

# Generative Models

Data Intelligence and Learning ([DIAL](#)) Lab

Prof. Jongwuk Lee



# Generative vs. Discriminative Models

# Generative vs. Discriminative



## ➤ Discriminative approach

- ◆ **How do we separate classes?**
  - Estimate parameters of a decision boundary from labeled samples.
- ◆ Requires only a model for the **conditional probability**  $p(y | \mathbf{x})$ .
- ◆ Maximizes the **conditional likelihood**  $\sum_i \log p(y_i | \mathbf{x}_i)$ .
- ◆ Model to learn mapping directly from feature space to the labels.

## ➤ Generative approach

- ◆ **What does the distribution of each class look like?**
  - Estimate the distribution of the characteristics of each class.
- ◆ Model the **joint probability**  $p(\mathbf{x}, y)$  and thus maximizes the **joint likelihood**  $\sum_i \log p(\mathbf{x}_i, y_i)$ .
- ◆ The generative models learn both  $p(\mathbf{x} | y)$  and  $p(y)$ .

# Discriminative Model

- The classifiers involve estimating  $f: \mathbf{x} \rightarrow \mathbf{y}$  or  $p(\mathbf{y} | \mathbf{x})$ .
- Assume a functional form for  $p(\mathbf{y} | \mathbf{x})$ .
- Estimate parameters of  $p(\mathbf{y} | \mathbf{x})$  **directly** from training data.



$$P(Y = \text{Bedroom} | X = \text{[Bedroom Image]}) = 0.8$$

$$P(Y = \text{Dining room} | X = \text{[Bedroom Image]}) = 0.2$$

$$P(Y = \text{Bedroom} | X = \text{[Dining Room Image]}) = 0.4$$

$$P(Y = \text{Dining room} | X = \text{[Dining Room Image]}) = 0.6$$

# Generative Model

- The classifiers involve estimating  $f: \mathbf{x} \rightarrow \mathbf{y}$  or  $p(\mathbf{y} | \mathbf{x})$ .
- Assume a functional form for  $p(\mathbf{y})$  and  $p(\mathbf{x} | \mathbf{y})$ .
- Estimate parameters of  $p(\mathbf{x} | \mathbf{y})$  and  $p(\mathbf{y})$  to learn the distribution of training data.



$$P(Y = \text{Bedroom}, X = \text{Bedroom}) = 0.5$$

$$P(Y = \text{Bedroom}, X = \text{Dining room}) = 0.02$$

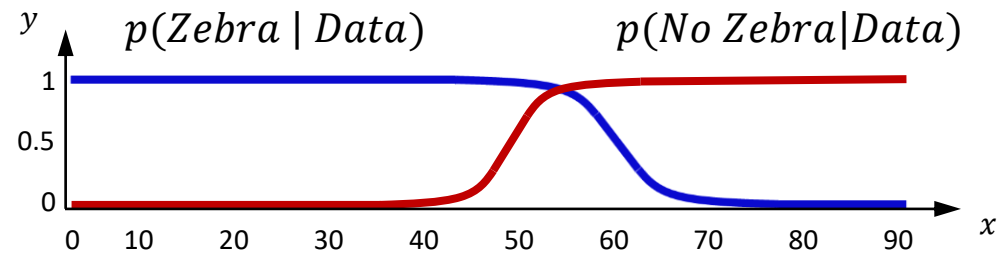
$$P(Y = \text{Dining room}, X = \text{Bedroom}) = 0.01$$

$$P(Y = \text{Dining room}, X = \text{Dining room}) = 0.3$$

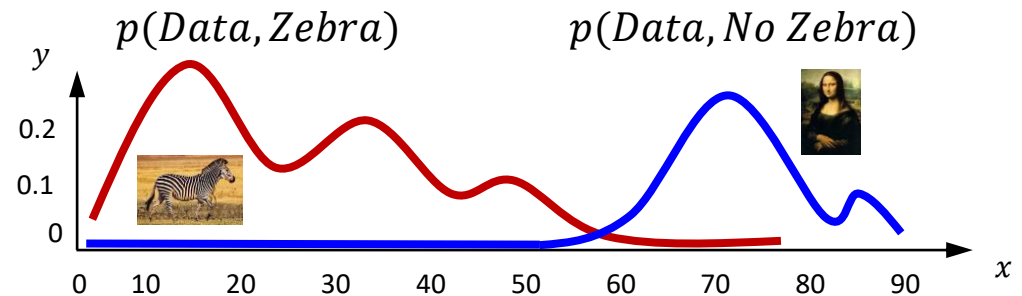
# Discriminative vs. Generative Models



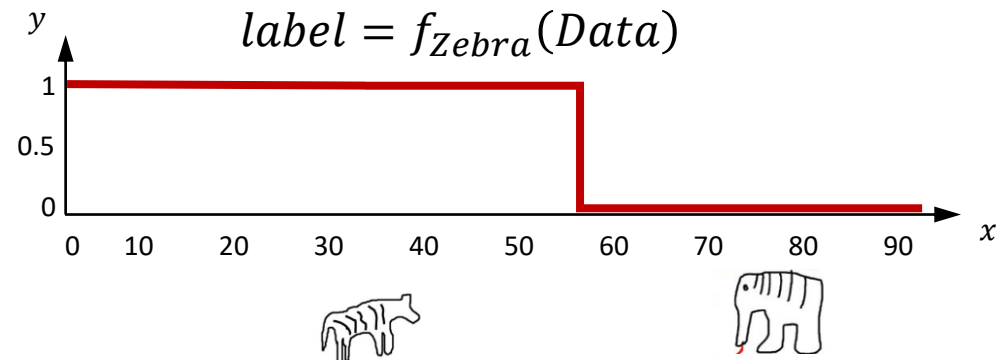
## ➤ Discriminative models



## ➤ Generative models

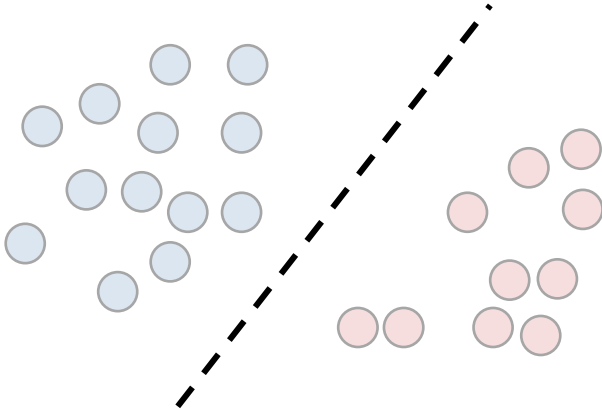
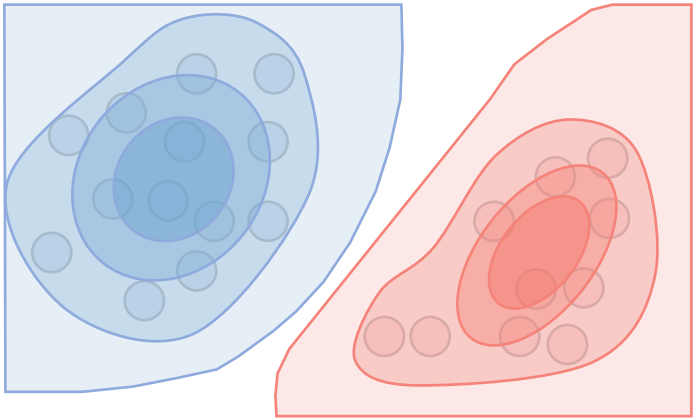


## ➤ Classification function



# Discriminative vs. Generative Models



	Discriminative models	Generative models
Goal	Directly estimate $p(y   \mathbf{x})$	Estimate $p(\mathbf{x}, y)$ to then deduce $p(y   \mathbf{x})$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Logistic regression, decision trees, neural networks, SVM	Gaussian discriminative analysis, Naïve Bayes



# Naïve Bayes Classifier



# Review: Bayes' Theorem

- **Posterior**  $p(y | x)$  is the probability of output  $y$  given  $x$ .
- **Likelihood**  $p(x | y)$  is the function of  $y$  given fixed  $x$ .
- **Prior**  $p(y)$  encapsulates our **subjective prior knowledge of the output**  $y$  before observing any data.

