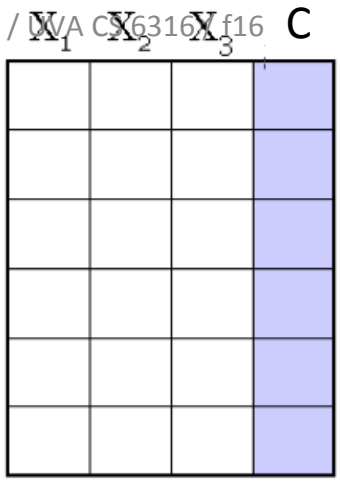
$3 \times 3 \times 2 \times 2 \times 2 \Rightarrow 72$ parameters from train

→ $\underset{i=1,2}{\text{argmax}}\ p(\bar{X}_{ts} \mid C_i)\, p(C_i)$ Generative BC

Example

$P(X_1, X_2, X_3, X_4 \mid Yes)$
$P(X_1, X_2, X_3, X_4 \mid No)$

36 x 2



- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

3 3 x 2 x 2 = 36

$P(C = C_i)$

{Yes, No}

$P(C = Yes) = 9/14$

$P(C = No) = 5/14$

- maximum likelihood estimates (explain later)
  - simply use the frequencies in the data

e.g. $P(\text{Overcast}, \text{Hot}, \text{High}, \text{Weak} \mid \text{Yes}) = \frac{1}{9}$

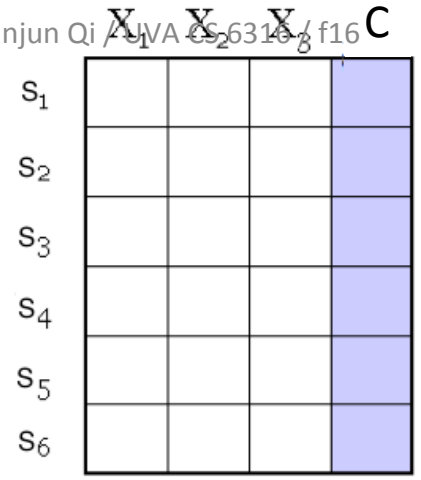
PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

e.g.

$P(\text{Overcast}, \text{Hot}, \text{High}, \text{Weak} / \text{No}) = \frac{0}{9}$

Generative Bayes Classifier:



- Learning Phase

$P(C_1), P(C_2), \ldots, P(C_L)$

$P(\text{Play}=Yes) = 9/14 \quad P(\text{Play}=No) = 5/14$

$P(X_1, X_2, \ldots, X_p|C_1), P(X_1, X_2, \ldots, X_p|C_2)$

a look up table of cond. prob

Outlook (3 values)	Temperature (3 values)	Humidity (2 values)	Wind (2 values)	Play=*Yes*	Play=*No*
sunny	*hot*	*high*	*weak*	0/9	1/5
sunny	*hot*	*high*	*strong*	…/9	…/5
sunny	*hot*	*normal*	*weak*	…/9	…/5
sunny	*hot*	*normal*	*strong*	…/9	…/5
….	….	….	….	….	….
….	….	….	….	….	….
….	….	….	….	….	….
….	….	….	….	….	….

3*3*2*2 [conjunctions of attributes] * 2 [two classes]=72 parameters

Generative Bayes Classifier:

$$[\hat{P}(a'_1|c^*)\cdots\hat{P}(a'_p|c^*)]\hat{P}(c^*) > [\hat{P}(a'_1|c)\cdots\hat{P}(a'_p|c)]\hat{P}(c)$$

- Test Phase
  - Given an unknown instance $\mathbf{X}'_{ts} = (a'_1, \cdots, a'_p)$
  - Look up tables to assign the label c^* to $\mathbf{X}_{ts}$ if

$$\hat{P}(a'_1, \cdots a'_p|c^*)\hat{P}(c^*) > \hat{P}(a'_1, \cdots a'_p|c)\hat{P}(c),$$

$$c \neq c^*, \; c = c_1, \cdots, c_L$$

  - Given a new instance,

    $\mathbf{x}'$ =(Outlook=*Sunny,* Temperature=*Cool,* Humidity=*High,* Wind=*Strong*)

$$\left\{ \begin{array}{l} P(x'|Yes)\,P(C=Yes) \\ P(x'|No)\,P(C=No) \end{array} \right\} \Rightarrow \underset{C}{\arg\max} \Rightarrow \text{predicted } C^*$$

Today : Generative Bayes Classifiers

- ✓ Bayes Classifier
  - MAP classification rule
  - Generative Bayes Classifier
- ✓ Naïve Bayes Classifier
- ✓ Gaussian Bayes Classifiers
  - Gaussian distribution
  - Gaussian NBC
  - LDA, QDA

Naïve Bayes Classifier

- Bayes classification

$$\underset{c_j \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_p \mid c_j) P(c_j)$$

Difficulty: learning the joint probability

- Naïve Bayes classification
  - Assumption that all input attributes are conditionally independent!

Naïve Bayes Classifier

- Naïve Bayes classification
  - Assumption that all input attributes are conditionally independent!

$$P(X_1, X_2, \cdots, X_p \mid C) = P(X_1 \mid X_2, \cdots, X_p, C) P(X_2, \cdots, X_p \mid C)$$
$$= P(X_1 \mid C) P(X_2, \cdots, X_p \mid C)$$
$$= P(X_1 \mid C) P(X_2 \mid C) \cdots P(X_p \mid C)$$

Naïve Bayes Classifier

- Naïve Bayes classification
  - Assumption that all input attributes are conditionally independent!

$$P(X_1, X_2, \cdots, X_p | C) = P(X_1 | C) P(X_2 | C) \cdots P(X_p | C)$$

  - MAP classification rule: for a sample $\mathbf{x} = (x_1, x_2, \cdots, x_p)$

$$[P(x_1 | c^*) \cdots P(x_p | c^*)] P(c^*) > [P(x_1 | c) \cdots P(x_p | c)] P(c),$$

$$c \neq c^*, \; c = c_1, \cdots, c_L$$

$\Rightarrow \operatorname{argmax}_{i=1,\cdots,L} P(c_i) P(x_1 | c_i) P(x_2 | c_i) \cdots P(x_p | c_i)$

Naïve Bayes Classifier

- Naïve Bayes classification
  - Assumption that all input attributes are conditionally independent!

$$P(X_1, X_2, \cdots, X_p | C) = P(X_1 | C) P(X_2 | C) \cdots P(X_p | C)$$

  - MAP classification rule: for a sample $\mathbf{x} = (x_1, x_2, \cdots, x_p)$

$$[P(x_1 | c^*) \cdots P(x_p | c^*)] P(c^*) > [P(x_1 | c) \cdots P(x_p | c)] P(c),$$

$$c \neq c^*, \; c = c_1, \cdots, c_L$$

$\{j=1,2,\cdots,p\}$ $\{i=1,2,\cdots,L\}$

$P(X_j | C_i)$

{ Bernolli (P)
Binomial (K,P)
multinomial
Gaussian

Naïve Bayes Classifier (for discrete input attributes) - training

- Naïve Bayes Algorithm (for discrete input attributes)
  - Learning Phase: Given a training set S,

For each target value of c_i $(c_i = c_1, \cdots, c_L)$

$\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in $\mathbf{S}$;

Naïve Bayes Classifier (for discrete input attributes) - training

- Naïve Bayes Algorithm (for discrete input attributes)
  - Learning Phase: Given a training set S,

For each target value of c_i $(c_i = c_1, \cdots, c_L)$

$\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in $\mathbf{S}$;

For every attribute value x_{jk} of each attribute X_j $(j = 1, \cdots, p;\ k = 1, \cdots, K_j)$

$\hat{P}(X_j = x_{jk} \mid C = c_i) \leftarrow$ estimate $P(X_j = x_{jk} \mid C = c_i)$ with examples in $\mathbf{S}$;

$P(C_1)\ P(C_2)\ \cdots\ P(C_L)$

L parameters

Output: conditional probability tables; for $X_j, \mathrm{K}_j \times L$ elements

Naïve Bayes Classifier (for discrete input attributes) - training

- Naïve Bayes Algorithm (for discrete input attributes)
  - Learning Phase: Given a training set S,

For each target value of c_i ($c_i = c_1, \cdots, c_L$)

$\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in **S**;

For every attribute value x_{jk} of each attribute X_j ($j = 1, \cdots, p;\ k = 1, \cdots, K_j$)

$\hat{P}(X_j = x_{jk} \mid C = c_i) \leftarrow$ estimate $P(X_j = x_{jk} \mid C = c_i)$ with examples in **S**;

Output: conditional probability tables; for $X_j, \mathrm{K}_j \times L$ elements

$K_1 + K_2 + \cdots + K_p$

$\{X_1, X_2, \cdots, X_p\}$

$K_1 \times L + K_2 \times L + K_p \times L$

Naïve Bayes (for discrete input attributes) - testing

- Naïve Bayes Algorithm (for discrete input attributes)
  - Test Phase: Given an unknown instance $\mathbf{X}' = (a_1', \cdots, a_p')$