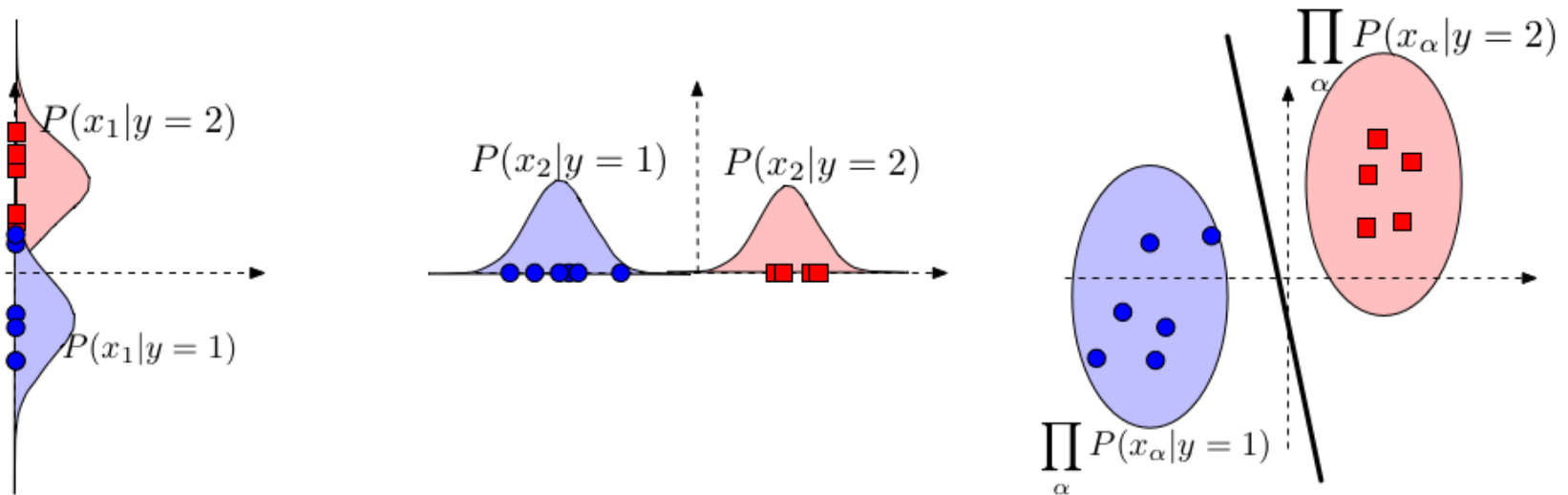
$$\begin{array}{ccc} & \text{Likelihood} & \text{Prior} \\ & P(x | y) & P(y) \\ P(y | x) = & \frac{P(x | y)P(y)}{p(x)} \\ \text{Posterior} & \text{Evidence} & \end{array}$$

# Bayes Classifiers

## ➤ Statistical classifiers

- ◆ Perform probabilistic prediction, i.e., **predicts class membership probabilities**.
- ◆ Based on **Bayes' Theorem**

➤ The **Naïve Bayes classifier** is a simple Bayesian classifier.



# Bayes Classifiers

- Given an input vector  $\mathbf{x}$ , we compute class probabilities using the **Bayes rule**.
  - ◆ Suppose that there are  $k$  classes  $c_1, c_2, \dots, c_k$ .
  - ◆ Find the maximum posterior  $p(C = c_i | \mathbf{x})$ .

$$p(c_i | \mathbf{x}) = \frac{p(\mathbf{x} | c_i)p(c_i)}{p(\mathbf{x})} \propto p(\mathbf{x} | c_i)p(c_i)$$

- Because  $p(\mathbf{x})$  is constant for all classes, only the **nominator needs to be maximized**.

# Classification: Good vs. Bad



➤ How to building a Bayes classifier?



Training data

	<i>Gender</i>	<i>Mask</i>	<i>Cape</i>	<i>Tie</i>	<i>Ears</i>	<i>Smokes</i>	<i>Label</i>
<b>Batman</b>	Male	Yes	Yes	No	Yes	No	<b>Good</b>
<b>Robin</b>	Male	Yes	Yes	No	No	No	<b>Good</b>
<b>Alfred</b>	Male	No	No	Yes	No	No	<b>Good</b>
<b>Penguin</b>	Male	No	No	Yes	No	Yes	<b>Bad</b>
<b>Catwoman</b>	Female	Yes	No	No	Yes	No	<b>Bad</b>
<b>Joker</b>	Male	No	No	No	No	No	<b>Bad</b>

# Classification: Good vs. Bad

## ➤ How to compute the class probability?

◆ Let  $\mathbf{x} = [male, yes, yes, no, yes, no]$ .

$$p(\text{Good} \mid male, yes, yes, no, yes, no) = \frac{p(male, yes, yes, no, yes, no \mid \text{Good})p(\text{Good})}{p(male, yes, yes, no, yes, no)}$$

$$p(\text{Bad} \mid male, yes, yes, no, yes, no) = \frac{p(male, yes, yes, no, yes, no \mid \text{Bad})p(\text{Bad})}{p(male, yes, yes, no, yes, no)}$$

Training data

	<i>Gender</i>	<i>Mask</i>	<i>Cape</i>	<i>Tie</i>	<i>Ears</i>	<i>Smokes</i>	<i>Label</i>
Batman	Male	Yes	Yes	No	Yes	No	Good
Robin	Male	Yes	Yes	No	No	No	Good
Alfred	Male	No	No	Yes	No	No	Good
Penguin	Male	No	No	Yes	No	Yes	Bad
Catwoman	Female	Yes	No	No	Yes	No	Bad
Joker	Male	No	No	No	No	No	Bad

# Challenge of Bayes Classifiers

- Assume that we have two classes.
- We have  $d$  binary features  $\mathbf{x} = [x_1, \dots, x_d]$ .
- If we define a joint distribution  $p(c, x_1, \dots, x_d)$ , it should give useful information to determine  $p(c)$  and  $p(\mathbf{x} | c)$ .
- Problem: a joint distribution over  $d + 1$  binary variables requires  $2^{d+1}$  entries.
  - ◆ It is computationally prohibitive.
  - ◆ It also would require a huge amount of data to fit.



# Conditional Independence

- Assume that each feature  $x_i$  is **conditionally independent** given the class  $c$ .
  - ◆  $x_i$  and  $x_j$  are **independent under the conditional distribution  $p(\mathbf{x} | c)$** .

$$p(x_1, \dots, x_d | c) = \prod_{i=1}^d p(x_i | c)$$



- **Compact representation of the joint distribution**

- ◆ Prior probability of class:  $p(c = 1) = \theta_c$
- ◆ Conditional probability of features given a class:  $p(x_j = 1 | c) = \theta_{jc}$
- ◆ It only requires  **$2d + 1$**  parameters total.

- **Discussion: is it really true for our data?**



# Naïve Bayes Classifiers (NBC)

➤ What is the most probable value  $p(c | \mathbf{x})$ ?

◆ We have  $d$  binary features  $\mathbf{x} = [x_1, \dots, x_d]$ .

➤ Let's estimate as follows.

$$f(\mathbf{x}) = \operatorname{argmax}_{c \in \mathcal{C}} p(c | x_1, x_2, \dots, x_d)$$

➤ It is equivalent to

$$f(\mathbf{x}) = \operatorname{argmax}_{c \in \mathcal{C}} \frac{p(x_1, x_2, \dots, x_d | c)p(c)}{p(x_1, x_2, \dots, x_d)} = \operatorname{argmax}_{c \in \mathcal{C}} p(x_1, x_2, \dots, x_d | c)p(c)$$



# Training Naïve Bayes Classifiers



- Learn the parameters efficiently because the log-likelihood decomposes into **independent terms for each feature**.

$$\begin{aligned}\log p(\mathcal{D}) &= \sum_{i=1}^n \log p(c^{(i)}, \mathbf{x}^{(i)}) \\ &= \sum_{i=1}^n \log p(\mathbf{x}^{(i)} | c^{(i)}) p(c^{(i)}) = \sum_{i=1}^n \log \prod_{j=1}^d p(x_j^{(i)} | c^{(i)}) p(c^{(i)}) \\ &= \sum_{i=1}^n \left[ \sum_{j=1}^d \log p(x_j^{(i)} | c^{(i)}) + \log p(c^{(i)}) \right] \\ &= \sum_{j=1}^d \sum_{i=1}^n \log p(x_j^{(i)} | c^{(i)}) + \sum_{i=1}^n \log p(c^{(i)})\end{aligned}$$

Bernoulli log-likelihood for feature  $x_j$

Bernoulli log-likelihood of labels

# Training Naïve Bayes Classifiers

➤ We want to maximize  $\sum_{i=1}^n \log p(x_j^{(i)} | c^{(i)})$ .

- ◆ This is similar to a coin-tossing example.
- ◆ Let  $\theta_{11} = p(x_j = 1 | c^{(i)} = 1)$  and  $\theta_{01} = 1 - \theta_{11}$

➤ Calculate the log-likelihood estimation.

$$\begin{aligned} \sum_{i=1}^n \log p(x_j^{(i)} | c^{(i)}) &= \sum_{i=1}^n c^{(i)} x_j^{(i)} \log \theta_{11} + \sum_{i=1}^n c^{(i)} (1 - x_j^{(i)}) \log(1 - \theta_{11}) \\ &\quad + \sum_{i=1}^n (1 - c^{(i)}) x_j^{(i)} \log \theta_{10} + \sum_{i=1}^n (1 - c^{(i)}) (1 - x_j^{(i)}) \log(1 - \theta_{10}) \end{aligned}$$

# Training Naïve Bayes Classifiers

➤ Use the maximum likelihood estimation.

- ◆ Let  $n_{ab}$  is the counts for  $x_j = a$  and  $c = b$ .

$$\theta_{11} = \frac{n_{11}}{n_{11} + n_{01}}$$

$$\theta_{10} = \frac{n_{10}}{n_{10} + n_{00}}$$

- Given a specific class, we count the number of samples for each value in  $x_j$ .
- Similarly, we can also calculate  $\sum_{i=1}^n \log p(c^{(i)})$ .

# Inference of Naïve Bayes Classifiers



## ➤ Apply the Bayes rule.

- ◆ We can ignore computing the denominator to determine the most probable class  $c_i$ .

$$p(c_i | \mathbf{x}) = \frac{p(\mathbf{x} | c_i) p(c_i)}{p(\mathbf{x})} \propto p(\mathbf{x} | c_i) p(c_i)$$

## ➤ Assume the conditional independence assumption.

$$p(c_i | \mathbf{x}) \propto \prod_{i=1}^d p(x_i | c_i) p(c_i)$$

Two terms are computed by MLE.