    Look up tables to assign the label c^* to $\mathbf{X}'$ if

$$[\hat{P}(a_1' \mid c^*) \cdots \hat{P}(a_p' \mid c^*)]\hat{P}(c^*) > [\hat{P}(a_1' \mid c) \cdots \hat{P}(a_p' \mid c)]\hat{P}(c),$$

$$c \neq c^*, \; c = c_1, \cdots, c_L$$

$$P(X' \mid C_i) P(C_i)$$

$$= P(a_1' \mid C_i) P(a_2' \mid C_i) \cdots P(a_p' \mid C_i) P(C_i)$$

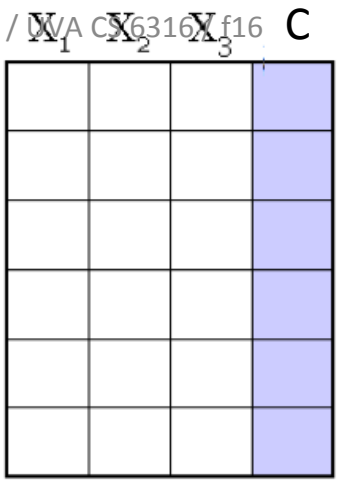
$$i = 1, 2, \cdots, L$$

An Example

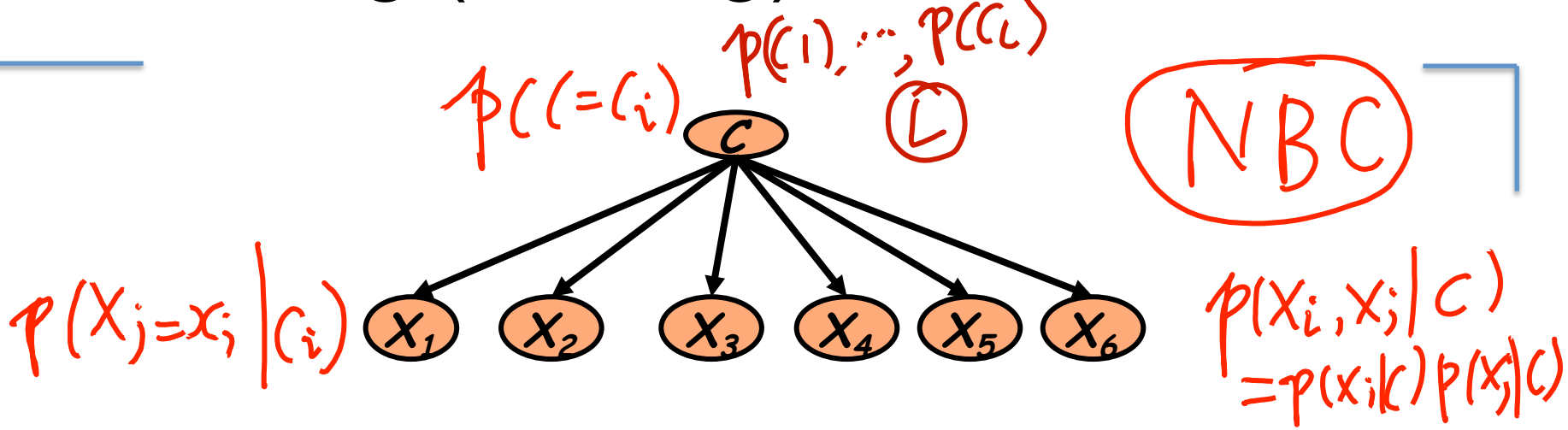
- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Learning (training) the NBC Model



- maximum likelihood estimates (explain later)
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$p(X_1 = \text{Rain} \mid C = \text{Yes}) = \frac{3}{9}$

$p(X_1 = \text{Rain} \mid C = \text{No}) = \frac{2}{5}$

Counting

$P(X_1, X_2, X_3, X_4 | C_i)$

- Learning Phase

Estimate $P(X_j = x_{jk} | C = c_i)$ with examples in training;

$P(X_2|C_1)$, $P(X_2|C_2)$

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

$P(X_4|C_1)$, $P(X_4|C_2)$

Humidity	Play=*Yes*	Play=*No*
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

3+3+2+2 [naïve assumption] * 2 [two classes]= 20 parameters

$P(\text{Play=}Yes) = 9/14$ $P(\text{Play=}No) = 5/14$

$P(C_1)$, $P(C_2)$, …, $P(C_L)$

$P(C_i)$

Testing the NBC Model

look up

$$[\hat{P}(a'_1|c^*)\cdots\hat{P}(a'_p|c^*)]\hat{P}(c^*) > [\hat{P}(a'_1|c)\cdots\hat{P}(a'_p|c)]\hat{P}(c)$$

- Test Phase
  - Given a new instance,

    x' =(Outlook=*Sunny,* Temperature=*Cool,* Humidity=*High,* Wind=*Strong*)

Testing the NBC Model

$$[\hat{P}(a'_1|c^*)\cdots\hat{P}(a'_p|c^*)]\hat{P}(c^*) > [\hat{P}(a'_1|c)\cdots\hat{P}(a'_p|c)]\hat{P}(c)$$

- Test Phase
  - Given a new instance,

    $\mathbf{x}'$ =(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)

→ $P(C_1)P(\text{Sunny}|C_1)P(\text{Cool}|C_1)P(\text{High}|C_1)P(\text{Strong}|C_1)$

$= \frac{9}{14} \times \frac{2}{9} \cdots = $

→ $P(C_2)P(\text{Su}|C_2)P(\text{Co}|C_2)P(\text{hi}|C_2)P(\text{St}|C_2)$

$= \frac{5}{14} \times \frac{3}{5} \times \cdots =$

Testing the NBC Model

$$[\hat{P}(a'_1|c^*)\cdots\hat{P}(a'_p|c^*)]\hat{P}(c^*) > [\hat{P}(a'_1|c)\cdots\hat{P}(a'_p|c)]\hat{P}(c)$$

- Test Phase
  - Given a new instance,

    x' =(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)
  - Look up in conditional-prob tables

P(Outlook=*Sunny* | Play=*Yes*) = 2/9	P(Outlook=*Sunny* | Play=*No*) = 3/5
P(Temperature=*Cool* | Play=*Yes*) = 3/9	P(Temperature=*Cool* | Play==*No*) = 1/5
P(Huminity=*High* | Play=*Yes*) = 3/9	P(Huminity=*High* | Play=*No*) = 4/5
P(Wind=*Strong* | Play=*Yes*) = 3/9	P(Wind=*Strong* | Play=*No*) = 3/5
P(Play=*Yes*) = 9/14	P(Play=*No*) = 5/14

  - MAP rule

P(*Yes* | **x'**): [P(*Sunny* | Yes)P(*Cool* | *Yes*)P(*High* | Yes)P(*Strong* | *Yes*)]P(Play=*Yes*) = 0.0053

P(*No* | **x'**): [P(*Sunny* | No) P(*Cool* | No)P(*High* | *No*)P(*Strong* | *No*)]P(Play=*No*) = 0.0206

Given the fact P(*Yes* | **x'**) < P(*No* | **x'**), we label **x'** to be "*No*".

WHY ? Naïve Bayes Assumption

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.
- $P(x_1,x_2,…,x_p|c_j)$
  - $O(|X_1|\cdot|X_2|\cdot|X_3|\cdot\cdot\cdot\cdot|X_p|\cdot|C|)$ parameters
  - Could only be estimated if a very, very large number of training examples was available.

If no naïve assumption

Adapt From Manning' textCat tutorial

WHY ? Naïve Bayes Assumption

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.

Not Naïve

- $P(x_1,x_2,\ldots,x_p|c_j)$
  - $O(|X_1|.|X_2|.|X_3|\ldots.|X_p|.|C|)$ parameters
  - Could only be estimated if a very, very large number of training examples was available.

Naïve

- $P(x_k|c_j)$
  - $O([|X_1|+|X_2|+|X_3|\ldots.+|X_p|].|C|)$ parameters
  - Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities $P(x_i|c_j)$.

WHY ? Naïve Bayes Assumption

Assuming $|c| = L$ (num of unique values)

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.

Assuming $|X_i| = 2$, $i = 1, 2, \cdots, p$

$\Rightarrow 2^p * L$ (Exp)

Not Naïve

- $P(x_1, x_2, \ldots, x_p | c_j)$
  - $O(|X_1| . |X_2| . |X_3| \ldots . |X_p| . |C|)$ parameters
  - Could only be estimated if a very, very large number of training examples was available.

$(2+2+2+\cdots 2) * L$

$= 2 * p * L$ (Linear)

Naïve

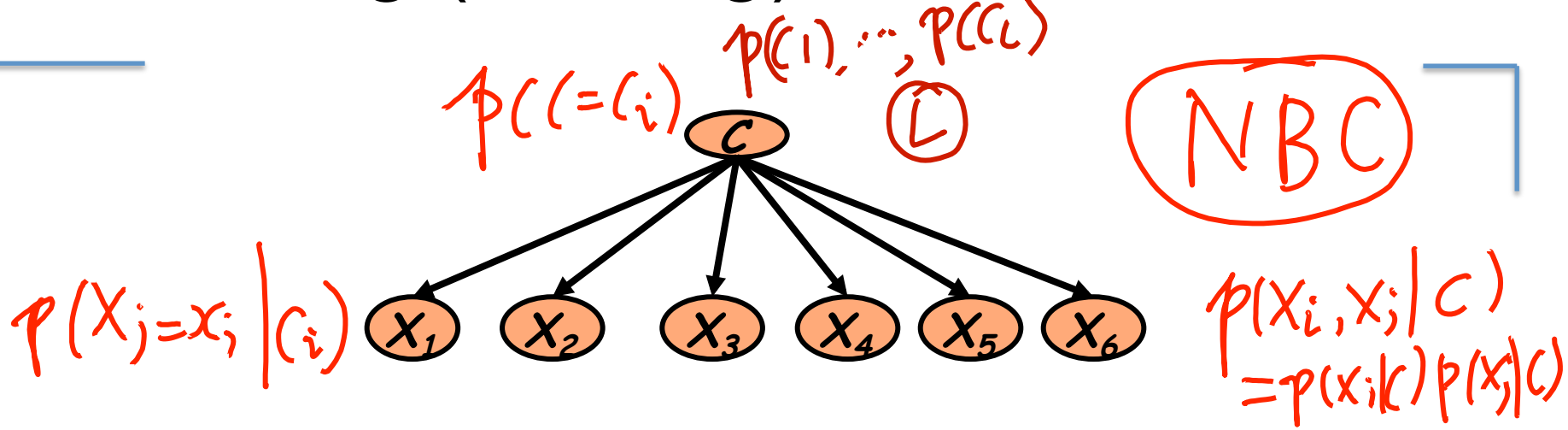
- $P(x_k | c_j)$
  - $O([|X_1| + |X_2| + |X_3| \ldots . + |X_p|] . |C|)$ parameters
  - Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities $P(x_i | c_j)$.

DETOUR: Course Schedule

- Midterm @ WED / In CLASS / 70mins
- Open to your notes + (printed) lecture + Four HWs we had so far
  - Nothing else is allowed
  - Please turn off your phone at the beginning
  - No Electronic Devices (other than basic calculator)

- Final Exam
  - Will be close-note !

Learning (training) the NBC Model

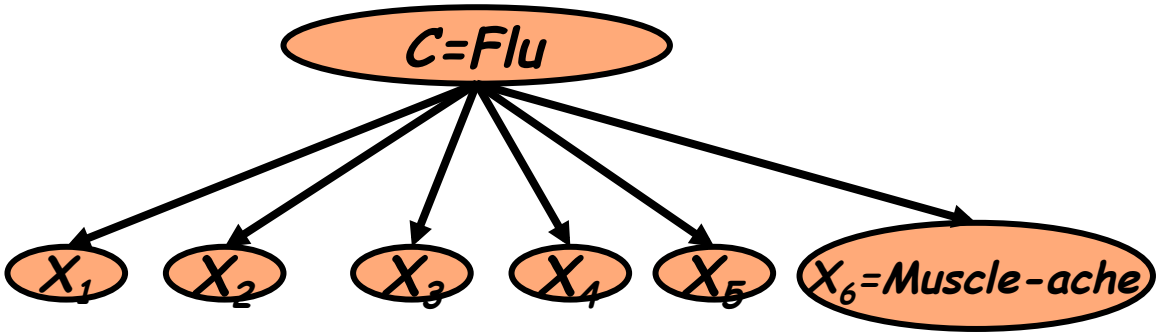


- maximum likelihood estimates (explain later)
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

For instance:



- What if we have seen no training cases where patient had no flu and muscle aches?
- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$\hat{P}(X_6 = t \mid C = not\_flu) = \frac{N(X_6 = t, C = nf)}{N(C = nf)} = 0$$

muscle-ache_Yes/no flu/nf

$$?? = \arg\max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

$$\delta_f = p(c=f)\,p(x_1|f)\,p(x_2|f)\,p(x_3|f)\,p(x_4|f)\,p(x_5|f)\,p(x_6|f)$$

$$\delta_{nf} = p(c=nf)\,p(x_1|nf)\,p(x_2|nf)\,p(x_3|nf)\,p(x_4|nf)\,p(x_5|nf)\,p(x_6|nf)$$

if any term gives 0,

$$\Rightarrow \delta_{nf} = 0$$

no matter other terms' value

Smoothing to Avoid Overfitting

Why necessary ??

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k_i}$$

# of values of feature X_i

To make sum_i (P(xi|Cj)=1

$|X_i| = k_i$

Smoothing to Avoid Overfitting

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k_i}$$

k_i: # of values of X_i

- Somewhat more subtle version

$$\hat{P}(x_{i,k} \mid c_j) = \frac{N(X_i = x_{i,k}, C = c_j) + mp_{i,k}}{N(C = c_j) + m}$$

$p_{i,k}$: overall fraction in data where $X_i = x_{i,k}$

$k \in \{1, 2, \ldots, k_i\}$

m: extent of "smoothing"

Today : Generative Bayes Classifiers

- ✓ Bayes Classifier
  - MAP classification rule
  - Generative Bayes Classifier
- ✓ Naïve Bayes Classifier
- ✓ Gaussian Bayes Classifiers
  - Gaussian distribution
  - Gaussian NBC
  - LDA, QDA

Review: Continuous Random Variables

- Probability density function (pdf) instead of probability mass function (pmf)
  - For discrete RV: Probability mass function (pmf): $P(X = x_i)$

- A pdf (prob. Density func.) is any function $f(x)$ that describes the probability density in terms of the input variable x.

Review: Probability of Continuous RV

- Properties of pdf
  - $f(x) \geq 0, \forall x$
  - $\int_{-\infty}^{+\infty} f(x) = 1$ $\longrightarrow$ $\sum_{i=1}^{k_i} P(X = x_i) = 1$
- Actual probability can be obtained by taking the integral of pdf
  - E.g. the probability of X being between 5 and 6 is

$$P(5 \leq X \leq 6) = \int_{5}^{6} f(x)dx$$

Review: Mean and Variance of RV

- Mean (Expectation): $\mu = E(\mathrm{X})$
  - Discrete RVs: $E(\mathrm{X}) = \sum_{v_i} v_i \mathrm{P}(\mathrm{X} = v_i)$

$$E(g(X)) = \sum_{v_i} g(v_i) P(X = v_i)$$

  - Continuous RVs: $E(\mathrm{X}) = \int_{-\infty}^{+\infty} x f(x) dx$

$$E(g(X)) = \int_{-\infty}^{+\infty} g(x) f(x) dx$$

Adapt From Carols' prob tutorial

Review: Mean and Variance of RV

- Variance: $Var(X) = E((X-\mu)^2)$

  $\sigma_x = \sqrt{V(x)}$

  - Discrete RVs: $V(X) = \sum_{v_i} (v_i - \mu)^2 P(X = v_i)$

  - Continuous RVs: $V(X) = \int_{-\infty}^{+\infty} (x-\mu)^2 f(x) dx$

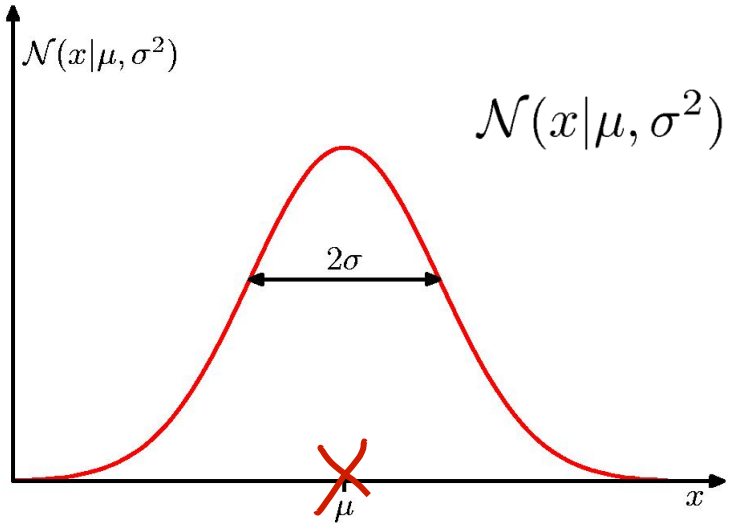
- Covariance:

  Correlation $\rho_{x,y} = Corr(X,Y) = \frac{Cov(X,Y)}{\sigma_x \sigma_y}$

$$Cov(X,Y) = E((X-\mu_x)(Y-\mu_y)) = E(XY) - \mu_x \mu_y$$

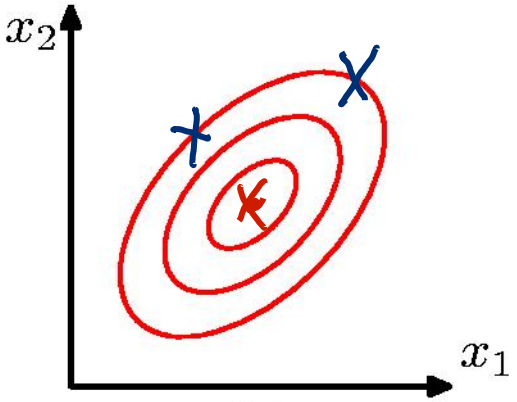
Adapt From Carols' prob tutorial

Gaussian Distribution



$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

$$X \sim \mathcal{N}(\mu,\sigma^2)$$



$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Mean

Covariance Matrix

Courtesy: http://research.microsoft.com/~cmbishop/PRML/index.htm

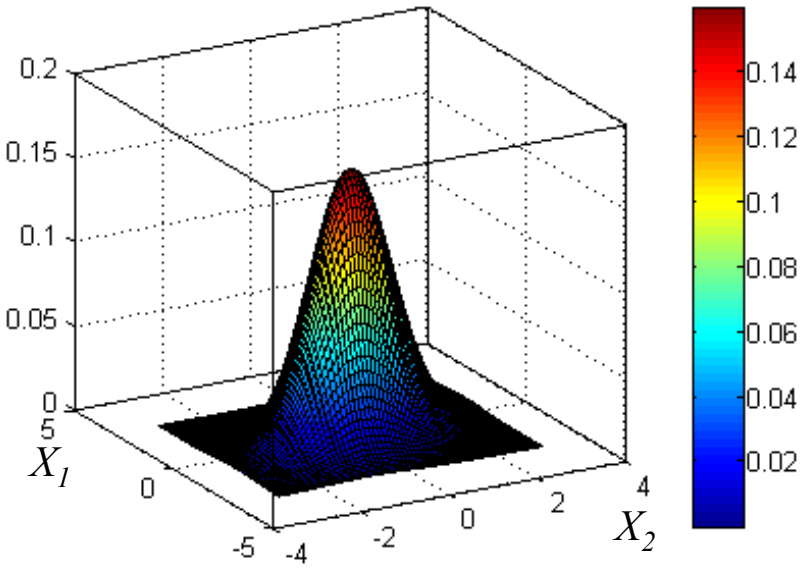
Multivariate Normal (Gaussian) PDFs

The only widely used continuous joint PDF is the multivariate normal (or Gaussian):

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Where |*| represents determinant

Bivariate normal PDF:



- Mean of normal PDF is at peak value. Contours of equal PDF form ellipses.

- The covariance matrix captures linear dependencies among the variables

Example: the Bivariate Normal distribution

$$f\left(x_1, x_2\right) = \frac{1}{\left(2\pi\right)\left|\Sigma\right|^{1/2}} \mathrm{e}^{-\frac{1}{2}\left(\vec{x}-\vec{\mu}\right)^T \Sigma^{-1}\left(\vec{x}-\vec{\mu}\right)}$$