# Example: Good or Bad

- Given sample  $x = (male, yes, yes, no, no, no)$ , predict one of the two classes **Good** and **Bad**.

Training data

	<i>Gender</i>	<i>Mask</i>	<i>Cape</i>	<i>Tie</i>	<i>Ears</i>	<i>Smokes</i>	<i>Label</i>
Batman	Male	Yes	Yes	No	Yes	No	Good
Robin	Male	Yes	Yes	No	No	No	Good
Alfred	Male	No	No	Yes	No	No	Good
Penguin	Male	No	No	Yes	No	Yes	Bad
Catwoman	Female	Yes	No	No	Yes	No	Bad
Joker	Male	No	No	No	No	No	Bad

Test data

Superman	Male	Yes	Yes	No	No	No
----------	------	-----	-----	----	----	----



# Example: Good or Bad



	<i>Gender</i>	<i>Mask</i>	<i>Cape</i>	<i>Tie</i>	<i>Ears</i>	<i>Smokes</i>	<i>Label</i>
<b>Batman</b>	Male	Yes	Yes	No	Yes	No	<b>Good</b>
<b>Robin</b>	Male	Yes	Yes	No	No	No	<b>Good</b>
<b>Alfred</b>	Male	No	No	Yes	No	No	<b>Good</b>

➤ For each attribute, the conditional probability is

- ♦  $p(G = \text{Male} \mid \text{Good}) = 1.0$ ,  $p(M = \text{Yes} \mid \text{Good}) = 0.67$ ,
- ♦  $p(C = \text{Yes} \mid \text{Good}) = 0.67$ ,  $p(T = \text{No} \mid \text{Good}) = 0.67$ ,
- ♦  $p(E = \text{No} \mid \text{Good}) = 0.67$ ,  $p(S = \text{No} \mid \text{Good}) = 0.67$

# Example: Good or Bad

	<i>Gender</i>	<i>Mask</i>	<i>Cape</i>	<i>Tie</i>	<i>Ears</i>	<i>Smokes</i>	<i>Label</i>
Penguin	Male	No	No	Yes	No	Yes	Bad
Catwoman	Female	Yes	No	No	Yes	No	Bad
Joker	Male	No	No	No	No	No	Bad

## ➤ For each attribute, the conditional probability is

- ◆  $p(G = \text{Male} \mid \text{Bad}) = 0.67$ ,  $p(M = \text{Yes} \mid \text{Bad}) = 0.33$
- ◆  $p(C = \text{Yes} \mid \text{Bad}) = 0.00$ ,  $p(T = \text{No} \mid \text{Bad}) = 0.67$
- ◆  $p(E = \text{No} \mid \text{Bad}) = 0.67$ ,  $p(S = \text{No} \mid \text{Bad}) = 0.67$

## ➤ Prior probability

- ◆  $P(\text{Good}) = 0.5$ ,  $P(\text{Bad}) = 0.5$

# Example: Good or Bad



	<i>Gender</i>	<i>Mask</i>	<i>Cape</i>	<i>Tie</i>	<i>Ears</i>	<i>Smokes</i>
Superman	Male	Yes	Yes	No	No	No

➤ For **Good** class,

$$\blacklozenge 1.0 \times 0.67 \times 0.67 \times 0.67 \times 0.67 \times 0.67 \times 0.5$$

➤ For **Bad** class,

$$\blacklozenge 0.67 \times 0.33 \times 0.00 \times 0.67 \times 0.67 \times 0.67 \times 0.5$$

➤ Therefore, we predict **Good** for x.



# Example: Buying Laptops

➤  $\mathbf{x} = (A \leq 30, I = \text{Medium}, S = \text{Yes}, C = \text{Fair})$

Training data

<i>Age</i>	<i>Income</i>	<i>Student</i>	<i>Credit</i>	<i>Buy</i>
$\leq 30$	High	N	Fair	No
$\leq 30$	High	N	Excellent	No
31 ... 40	High	N	Fair	Yes
$> 40$	Medium	N	Fair	Yes
$> 40$	Low	Y	Fair	Yes
$> 40$	Low	Y	Excellent	No
31 ... 40	Low	Y	Excellent	Yes
$\leq 30$	Medium	N	Fair	No
$\leq 30$	Low	Y	Fair	Yes
$> 40$	Medium	Y	Fair	Yes
$\leq 30$	Medium	Y	Excellent	Yes
31 ... 40	Medium	N	Excellent	Yes
31 ... 40	High	Y	Fair	Yes
$> 40$	Medium	N	Excellent	No

# Example: Buying Laptops

➤  $\mathbf{x} = (A \leq 30, I = \text{Medium}, S = \text{Yes}, C = \text{Fair})$

➤ **Prior probability**  $p(c_i)$

- ◆  $p(\text{Buy} = \text{Yes}) = 9/14 = 0.643,$

- ◆  $p(\text{Buy} = \text{No}) = 5/14 = 0.357$

➤ **Conditional probability**  $p(x_j | c_i)$

- ◆  $p(\text{Age} = \leq 30 | \text{Buy} = \text{Yes}) = 2/9 = 0.222$

- ◆  $p(\text{Age} = \leq 30 | \text{Buy} = \text{No}) = 3/5 = 0.6$

- ◆  $p(\text{Income} = \text{Medium} | \text{Buy} = \text{Yes}) = 4/9 = 0.444$

- ◆  $p(\text{Income} = \text{Medium} | \text{Buy} = \text{No}) = 2/5 = 0.4$

- ◆  $p(\text{Student} = \text{Yes} | \text{Buy} = \text{Yes}) = 6/9 = 0.667$

- ◆  $p(\text{Student} = \text{Yes} | \text{Buy} = \text{No}) = 1/5 = 0.2$

- ◆  $p(\text{Credit} = \text{Fair} | \text{Buy} = \text{Yes}) = 6/9 = 0.667$

- ◆  $p(\text{Credit} = \text{Fair} | \text{Buy} = \text{No}) = 2/5 = 0.4$

# Example: Buying Laptops

➤  $\mathbf{x} = (A \leq 30, I = \text{Medium}, S = \text{Yes}, C = \text{Fair})$

➤ **Conditional probability**

◆  $p(\mathbf{x} \mid \text{Buy} = \text{Yes}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$

◆  $p(\mathbf{x} \mid \text{Buy} = \text{No}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$

➤ **Which class maximizes  $p(x \mid c_i) \times p(c_i)$ ?**

◆  $p(\mathbf{x} \mid \text{Buy} = \text{Yes}) \times p(\text{Buy} = \text{Yes}) = 0.028$

◆  $p(\mathbf{x} \mid \text{Buy} = \text{No}) \times p(\text{Buy} = \text{No}) = 0.007$

➤ **We predict the class **Yes** for  $\mathbf{x}$ .**

# Zero Probability Problem

➤ Given a sample  $\mathbf{x} = (Male, Yes, Yes)$

$$p(\text{Good} | Male, Yes, Yes) = p(G = Male | \text{Good})p(M = Yes | \text{Good})p(C = Yes | \text{Good})p(\text{Good})$$

$$= 1.0 \times 0.0 \times 0.33 \times 0.5 = 0.0$$

$$p(\text{Bad} | Male, Yes, Yes) = p(G = Male | \text{Bad})p(M = Yes | \text{Bad})p(C = Yes | \text{Bad})P(\text{Bad})$$

$$= 0.67 \times 1.0 \times 0.0 \times 0.5 = 0.0$$

	<i>Gender</i>	<i>Mask</i>	<i>Cape</i>	<i>Label</i>
<b>Batman</b>	Male	No	Yes	<b>Good</b>
<b>Robin</b>	Male	No	No	<b>Good</b>
<b>Alfred</b>	Male	No	No	<b>Good</b>
<b>Penguin</b>	Male	Yes	No	<b>Bad</b>
<b>Catwoman</b>	Female	Yes	No	<b>Bad</b>
<b>Joker</b>	Male	Yes	No	<b>Bad</b>



# Laplace Smoothing

- Given 1000 samples, they have Income = low (0), Income = medium (990), and Income = high (10).
- Use **Laplacian correction** (or Laplacian estimator).
  - ◆ **Adding 1 to each case**
  - ◆ The corrected probability estimates are close to their uncorrected counterparts.
- **Update the probability for each value.**
  - ◆  $p(\text{Income} = \text{low}) = 1/1003$
  - ◆  $p(\text{Income} = \text{medium}) = 991/1003$
  - ◆  $p(\text{Income} = \text{high}) = 11/1003$

# Laplace Smoothing

## ➤ Adding $k$ to every outcome

$$p_{LAP,k}(X) = \frac{c(x_i) + k}{n + k|X|}$$

- ◆  $c(x_i)$  is the number of samples for  $X = x_i$
- ◆  $n$  is the number of entire samples.
- ◆  $k$  is the **strength** of prior.
- ◆  $|X|$  is the number of values in  $X$ .



$$p_{LAP,0}(X) = \left\langle \frac{4}{4}, \frac{0}{4} \right\rangle$$

$$p_{LAP,1}(X) = \left\langle \frac{5}{6}, \frac{1}{6} \right\rangle$$

$$p_{LAP,10}(X) = \left\langle \frac{14}{24}, \frac{10}{24} \right\rangle$$

$$p_{LAP,100}(X) = \left\langle \frac{104}{204}, \frac{100}{204} \right\rangle$$

# Application: Spam Detection



➤ Classify text into **spam/non-spam**

➤ Example: “You are the winner for the free \$1000 gift card.”

➤ Use **bag-of-words** features.