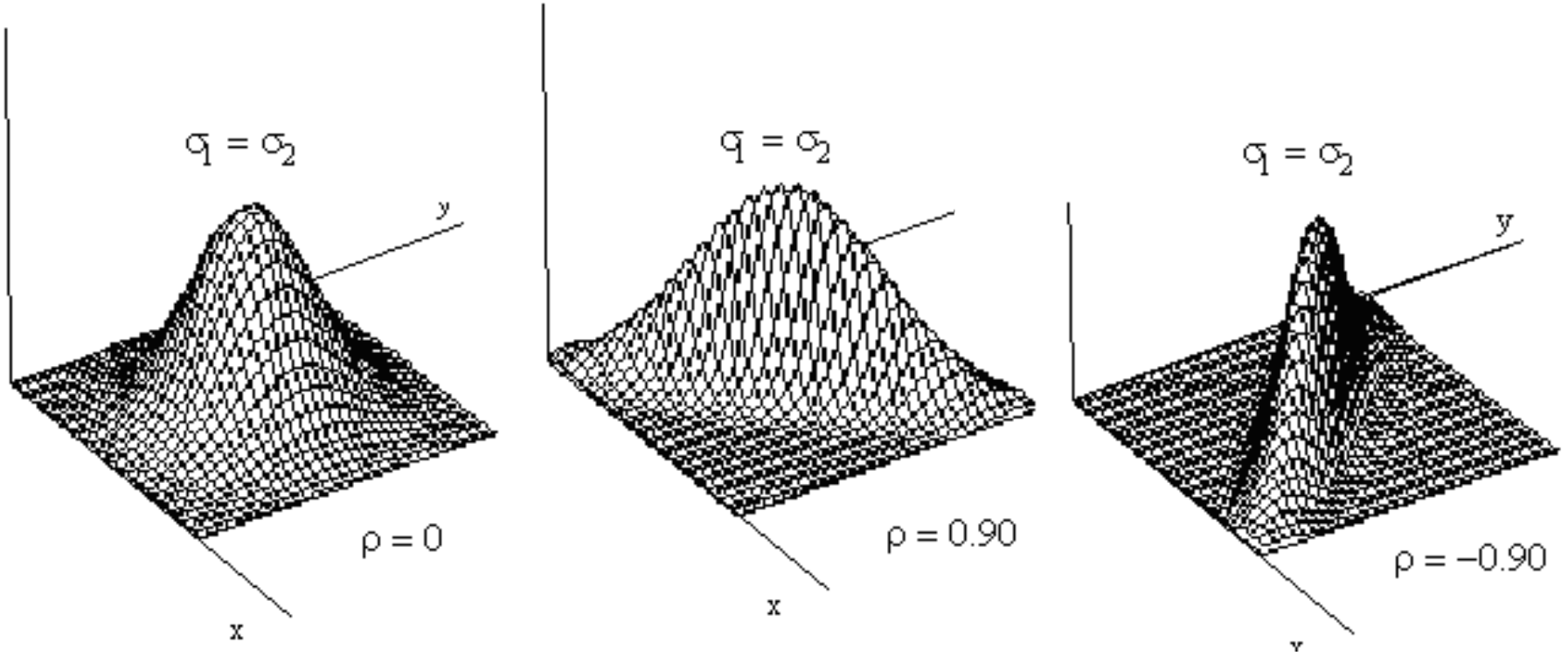
with $\vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ 2x1 and

$$\underset{2\times 2}{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

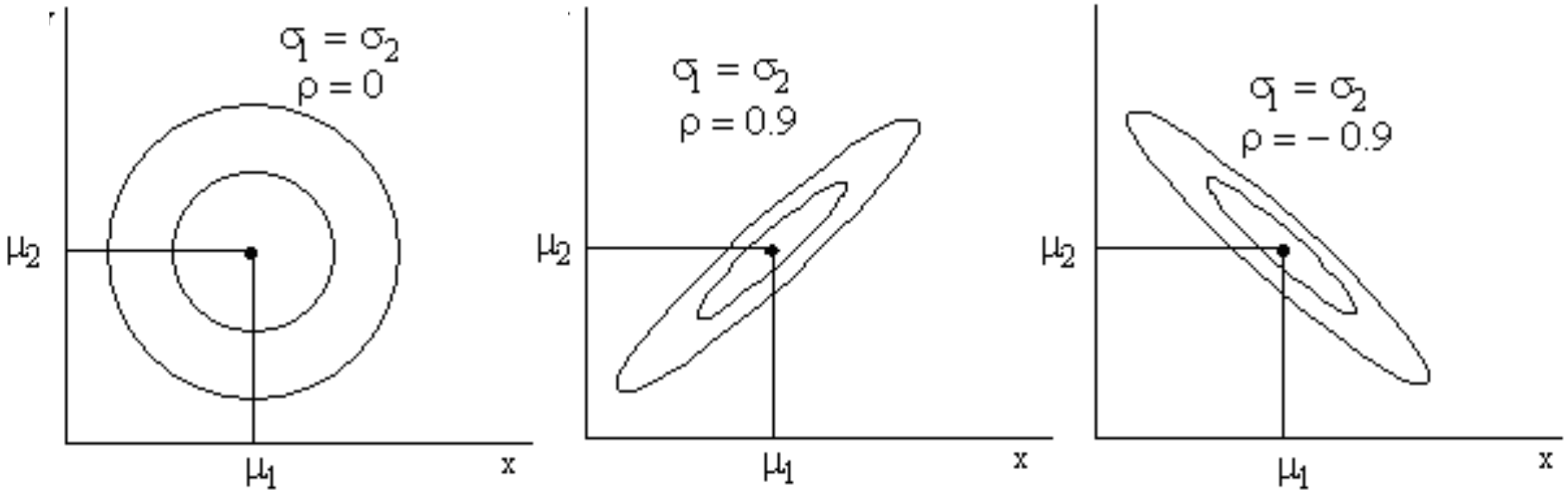
V(X1) Cov(X1, X2) V(X2) 2 x 2

$$\left|\Sigma\right| = \sigma_{11}\sigma_{22} - \sigma_{12}^2 = \sigma_1^2\sigma_2^2\left(1-\rho^2\right)$$

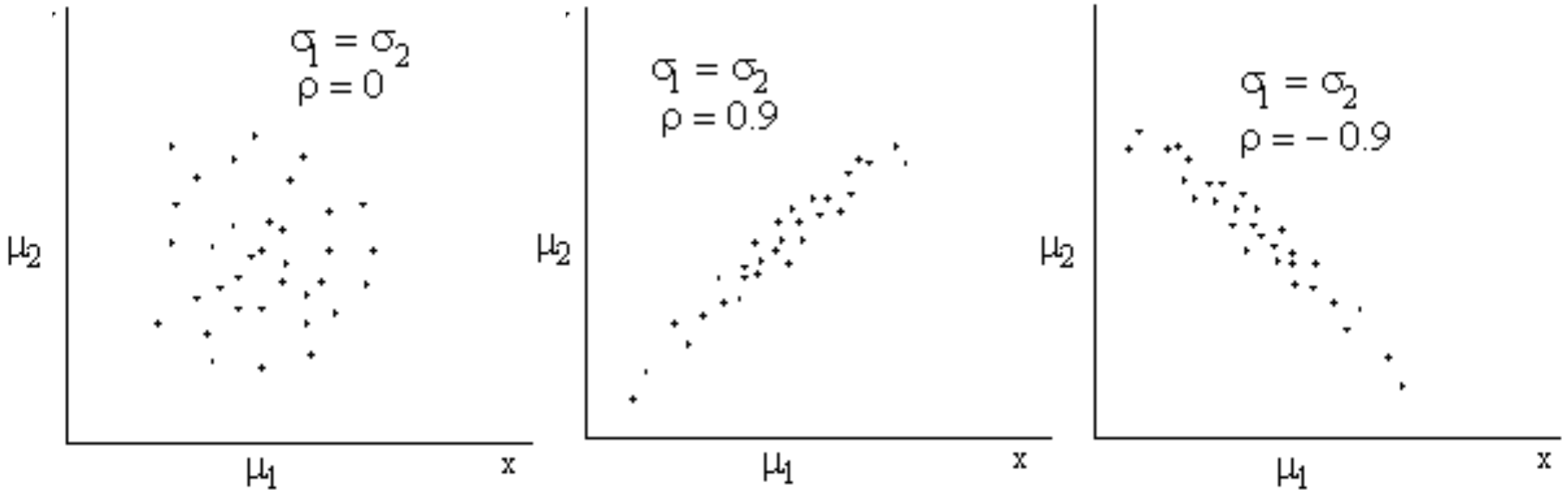
Surface Plots of the bivariate Normal distribution



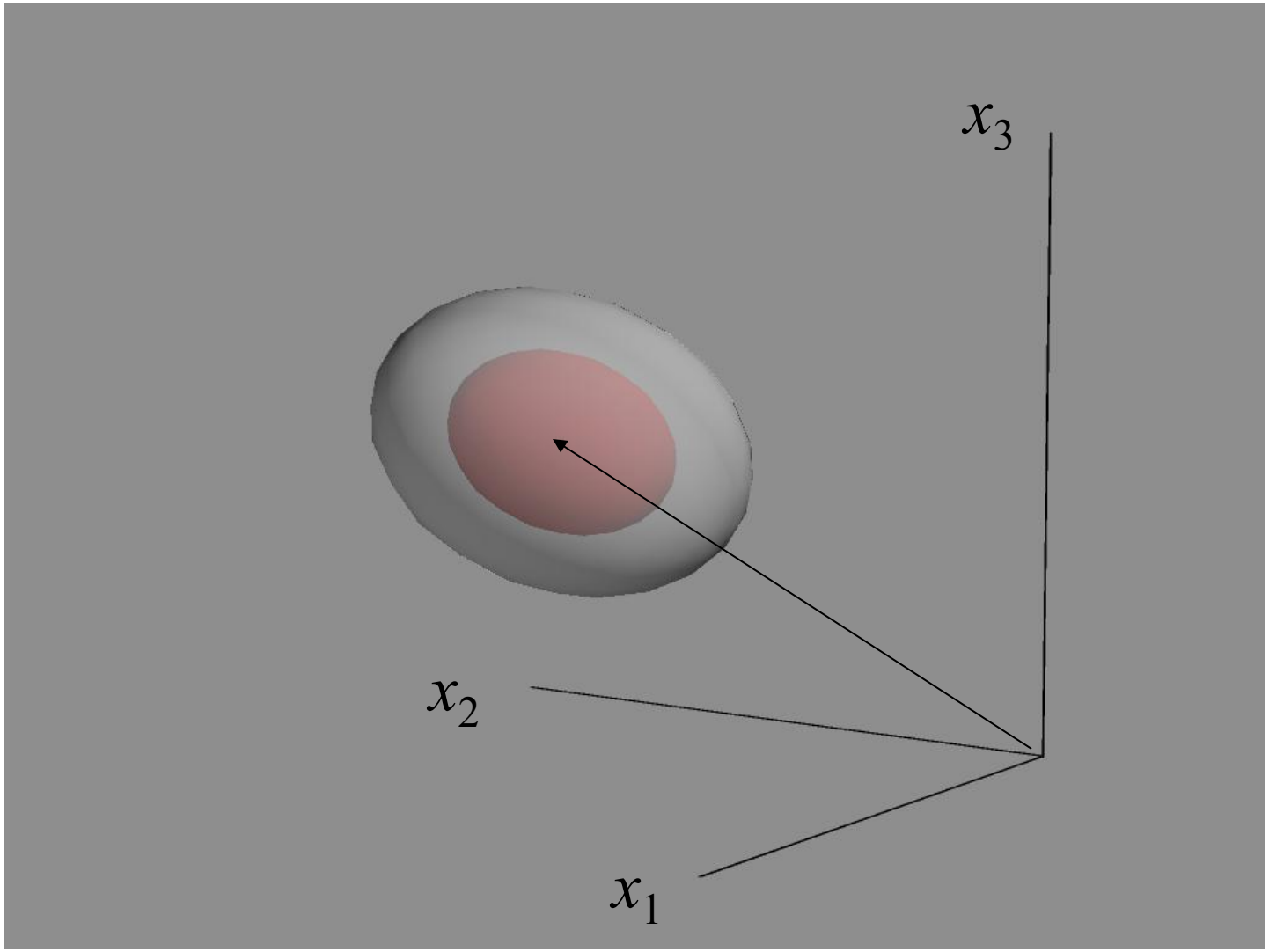
Contour Plots of the bivariate Normal distribution



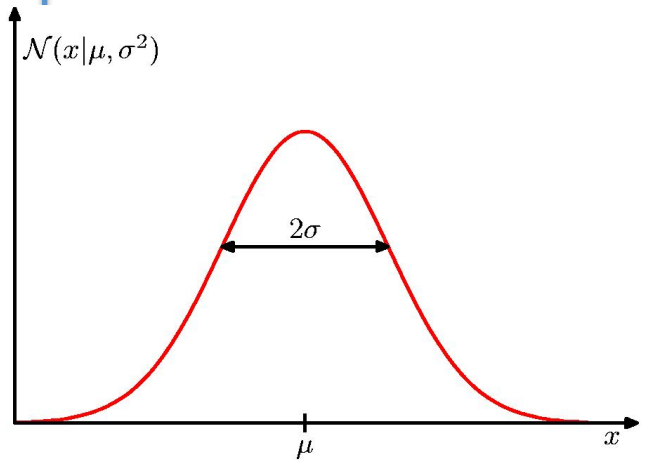
Scatter Plots of data from the bivariate Normal distribution



Trivariate Normal distribution



How to Estimate Gaussian: MLE (Later)



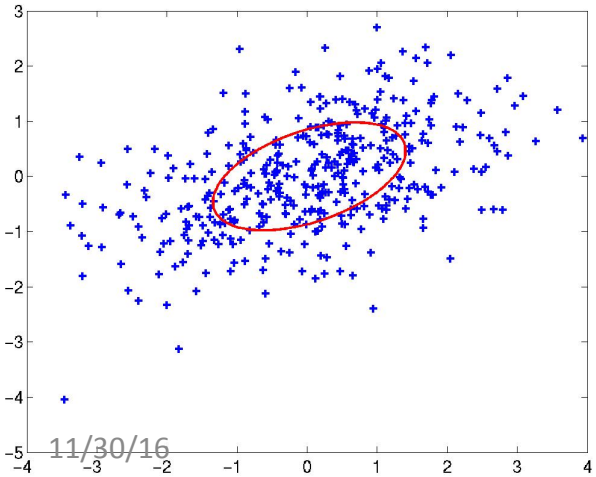
- We can fit statistical models by maximizing the probability / likelihood of generating the observed samples:

L(x$_1$, … ,x$_n$ | \theta) = p(x$_1$ | \theta) … p(x$_n$ | \theta)

(the samples are assumed to be IID)

- In the 1D Gaussian case, we simply set the mean and the variance to the sample mean and the sample variance:

$$\bar{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \bar{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{\mu})^2$$



The p-multivariate Normal distribution

$$< X_1, X_2 \cdots, X_p > \sim N\left(\vec{\mu}, \Sigma\right)$$

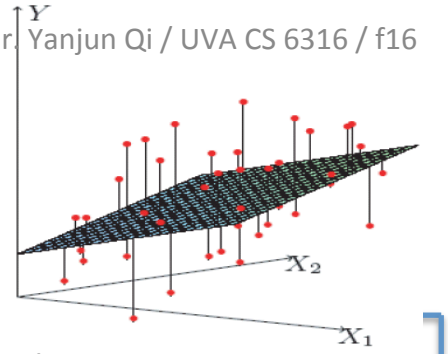
$$\vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}_{p \times 1} \qquad \mu_i = \frac{1}{n} \sum_{j=1}^{N} x_j^{(i)}$$

$i \in \{1, 2, \cdots, p\}$: i-th feature

$j \in \{1, 2, \cdots, N\}$: j-th sample

$$\underset{p \times p}{\Sigma} = \begin{bmatrix} \mathrm{Var}(X_1) & \cdots & \mathrm{Cov}(X_i, X_j) & \cdots \\ & \ddots & & \\ & & \mathrm{Var}(X_p) & \end{bmatrix}$$

DETOUR: Probabilistic Interpretation of Linear Regression

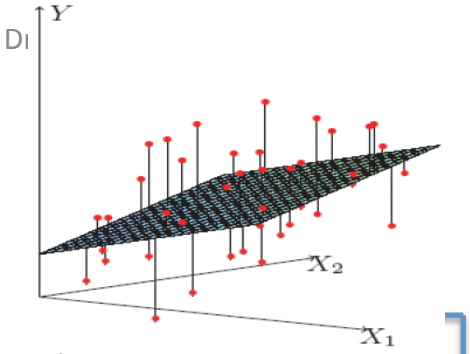


- Let us assume that the target variable and the inputs are related by the equation:

$$y_i = \theta^T \mathbf{x}_i + \varepsilon_i$$

where ε is an error term of unmodeled effects or random noise

DETOUR: Probabilistic Interpretation of Linear Regression



- Let us assume that the target variable and the inputs are related by the equation:

$$y_i = \theta^T \mathbf{x}_i + \varepsilon_i$$

RV $\varepsilon \sim N(0, \sigma^2)$

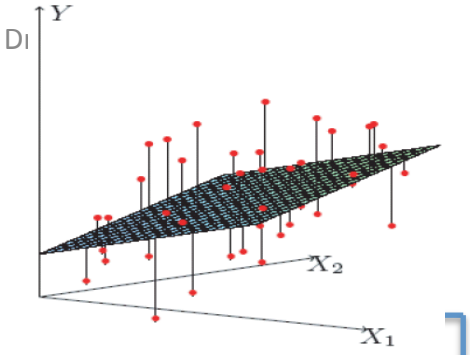
where ε is an error term of unmodeled effects or random noise

- Now assume that ε follows a Gaussian $N(0,\sigma^2)$, then we have:

$$p(y_i \mid x_i; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2} \right)$$

RV $y|x;\theta \sim N(\theta^T x, \sigma)$

DETOUR: Probabilistic Interpretation of Linear Regression



- Let us assume that the target variable and the inputs are related by the equation:

$$y_i = \theta^T \mathbf{x}_i + \varepsilon_i$$

where ε is an error term of unmodeled effects or random noise

- Now assume that ε follows a Gaussian $N(0,\sigma^2)$, then we have:

$$p(y_i \mid x_i;\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_i-\theta^T\mathbf{x}_i)^2}{2\sigma^2}\right)$$

- By IID (independent and identically distributed) assumption:

$$L(\theta) = \prod_{i=1}^{n} p(y_i \mid x_i;\theta) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}(y_i-\theta^T\mathbf{x}_i)^2}{2\sigma^2}\right)$$

$$L(\theta)=\prod_{i=1}^{n} p(y_i \mid x_i;\theta)=\left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}(y_i-\theta^T\mathbf{x}_i)^2}{2\sigma^2}\right)$$

We can learn \theta by maximizing the probability / likelihood of generating the observed samples:

$$p\{(\vec{x}_1,y_1)\wedge(\vec{x}_2,y_2)\wedge\cdots(\vec{x}_N,y_N)\}$$

$$= \prod_{i=1}^{N} p(y_i,\vec{x}_i) \overset{IID}{=} \prod_{i=1}^{N} p(y_i\mid\vec{x}_i;\theta)p(\vec{x}_i)$$

$$\theta^* = \arg\max_{\theta} \prod_{i=1}^{N} p(y_i\mid\vec{x}_i;\theta)$$

$$L(\theta)=\prod_{i=1}^{n} p(y_i \mid x_i;\theta)=\left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}(y_i-\theta^T\mathbf{x}_i)^2}{2\sigma^2}\right)$$

We can learn \theta by maximizing the probability / likelihood of generating the observed samples:

$\varepsilon \sim N(0, \sigma^2)$

$$l(\theta)=\log(L(\theta))=n\log\frac{1}{\sqrt{2\pi\sigma}}-\frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{n}(y_i-\theta^T\mathbf{x}_i)^2$$

$\arg\max_{\theta} l(\theta) \Rightarrow \arg\min_{\theta} \sum_{i=1}^{n}(y_i-\theta^T\vec{x}_i)^2$

SSE

$$L(\theta)=\prod_{i=1}^{n} p(y_i \mid x_i;\theta)=\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}(y_i-\theta^T\mathbf{x}_i)^2}{2\sigma^2}\right)$$

We can learn \theta by maximizing the probability / likelihood of generating the observed samples:

$$l(\theta)=\log(L(\theta))=n\log\frac{1}{\sqrt{2\pi}\sigma}-\frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{n}(y_i-\theta^T\mathbf{x}_i)^2$$

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^T\theta-y_i)^2$$

Maximum Likelihood Estimation

A general Statement

Consider a sample set T=(X_1...X_n) which is drawn from a probability distribution P(X|\theta) where \theta are parameters.

If the Xs are independent with probability density function P(X_i|\theta), the joint probability of the whole set is

$$P(X_1 \cdots X_n | \theta) = \prod_{i=1}^{n} P(X_i | \theta)$$

Maximum Likelihood Estimation

A general Statement

Consider a sample set T=(X_1...X_n) which is drawn from a probability distribution P(X|\theta) where \theta are parameters.