◆ Each word is conditionally independent of each class.

a	...	car	card	...	win	winner	...	you	...
0	...	0	1	...	0	1	...	1	...

# Summary of Naïve Bayes Classifiers



- It is an amazingly **cheap** model!
  
- **Training time: estimate parameters using maximum likelihood.**
  - ◆ Compute co-occurrence counts of each feature with the labels.
  - ◆ Requires only one pass through the data!
  
- **Test time: apply Bayes' Rule.**
  
- **It is easily extended to other probability distributions.**
  - ◆ Unfortunately, it is usually **less accurate** in practice compared to discriminative models due to the **naïve independence assumption**.



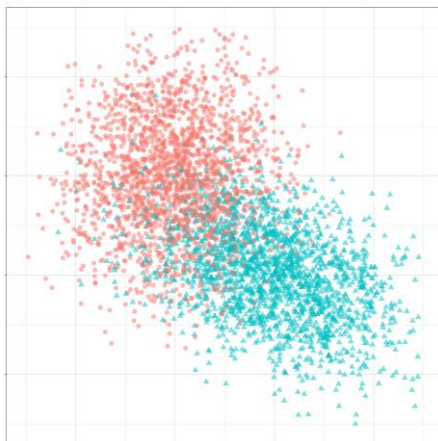


# **Gaussian Discriminant Analysis (GDA)**

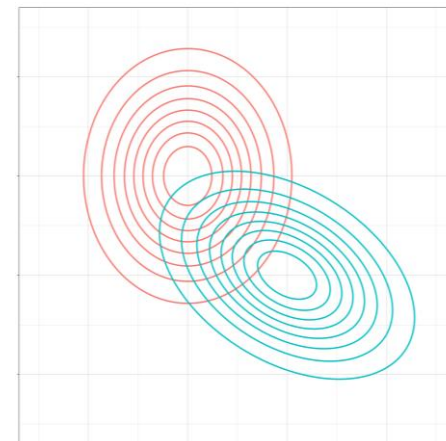
# Motivation

- We want to model what each class like look.
  - ◆ However,  $p(\mathbf{x} \mid C = k)$  may be very complex.
  - ◆ Naïve Bayes used the conditional independence assumption.
- Q: What else can we do? **choose a simple distribution.**
- A: Fit the **Gaussian distribution** to our data.

$$p(\mathbf{x} \mid C = k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



Gaussian Discriminant  
Analysis



# Why the Gaussian Distribution?

- It commonly appears by nature.
  - ◆ Almost all variables are distributed approximately normally.



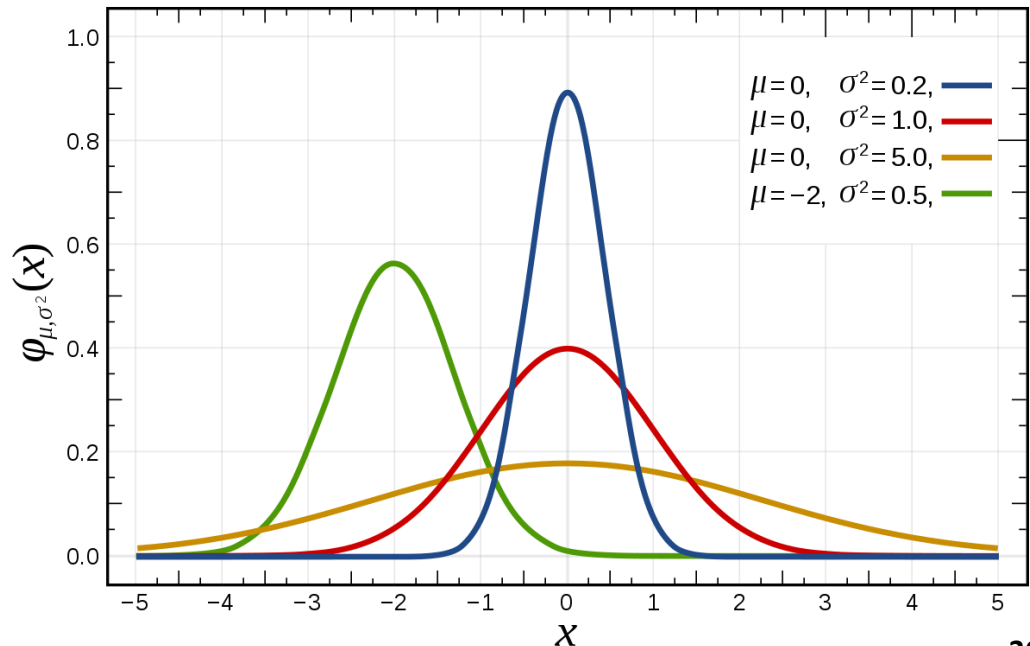
- The idea behind it is the **central limit theorem**.
  - ◆ The sampling distribution is approximately **normally distributed** if the sample size is large enough.

# Univariate Normal Distribution

- Given a mean and a variance, we can calculate the probability distribution function of a normal distribution.

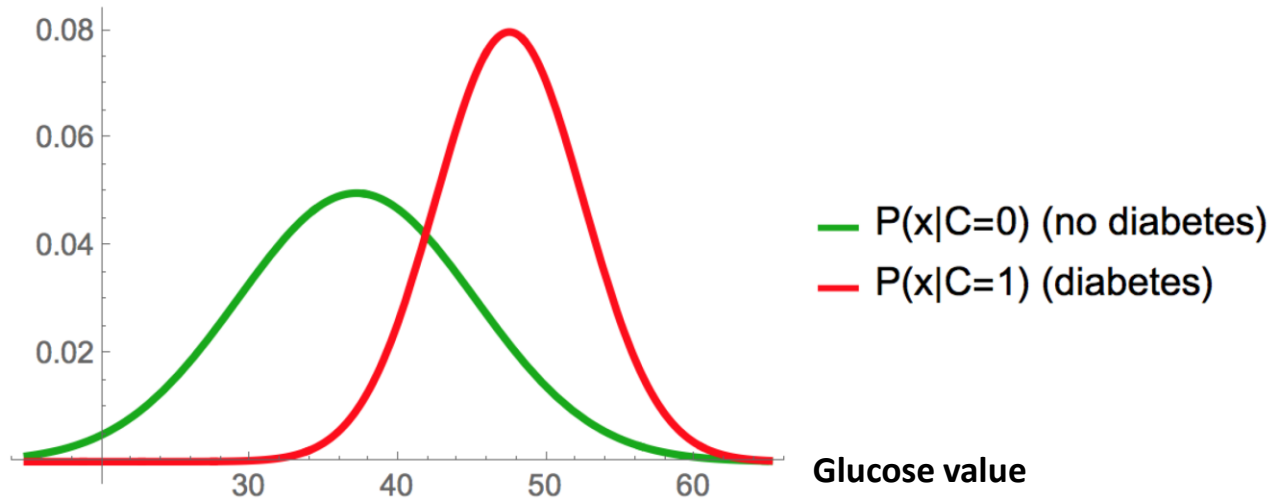
$$p(x | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

- It has two parameters.
  - ◆ Mean  $\mu$  and variance  $\sigma^2$



# Example: Diabetes Classification

- Binary classification: People with/without diabetes



- Q: What if we have two or higher dimensional data?
- A: Use the **multivariate Gaussian distribution**.

# Multivariate Gaussian Distribution

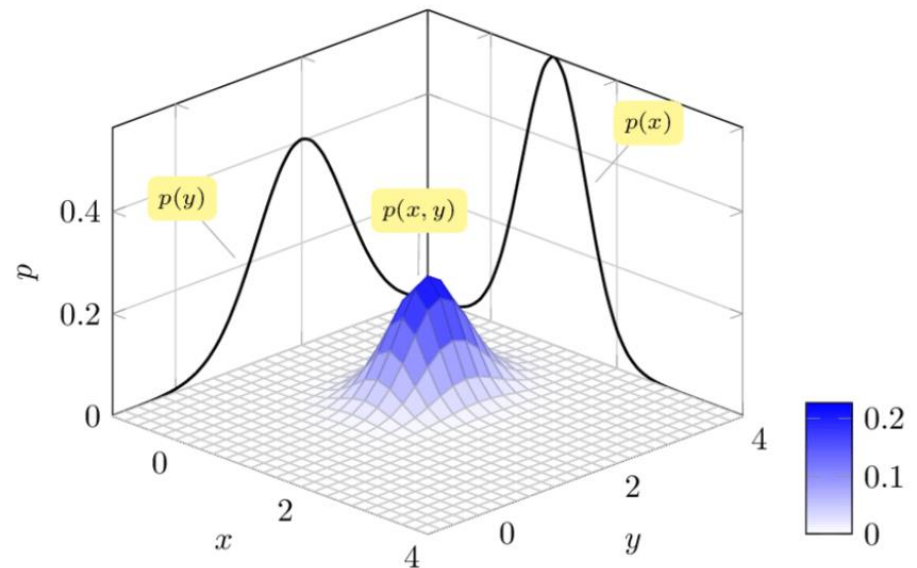
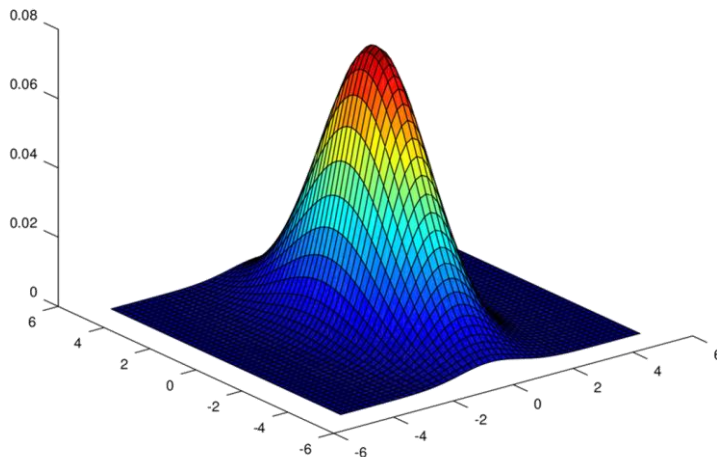


➤ Assume that  $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , where  $\boldsymbol{\mu} \in \mathbb{R}^d$  and  $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ .

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

Let  $|\boldsymbol{\Sigma}|$  be the determinant of the matrix  $\boldsymbol{\Sigma}$ .

Let  $\boldsymbol{\Sigma}^{-1}$  be inverse matrix of  $\boldsymbol{\Sigma}$ .



# Multivariate Data

- Multiple measurements (sensors)
- $d$  features/attributes
- $n$  samples/instances/examples

$$\mathbf{X} = \begin{bmatrix} [\mathbf{x}^{(1)}]^T \\ [\mathbf{x}^{(2)}]^T \\ \vdots \\ [\mathbf{x}^{(n)}]^T \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(n)} & x_2^{(n)} & \dots & x_d^{(n)} \end{bmatrix}$$

# of samples

# of features

# Multivariate Parameters

## ➤ Mean

$$\mathbb{E}[\mathbf{x}] = [\mu_1, \dots, \mu_d]^T \in \mathbb{R}^d$$

## ➤ Covariance

$$\Sigma = \text{Cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})^T(\mathbf{x} - \boldsymbol{\mu})] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

# of features

# of features



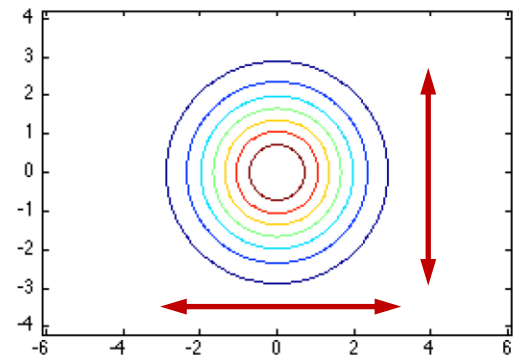
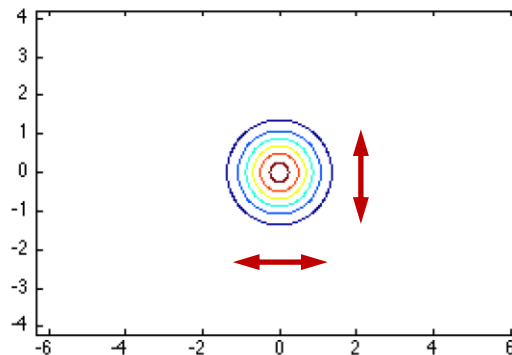
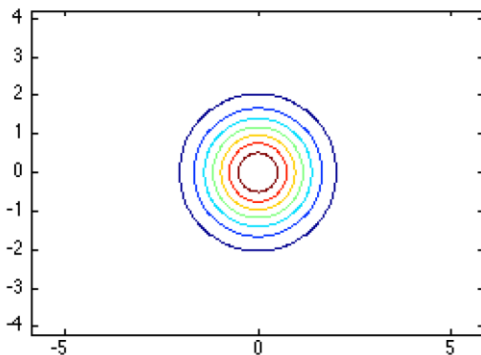
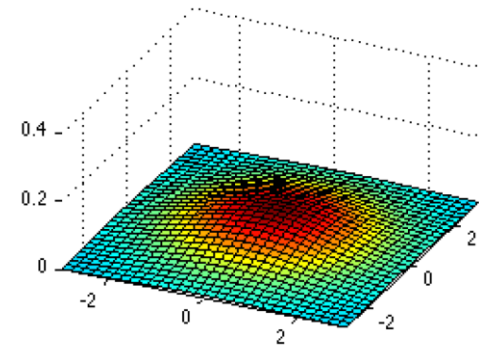
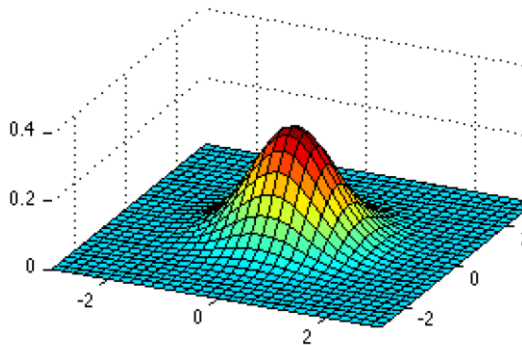
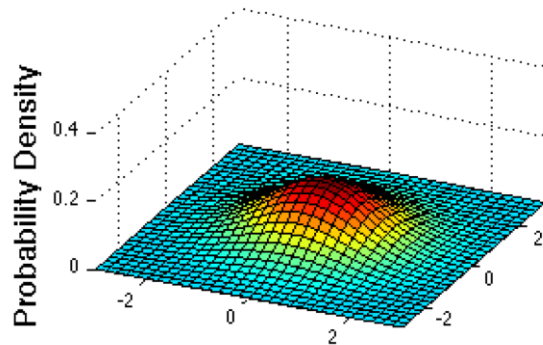
# Bivariate Gaussian Distribution



$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = 0.5 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = 2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

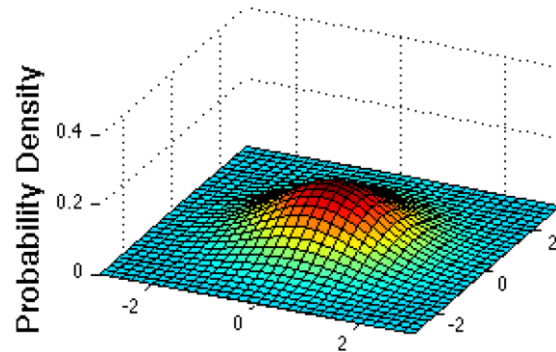


Contour plot of the pdf

# Bivariate Gaussian Distribution

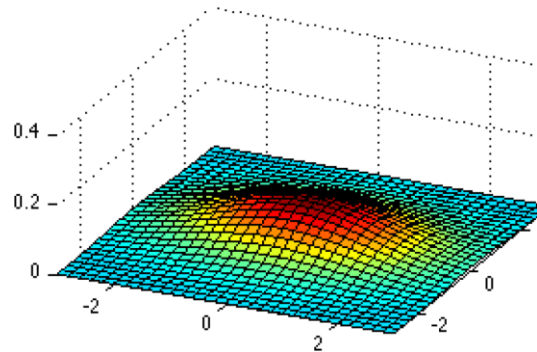
$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\text{Var}(x_1) = \text{Var}(x_2)$$



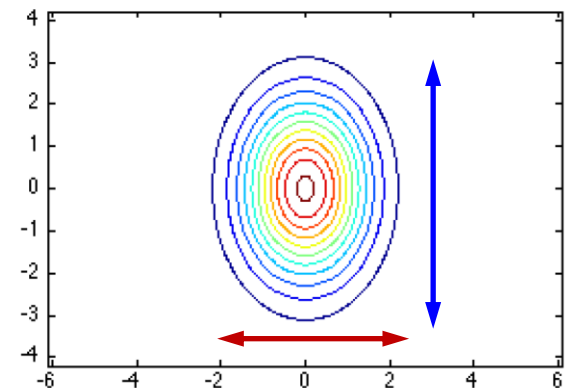
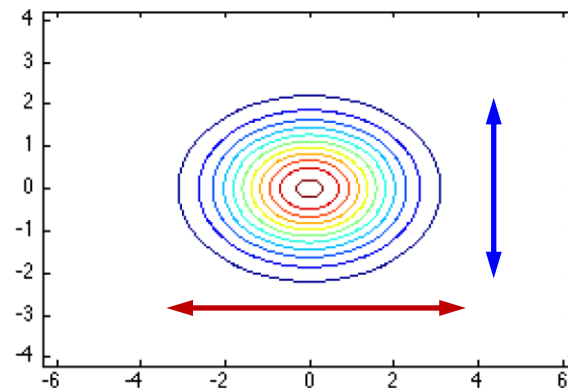
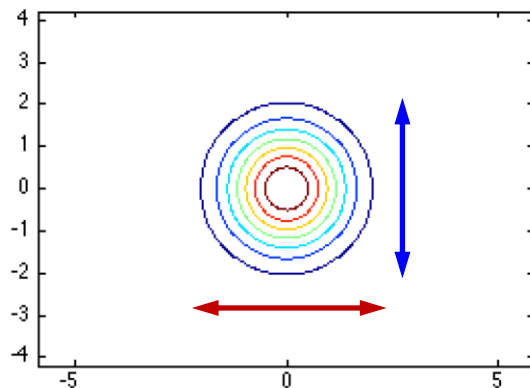
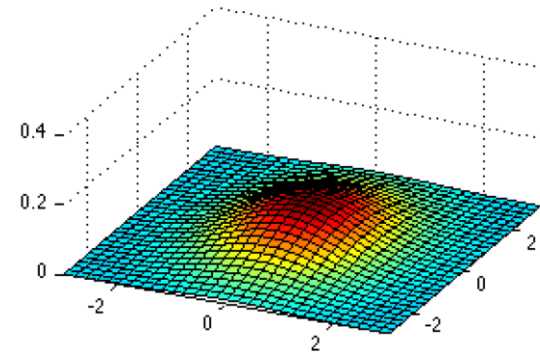
$$\Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\text{Var}(x_1) > \text{Var}(x_2)$$