If the Xs are independent with probability density function P(X_i|\theta), the joint probability of the whole set is

$$P(X_1 \cdots X_n|\theta) = \prod_{i=1}^{n} P(X_i|\theta)$$

$L(\theta)$ likelihood

$LL(\theta) = \log(L(\theta))$

➔ This may be maximised with respect to \theta to give the maximum likelihood estimates (MLE) of \theta :

$$\hat{\theta} = \arg\max_{\theta} P(X_1 \cdots X_n|\theta)$$

The idea is to

- ✓ assume a particular model with unknown parameters: θ
- ✓ we can then define the probability of observing a given event conditional on a particular set of parameters. $P(X_i|\theta)$
- ✓ We have observed a set of outcomes in the real world. $x_1, x_2, \ldots, x_n$

The idea is to

- ✓ assume a particular model with unknown parameters, θ
- ✓ we can then define the probability of observing a given event conditional on a particular set of parameters. $P(X_i|\theta)$
- ✓ We have observed a set of outcomes in the real world. $x_1, x_2, \ldots, x_n$
- ✓ It is then possible to choose a set of parameters which are most likely to have produced the observed results.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} P(X_1 \ldots X_n / \theta)$$

likehood

The idea is to

- ✓ assume a particular model with unknown parameters, θ
- ✓ we can then define the probability of observing a given event conditional on a particular set of parameters. $P(X_i|\theta)$
- ✓ We have observed a set of outcomes in the real world. $x_1, x_2, \ldots, x_n$
- ✓ It is then possible to choose a set of parameters which are most likely to have produced the observed results.

$$\hat{\theta} = \arg\max_{\theta} P(X_1 \ldots X_n / \theta)$$

likelihood

This is maximum likelihood. In most cases it is both consistent and efficient. It provides a standard to compare other estimation techniques.

$$\log(L(\theta)) = \sum_{i=1}^{n} \log(P(X_i / \theta))$$

log likelihood

It is often convenient to work with the Log of the likelihood function.

DETOUR: Probabilistic Interpretation of Linear Regression

- Hence the log-likelihood is:

$$l(\theta) = \log(L(\theta)) = n\log\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{n}(y_i - \theta^T \mathbf{x}_i)^2$$

- Recognize the last term?

Yes it is: $$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^T\theta - y_i)^2$$

- Thus under independence Gaussian residual assumption, residual square error is equivalent to MLE of ϑ !

$y_i \sim N(exp(wx_i), 1)$

(b) (6 points) (no explanation required) Suppose you decide to do a maximum likelihood estimation of w. You do the math and figure out that you need w to satisfy one of the following equations. Which one?

A. $\Sigma_i x_i exp(wx_i) = \Sigma_i x_i y_i exp(wx_i)$
B. $\Sigma_i x_i exp(2wx_i) = \Sigma_i x_i y_i exp(wx_i)$
C. $\Sigma_i x_i^2 exp(wx_i) = \Sigma_i x_i y_i exp(wx_i)$
D. $\Sigma_i x_i^2 exp(wx_i) = \Sigma_i x_i y_i exp(wx_i/2)$
E. $\Sigma_i exp(wx_i) = \Sigma_i y_i exp(wx_i)$

$y_i \sim N(exp(wx_i), 1)$

Answer: B (this is an extra credit question.)

$L(\theta)$

$\downarrow$

$LL(\theta)$

$\downarrow$

$\frac{\partial LL(\theta)}{\partial \theta} = 0 \Rightarrow (B)$

Today : Generative Bayes Classifiers

- ✓ Bayes Classifier
  - MAP classification rule
  - Generative Bayes Classifier
- ✓ Naïve Bayes Classifier
- ✓ Gaussian Bayes Classifiers
  - Gaussian distribution
  - Gaussian NBC
  - Not-naïve Gaussian BC ➔ LDA, QDA

Gaussian Naïve Bayes Classifier

$$\underset{C}{\operatorname{argmax}} P(C \mid X) = \underset{C}{\operatorname{argmax}} P(X, C) = \underset{C}{\operatorname{argmax}} P(X \mid C) P(C)$$

Naïve Bayes Classifier

$$\begin{aligned} P(X \mid C) &= P(X_1, X_2, \cdots, X_p \mid C) \\ &= P(X_1 \mid X_2, \cdots, X_p, C) P(X_2, \cdots, X_p \mid C) \\ &= P(X_1 \mid C) P(X_2, \cdots, X_p \mid C) \\ &= P(X_1 \mid C) P(X_2 \mid C) \cdots P(X_p \mid C) \end{aligned}$$

$$\hat{P}(X_j \mid C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2} \right)$$

$N(\mu_{ji}, \sigma_{ji}^2)$

μ_{ji} : mean (avearage) of attribute values X_j of examples for which $C = c_i$

σ_{ji} : standard deviation of attribute values X_j of examples for which $C = c_i$

Gaussian Naïve Bayes Classifier

- Continuous-valued Input Attributes
  - Conditional probability modeled with the normal distribution

$$\hat{P}(X_j \mid C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

μ_{ji} : mean (avearage) of attribute values X_j of examples for which $C = c_i$

σ_{ji} : standard deviation of attribute values X_j of examples for which $C = c_i$

  - Learning Phase: for $\mathbf{X} = (X_1, \cdots, X_p)$, $C = c_1, \cdots, c_L$
    Output: $p \times L$ normal distributions and $P(C = c_i)$ $i = 1, \cdots, L$

$\{\mu_{ji}, \sigma_{ji}\}$ → $j \in \{1, 2, \cdots, p\}$, $i \in \{1, 2, \cdots, L\}$

MLE { sample mean, sample variance }

Gaussian Naïve Bayes Classifier

- Continuous-valued Input Attributes
  - Conditional probability modeled with the normal distribution

$$\hat{P}(X_j \mid C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

μ_{ji} : mean (avearage) of attribute values X_j of examples for which $C = c_i$

σ_{ji} : standard deviation of attribute values X_j of examples for which $C = c_i$

  - Learning Phase: for $\mathbf{X} = (X_1, \cdots, X_p)$, $C = c_1, \cdots, c_L$
    Output: $p \times L$ normal distributions and $P(C = c_i)$ $i = 1, \cdots, L$

  - Test Phase: for $\mathbf{X}' = (X_1', \cdots, X_p')$
    - Calculate conditional probabilities with all the normal distributions
    - Apply the MAP rule to make a decision

$$\underset{i}{\operatorname{argmax}}\ P(C=(i)) P(X_1|(i)) \cdots P(X_p|(i))$$

Naïve Gaussian means ?

Not Naïve

$$P(X_1, X_2, \cdots, X_p \mid C) =$$

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Total # para $\Rightarrow L \times \{p + p \times p\}$; μ/c ; Σ/c

Naïve

$$P(X_1, X_2, \cdots, X_p \mid C = c_j) = P(X_1 \mid C) P(X_2 \mid C) \cdots P(X_p \mid C)$$

$$= \prod_i \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

Total # para $\Rightarrow L \times (p + p)$

$\Sigma/c_i = \begin{bmatrix} \sigma_{1i} & & & \\ & \sigma_{2i} & & \\ & & \ddots & \\ & & & \sigma_{pi} \end{bmatrix}$

Diagonal Matrix

$$\Sigma\_c_k = \Lambda\_c_k$$

Each class' covariance matrix is diagonal

Today : Generative Bayes Classifiers

- ✓ Bayes Classifier
  - MAP classification rule
  - Generative Bayes Classifier
- ✓ Naïve Bayes Classifier
- ✓ Gaussian Bayes Classifiers
  - Gaussian distribution
  - Gaussian NBC
  - Not-naïve Gaussian BC → LDA, QDA

(1) covariance matrix are the same across classes

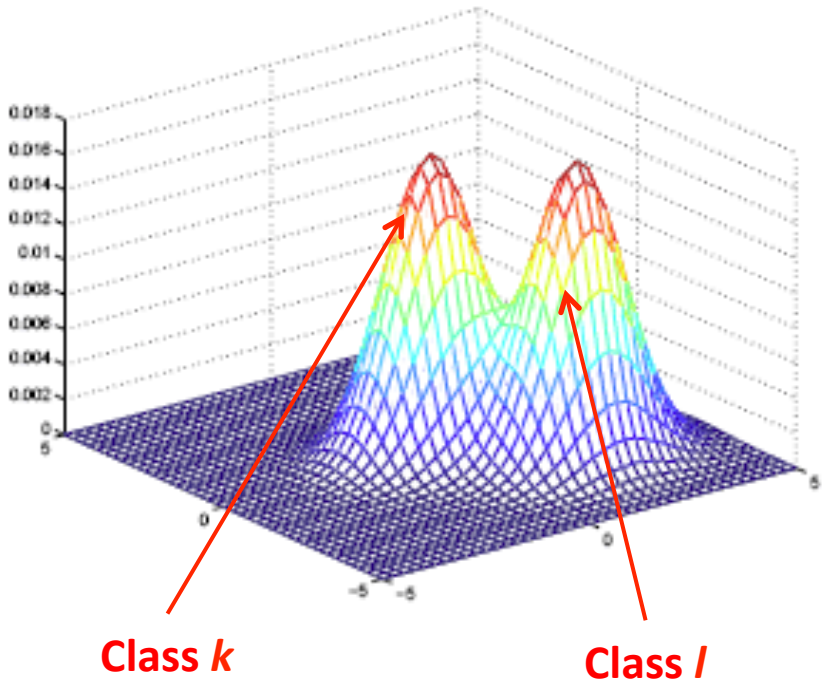
➔ LDA (Linear Discriminant Analysis)

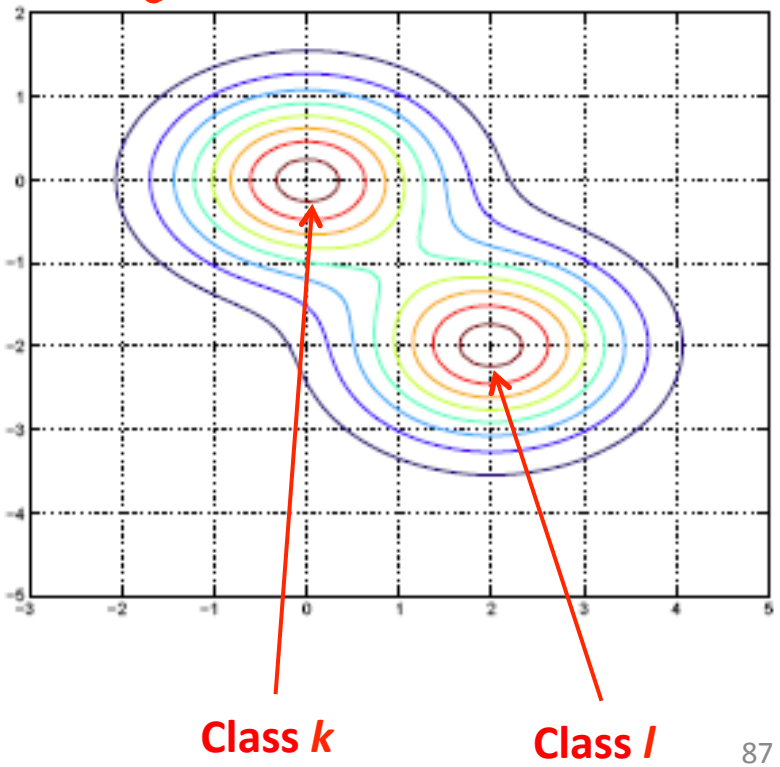
Linear Discriminant Analysis : $\Sigma_k = \Sigma, \ \forall k$ (p×p)