$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\text{Var}(x_1) < \text{Var}(x_2)$$



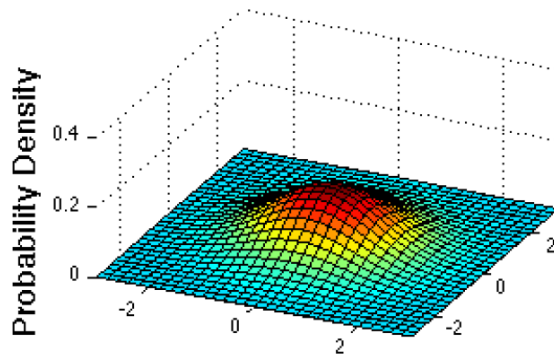
Contour plot of the pdf

# Bivariate Gaussian Distribution

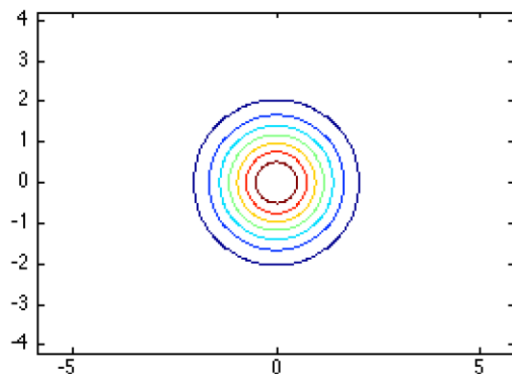


$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\text{Cov}(x_1, x_2) = 0$$

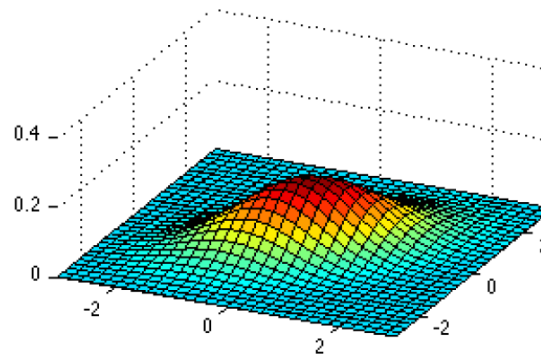


Independent

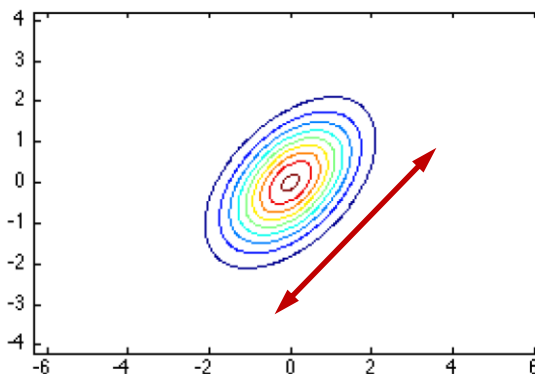


$$\Sigma = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\text{Cov}(x_1, x_2) > 0$$

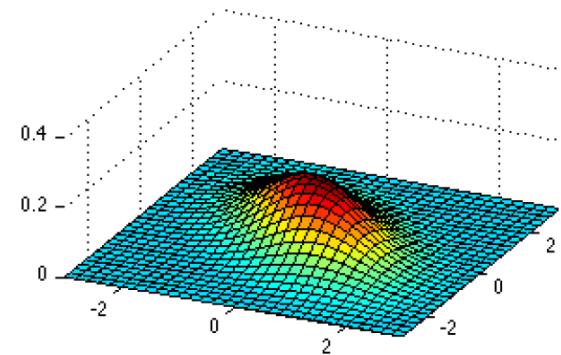


Positively correlated

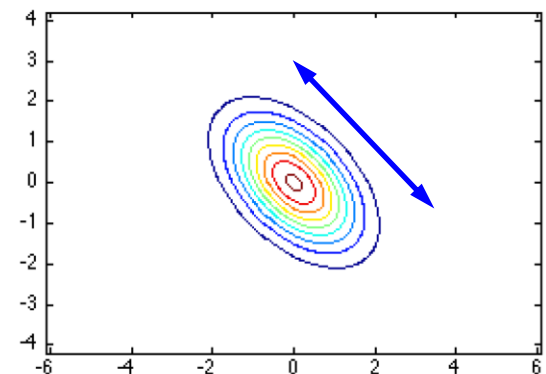


$$\Sigma = 2 \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$\text{Cov}(x_1, x_2) < 0$$



Negatively correlated



Contour plot of the pdf

# Gaussian Discriminant Analysis (GDA)



- Assume that  $p(\mathbf{x} \mid C = k)$  is distributed according to a multivariate normal (Gaussian) distribution.

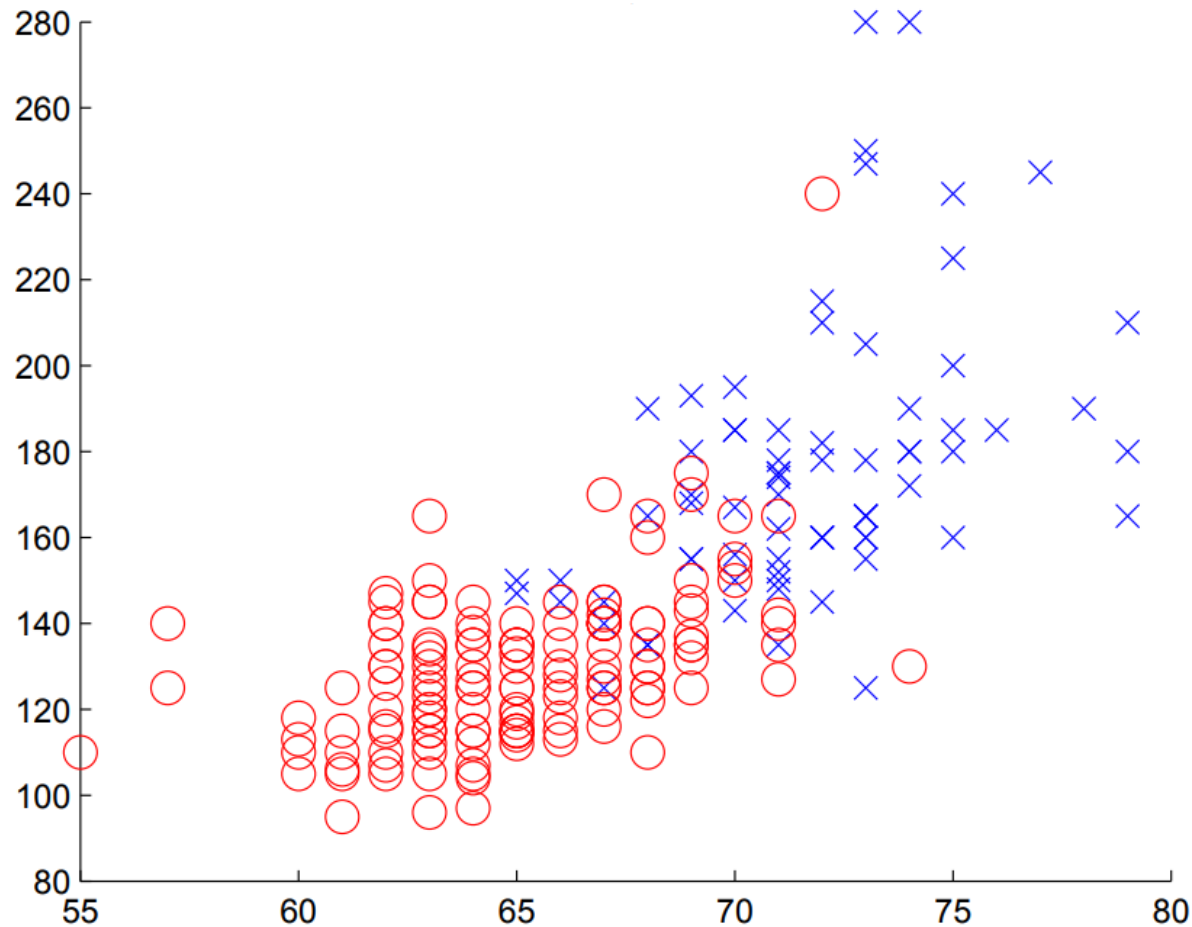
- **Multivariate Gaussian distribution**

$$p(\mathbf{x} \mid C = k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right)$$

- ◆ Let  $|\Sigma_k|$  denote the determinant of the matrix.
  - ◆ Let  $d$  denote the dimensionality of  $\mathbf{x}$ .
- Each class has associated with the **mean vector  $\boldsymbol{\mu}_k$**  and the **covariance matrix  $\Sigma_k$** .
    - ◆ Note:  $\Sigma_k$  has  $O(d^2)$  parameters, which is difficult to estimate.

# Example: Determining Genders

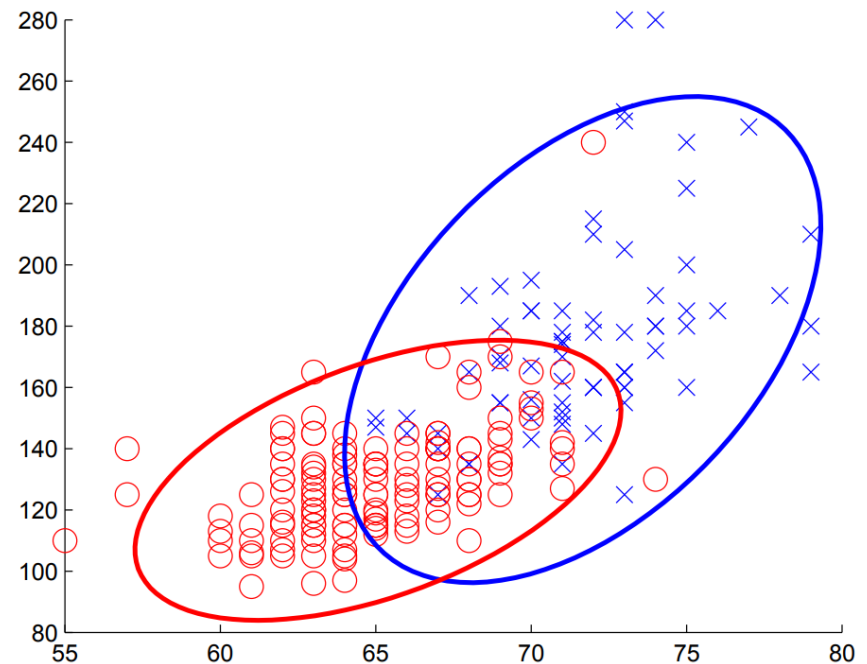
➤ **Red = Female**, **Blue = Male**



# Example: Determining Genders

➤ Modeling the joint distribution of  $p(\mathbf{x}, y)$

$x_1$	$x_2$	Label
65	185	1
52	125	0
56	140	1
62	240	0
57	130	1
...	...	...



➤ For each class, we assume multivariate Gaussian distributions.

# Modeling the Joint Distribution (1D)

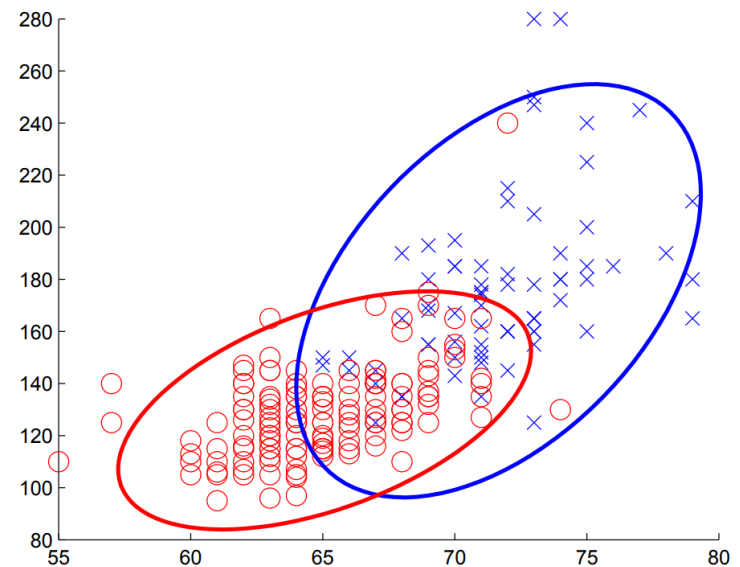


➤ What are the parameters to learn?

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

$$= \begin{cases} p_0 \frac{1}{\sqrt{2\pi}|\boldsymbol{\Sigma}_0|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1}(\mathbf{x} - \boldsymbol{\mu}_0)\right) & \text{if } y = 0 \\ p_1 \frac{1}{\sqrt{2\pi}|\boldsymbol{\Sigma}_1|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x} - \boldsymbol{\mu}_1)\right) & \text{if } y = 1 \end{cases}$$

$p_0 + p_1 = 1$  are **prior probabilities**, and  $p(\mathbf{x} | y)$  is a **conditional distribution**.



# Parameter Estimation ( $d = 1$ )

➤ **Log likelihood of training data**  $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$ .

- ◆ Assume that we have binary classes  $y^{(i)} \in \{0, 1\}$

$$\begin{aligned}\ln p(\mathcal{D}) &= \sum_i \ln p(x^{(i)}, y^{(i)}) \\ &= \sum_{i: y^{(i)}=0} \ln \left( p_0 \frac{1}{\sqrt{2\pi}\sigma_0} \exp \left( -\frac{(x - \mu_0)^2}{2\sigma_0^2} \right) \right) + \\ &\quad \sum_{i: y^{(i)}=1} \ln \left( p_1 \frac{1}{\sqrt{2\pi}\sigma_1} \exp \left( -\frac{(x - \mu_1)^2}{2\sigma_1^2} \right) \right)\end{aligned}$$

➤ **Max log likelihood**  $(p_0^*, p_1^*, \mu_0^*, \mu_1^*, \sigma_0^*, \sigma_1^*) = \operatorname{argmax} \log p(\mathcal{D})$

- ◆ For  $d = 2$ , we compute  $\Sigma_0^*$  and  $\Sigma_1^*$  instead of  $\sigma_0^*$  and  $\sigma_1^*$ .



# MLE for Gaussian Distributions ( $d = 1$ )



- Assume that each sample is **independent and identically distributed**.

$$p(x^{(1)}, \dots, x^{(n)} | y) = \prod_{i=1}^n p(x^{(i)} | y) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x^{(i)} - \mu)^2}{2\sigma^2}\right)$$

- We minimize negative log-likelihood.

$$\begin{aligned} -\ln p(x^{(1)}, \dots, x^{(n)} | y) &= -\ln \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x^{(i)} - \mu)^2}{2\sigma^2}\right) \\ &= \sum_{i=1}^n \ln \sqrt{2\pi}\sigma + \sum_{i=1}^n \frac{(x^{(i)} - \mu)^2}{2\sigma^2} = \frac{n}{2} \ln 2\pi\sigma^2 + \sum_{i=1}^n \frac{(x^{(i)} - \mu)^2}{2\sigma^2} \end{aligned}$$

# MLE for Gaussian Distributions ( $d = 1$ )



➤ Compute the partial derivative and set it to zero.

$$-\ln p(x^{(1)}, \dots, x^{(n)} | y) = \frac{n}{2} \ln 2\pi\sigma^2 + \sum_{i=1}^n \frac{(x^{(i)} - \mu)^2}{2\sigma^2}$$

$$\frac{\partial \mathcal{L}}{\partial \mu} = \frac{1}{2\sigma^2} \sum_{i=1}^n -2(x^{(i)} - \mu) = - \sum_{i=1}^n \frac{(x^{(i)} - \mu)}{\sigma^2} = \frac{n\mu - \sum_{i=1}^n x^{(i)}}{\sigma^2}$$

$$\frac{\partial \mathcal{L}}{\partial \mu} = 0 = \frac{n\mu - \sum_{i=1}^n x^{(i)}}{\sigma^2} \Rightarrow$$

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x^{(i)}$$

# MLE for Gaussian Distributions ( $d = 1$ )



➤ Compute the partial derivative and set it to zero.

$$-\ln p(x^{(1)}, \dots, x^{(n)} | y) = \frac{n}{2} \ln 2\pi\sigma^2 + \sum_{i=1}^n \frac{(x^{(i)} - \mu)^2}{2\sigma^2}$$

$$\frac{\partial \mathcal{L}}{\partial \sigma^2} = \frac{n}{2} \frac{1}{2\pi\sigma^2} 2\pi + \sum_{i=1}^n \frac{(x^{(i)} - \mu)^2}{2} \left( -\frac{1}{\sigma^4} \right) = \frac{n}{2\sigma^2} - \frac{\sum_{i=1}^n (x^{(i)} - \mu)^2}{2\sigma^4}$$

$$\frac{\partial \mathcal{L}}{\partial \sigma^2} = 0 = \frac{n}{2\sigma^2} - \frac{\sum_{i=1}^n (x^{(i)} - \mu)^2}{2\sigma^4} = \frac{n\sigma^2 - \sum_{i=1}^n (x^{(i)} - \mu)^2}{2\sigma^4}$$

$$\Rightarrow \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x^{(i)} - \mu)^2$$

# MLE for Gaussian Distributions ( $d = 1$ )



- We can compute the parameters of a Gaussian distribution by taking the training samples for a specific class.

$$\hat{\mu} = \frac{\sum_{i=1}^n \mathbb{I}[y^{(i)} = k] \cdot x^{(i)}}{\sum_{i=1}^n \mathbb{I}[y^{(i)} = k]} = \frac{\sum_{j:y^{(i)}=k} x^{(j)}}{\text{\# of training samples in class } k}$$



$$\sigma^2 = \frac{\sum_{i=1}^n \mathbb{I}[y^{(i)} = k] (x^{(i)} - \mu)^2}{\sum_{i=1}^n \mathbb{I}[y^{(i)} = k]} = \frac{\sum_{j:y^{(i)}=k} (x^{(j)} - \mu)^2}{\text{\# of training samples in class } k}$$

- It can be extended to the multivariate Gaussian distribution.