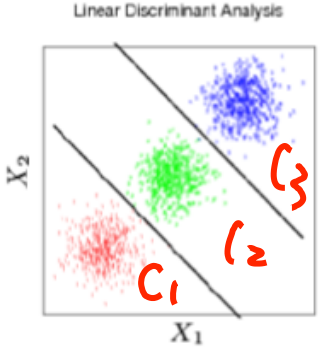
Each class' covariance matrix is the same

The Gaussian Distribution are shifted versions of each other

$k \in \{1, 2, \ldots, L\}$







Optimal Classification

$k \in \{1, 2, \cdots, L\}$

$$\underset{k}{\operatorname{argmax}}\, P(C\_k \mid X) = \underset{k}{\operatorname{argmax}}\, P(X, C_k) = \underset{k}{\operatorname{argmax}}\, P(X \mid C_k) P(C_k)$$

$$= \arg\max_k \left[-\log((2\pi)^{p/2}|\Sigma|^{1/2}) - \frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) + \log(\pi_k) \right]$$

Short for $\pi_k = P(C_k)$, $k \in \{1, 2, \cdots, L\}$

$$= \arg\max_k \left[-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) + \log(\pi_k) \right]$$

- Note

Linear Discriminant Function for LDA

$$-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k - \frac{1}{2}x^T\Sigma^{-1}x$$

$$\underset{k}{\operatorname{argmax}} P(C_k \mid X) = \underset{k}{\operatorname{argmax}} P(X, C_k) = \underset{k}{\operatorname{argmax}} P(X \mid C_k) P(C_k)$$

$$= \underset{k}{\operatorname{argmax}} \log\{P(X \mid C_k) P(C_k)\}$$

Decision Boundary means those points

Satisfying: $p(C_i \mid X) = p(C_j \mid X)$

$$\frac{p(C_i \mid X)}{p(C_j \mid X)} = 1$$

$$\Rightarrow \log \frac{p(C_i \mid X)}{p(C_j \mid X)} = 0$$

$$\underset{k}{\text{argmax}}\, P(C_k \mid X) = \underset{k}{\text{argmax}}\, P(X, C_k) = \underset{k}{\text{argmax}}\, P(X \mid C_k) P(C_k)$$

$$= \underset{k}{\text{argmax}} \log\{P(X \mid C_k) P(C_k)\}$$

$$= \underset{k}{\text{argmax}}\ \log P(X \mid C_k) + \log P(C_k) \Rightarrow \pi_k$$

Decision Boundary points

$$\log \frac{P(C_k \mid X)}{P(C_l \mid X)} = 0 = \log \frac{P(X \mid C_k)}{P(X \mid C_l)} + \log \frac{\pi_k}{\pi_l}$$

$$= \log P(X \mid C_k) - \log P(X \mid C_l) + \log \frac{\pi_k}{\pi_l}$$

$$\log \frac{P(C_k \mid X)}{P(C_l \mid X)} = \log \frac{P(X \mid C_k)}{P(X \mid C_l)} + \log \frac{P(C_k)}{P(C_l)}$$

$$= \log \frac{\pi_k}{\pi_\ell} - \frac{1}{2}(\mu_k + \mu_\ell)^T \Sigma^{-1}(\mu_k - \mu_\ell) \quad (4.9)$$

$$+ x^T \Sigma^{-1}(\mu_k - \mu_\ell) = 0$$

b

a

$\Rightarrow$ $x^T a + b = 0 \Rightarrow$ a linear line decision boundary

Define Linear Discriminant Function

$$\delta(x) = -\frac{1}{2}(x-\mu_k)^T \sum^{-1} (x-\mu_k) + log\pi_k$$

➔ The Decision Boundary Between class *k* and *l*, $\{x : \delta_k(x) = \delta_l(x)\}$, is linear

$$= \log\frac{\pi_k}{\pi_\ell} - \frac{1}{2}(\mu_k+\mu_\ell)^T\Sigma^{-1}(\mu_k-\mu_\ell) \qquad (4.9)$$
$$+ x^T\Sigma^{-1}(\mu_k-\mu_\ell),$$

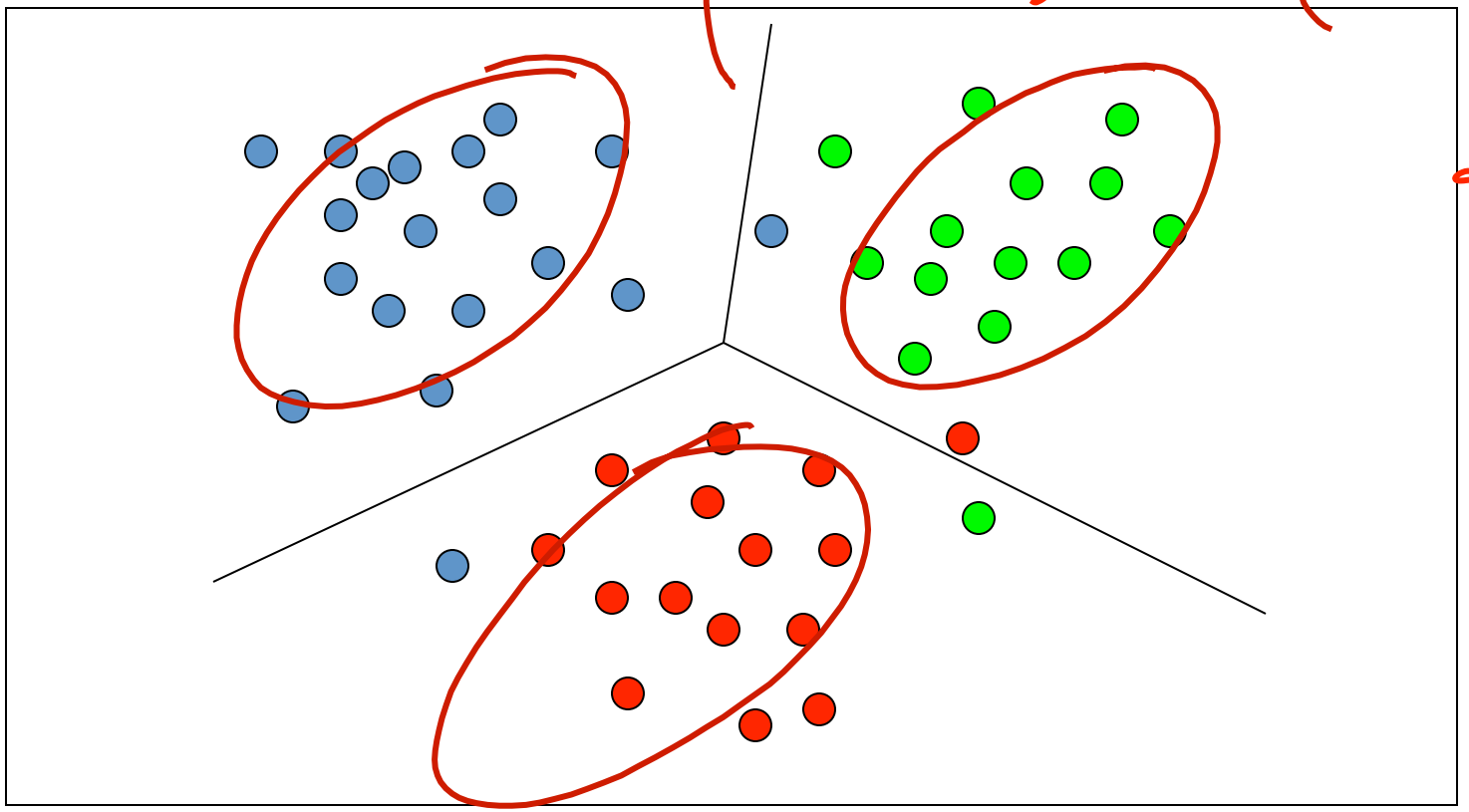
Equals to zero

Boundary points X :

when P(c_k|X) == P(c_l|X),

the left linear equation ==0, a linear line / plane

Visualization (three classes)



(2) If covariance matrix are not same

e.g. → QDA (Quadratic Discriminant Analysis)

- Estimate the covariance matrix Σ_k separately for each class k, $k = 1, 2, ..., K$.
- *Quadratic discriminant function*:

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k \ .$$

$\{\Sigma_1, \Sigma_2, .. \Sigma_K, \mu_1, \mu_2, .. \mu_K\}$

- Classification rule:

$$\hat{G}(x) = \arg\max_k \delta_k(x) \ .$$

$\delta_k(x) - \delta_\ell(x) = 0$

Total # para

$K(p + p^2)$

$\{\mu_k, \Sigma_k\}$

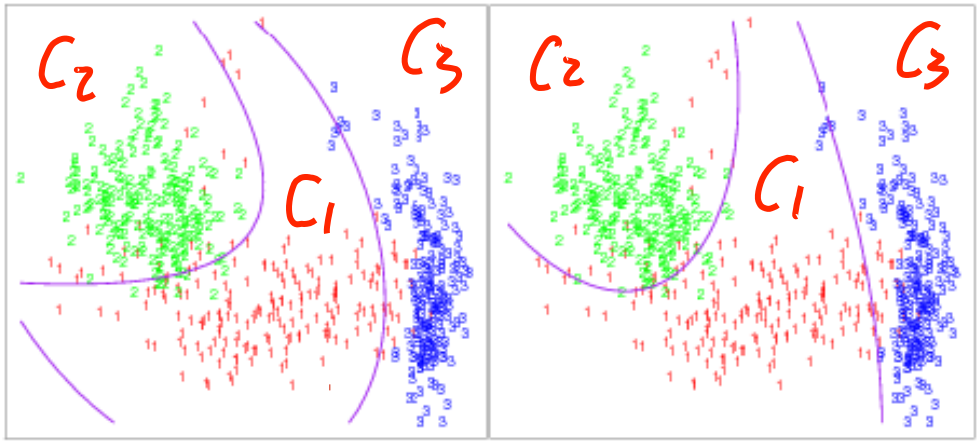
- Decision boundaries are quadratic equations in x.
- QDA fits the data better than LDA, but has more parameters to estimate.

LDA on Expanded Basis

- Expand input space to include X_1X_2, X_1^2, and X_2^2.
- Input is five dimensional: $X = (X_1, X_2, X_1X_2, X_1^2, X_2^2)$.

Q(x) ← LDA

LDA with Q(X)



QDA

LDA with quadratic basis Versus QDA

Figure 4.6: *Two methods for fitting quadratic boundaries. The left plot shows the quadratic decision boundaries for the data in Figure 4.1 (obtained using LDA in the five-dimensional space $x_1, x_2, x_{12}, x_1^2, x_2^2$). The right plot shows the quadratic decision boundaries found by QDA. The differences are small, as is usually the case.*

Both with Quadratic decision Boundary

(3) Regularized Discriminant Analysis

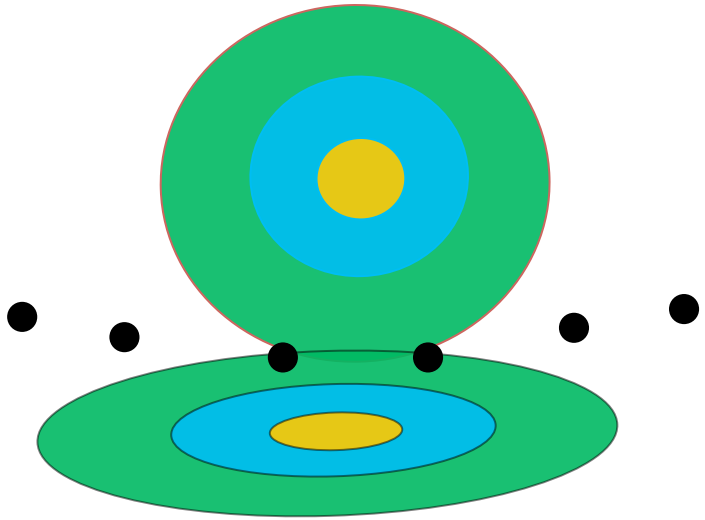
- A compromise between LDA and QDA.
- Shrink the separate covariances of QDA toward a common covariance as in LDA.
- Regularized covariance matrices:

$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma} .$$

- The quadratic discriminant function $\delta_k(x)$ is defined using the shrunken covariance matrices $\hat{\Sigma}_k(\alpha)$.
- The parameter α controls the complexity of the model.

An example: Gaussian Bayes Classifier

Naive BC



$\Sigma_1 \neq \Sigma_2$

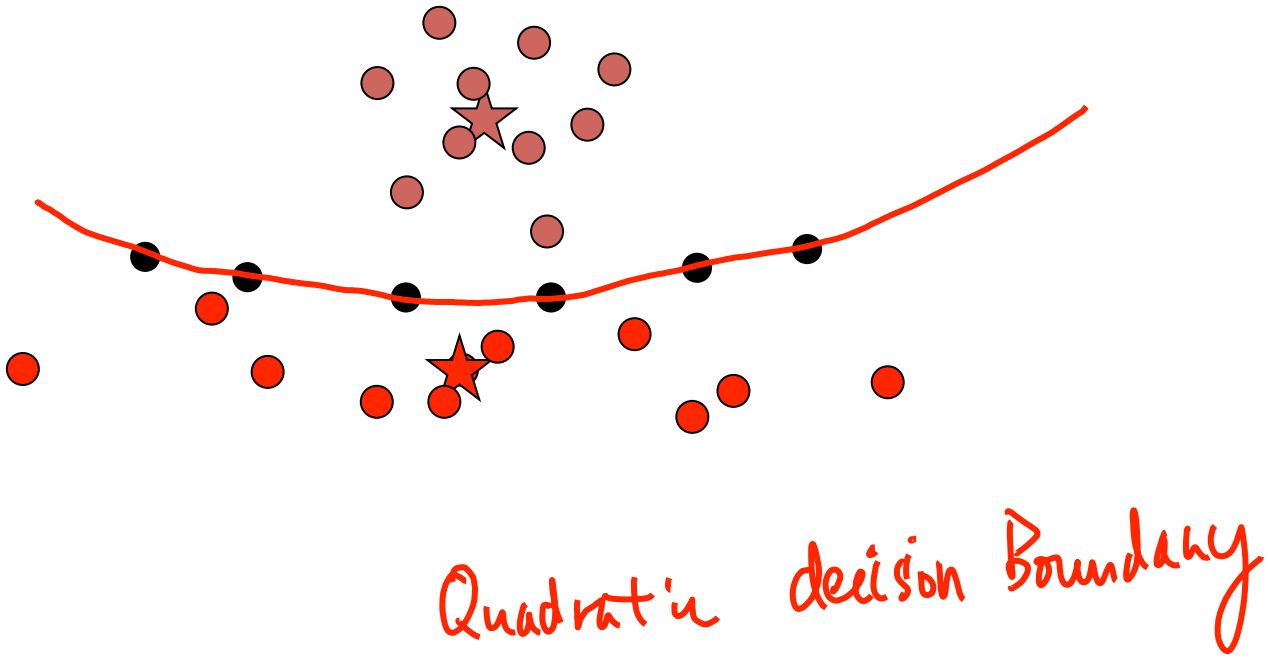
diagonal

$\Sigma_1 = \Lambda_1$

$\Sigma_2 = \Lambda_2$

diagonal

Gaussian Bayes Classifier

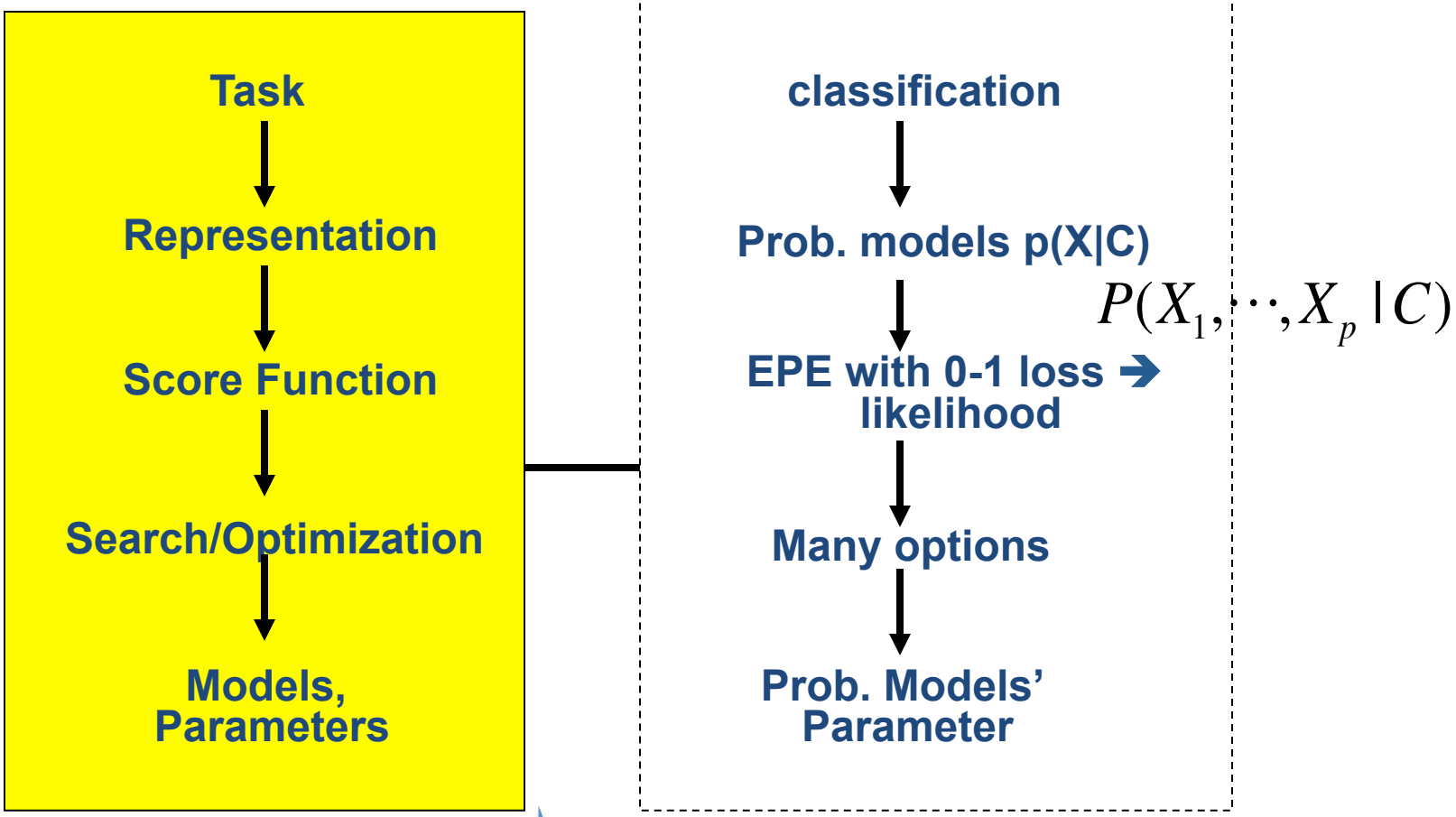


Today Recap : Generative Bayes Classifiers

- ✓ Bayes Classifier
  - MAP classification rule
  - Generative Bayes Classifier
- ✓ Naïve Bayes Classifier
- ✓ Gaussian Naïve Bayes Classifiers
  - Gaussian distribution
  - Gaussian NBC
  - Not-naïve Gaussian BC → LDA, QDA

$$\underset{k}{\operatorname{argmax}} P(C\_k \mid X) = \underset{k}{\operatorname{argmax}} P(X,C) = \underset{k}{\operatorname{argmax}} P(X \mid C)P(C)$$

Generative Bayes Classifier



Bernoulli Naïve

$$p(W_i = true \mid c_k) = p_{i,k}$$

Gaussian Naive

$$\hat{P}(X_j \mid C = c_k) = \frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp\left(-\frac{(X_j - \mu_{jk})^2}{2\sigma_{jk}^2}\right)$$

Multinomial

$$P(W_1 = n_1,\ldots,W_v = n_v \mid c_k) = \frac{N!}{n_{1k}!n_{2k}!..n_{vk}!}\theta_{1k}^{n_{1k}}\theta_{2k}^{n_{2k}}..\theta_{vk}^{n_{vk}}$$

References

- Prof. Andrew Moore's review tutorial
- Prof. Ke Chen NB slides
- Prof. Carlos Guestrin recitation slides