# Parameter Estimation ( $d > 1$ )

## ➤ Learning the parameter for each class using MLE.

- ◆ Assume that the prior is Bernoulli (we have two classes).

$$p(y^{(i)}) = \phi^{y^{(i)}} (1 - \phi)^{1-y^{(i)}}$$

## ➤ Compute the ML estimate for parameters.

$$\phi = \frac{1}{n} \sum_{i=1}^n \mathbb{I}[y^{(i)} = k]$$

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^n \mathbb{I}[y^{(i)} = k] \cdot \mathbf{x}^{(i)}}{\sum_{i=1}^n \mathbb{I}[y^{(i)} = k]}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_{i=1}^n \mathbb{I}[y^{(i)} = k]} \sum_{i=1}^n \mathbb{I}[y^{(i)} = k] (\mathbf{x}^{(i)} - \boldsymbol{\mu}_{y^{(i)}}) (\mathbf{x}^{(i)} - \boldsymbol{\mu}_{y^{(i)}})^T$$

# Parameter Estimation ( $d > 1$ )

➤ For binary classification, we estimate parameters using MLE.

◆ Similarly, we can compute  $p_0$ ,  $\mu_0$ , and  $\Sigma_0$ .

$$p_1 = \frac{1}{n} \sum_{i=1}^n \mathbb{I}[y^{(i)} = 1] = \frac{\text{\# of training samples in class 1}}{\text{\# of training samples}}$$

$$\mu_1 = \frac{\sum_{j=1}^n \mathbb{I}[y^{(j)} = 1] \cdot \mathbf{x}^{(j)}}{\sum_{i=1}^n \mathbb{I}[y^{(i)} = 1]} = \frac{\sum_{j:y^{(j)}=1} \mathbf{x}^{(j)}}{\text{\# of training samples in class 1}}$$

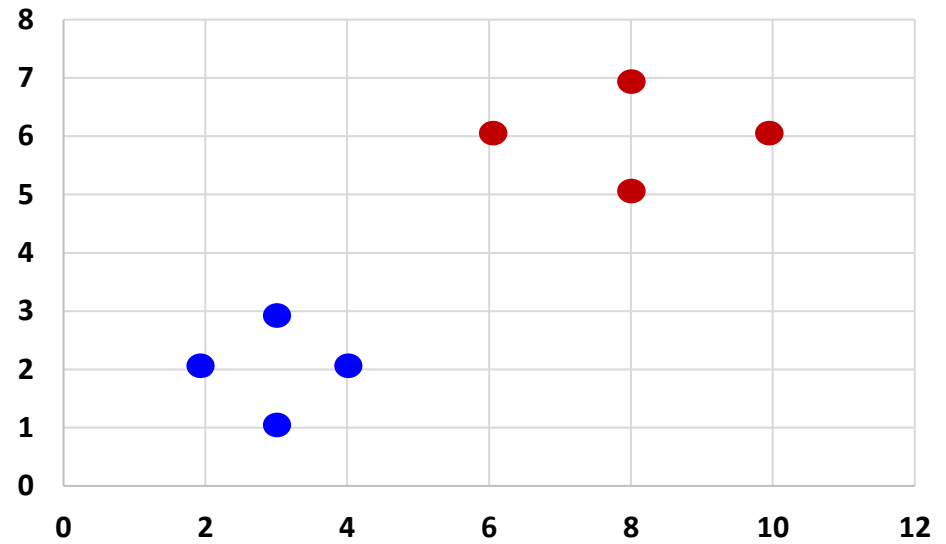
$$\begin{aligned} \Sigma_1 &= \frac{1}{\sum_{i=1}^n \mathbb{I}[y^{(i)} = 1]} \sum_{j=1}^n \mathbb{I}[y^{(j)} = 1] (\mathbf{x}^{(j)} - \mu_{y^{(j)}}) (\mathbf{x}^{(j)} - \mu_{y^{(j)}})^T \\ &= \frac{\sum_{j:y^{(j)}=1} (\mathbf{x}^{(j)} - \mu_1)^2}{\text{\# of training samples in class 1}} \end{aligned}$$

# Example



➤ Each class consists of four samples.

$x_1$	$x_2$	$y$
2	2	0
3	3	0
4	2	0
3	1	0
6	6	1
8	5	1
10	6	1
8	7	1



$$p_0 = 0.5$$

$$p_1 = 0.5$$

$$\mu_0 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$\mu_1 = \begin{pmatrix} 8 \\ 6 \end{pmatrix}$$

$$\Sigma_0 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$

$$\Sigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 0.5 \end{pmatrix}$$

# Decision Boundary

- Under the joint distribution, the prediction depends on

$$p(y = 1 | \mathbf{x}) \geq p(y = 0 | \mathbf{x})$$



$$p(\mathbf{x} | y = 1)p(y = 1) \geq p(\mathbf{x} | y = 0)p(y = 0)$$

- Under the Gaussian distribution,

$$-\frac{(x - \mu_1)^2}{2\sigma_1^2} - \ln \sqrt{2\pi}\sigma_1 + \ln p_1 \geq -\frac{(x - \mu_0)^2}{2\sigma_0^2} - \ln \sqrt{2\pi}\sigma_0 + \ln p_0$$

This inequality represents  $ax^2 + bx + c \geq 0$ .

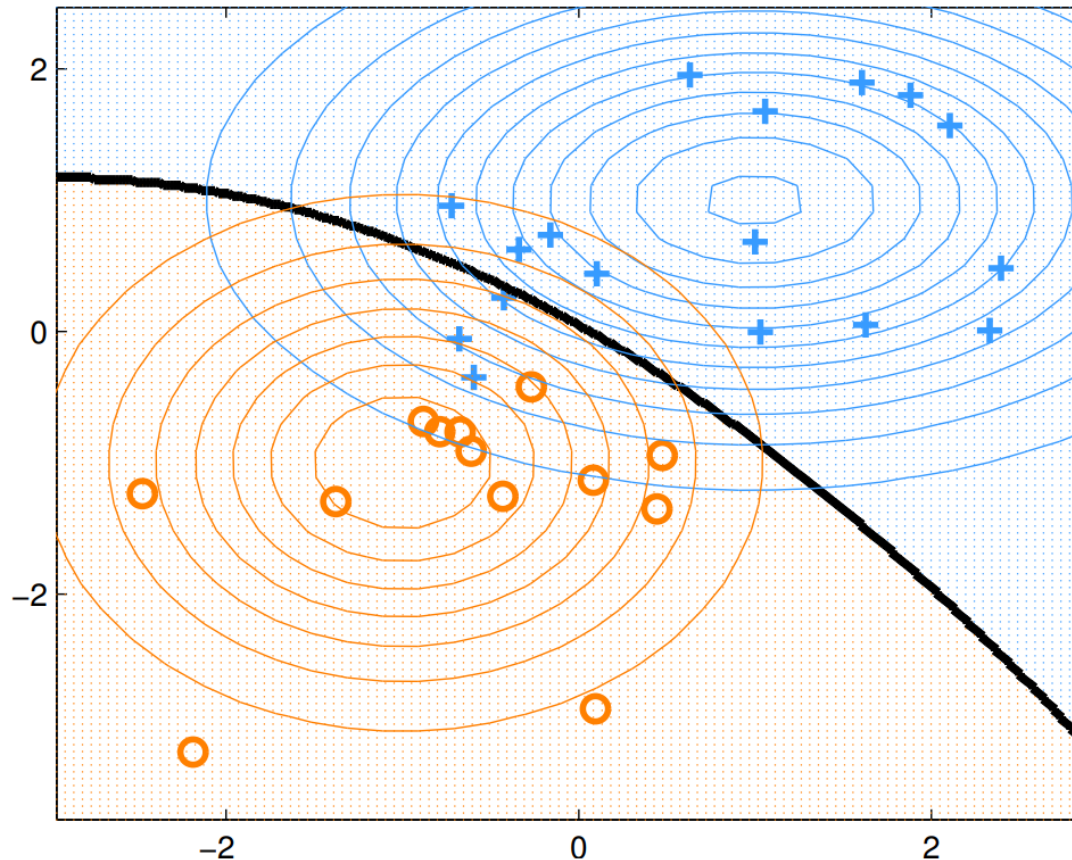
The decision boundary is not **linear**.



# Visualizing the Decision Boundary



- The boundary is characterized by a **quadratic function**.
  - ◆ The shape of the boundary looks like a **parabolic curve**.



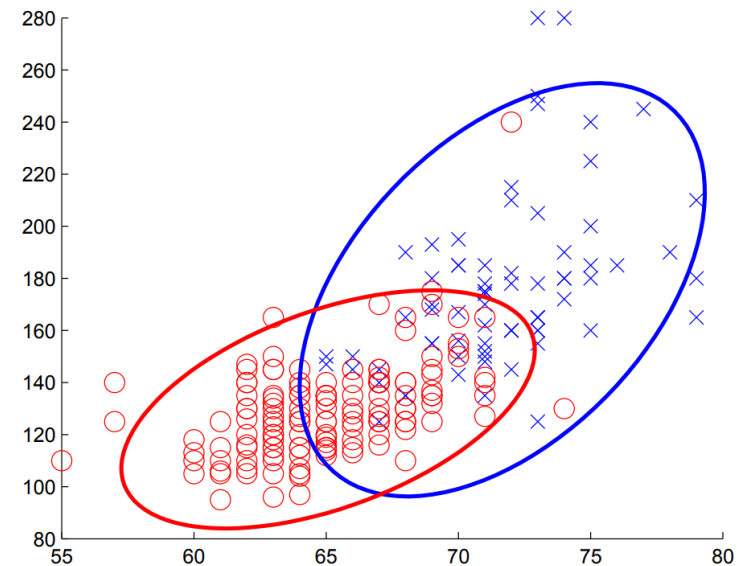
# Special Case: Same Variance



➤ Equal variance for classes can be a **strong assumption**.

➤ **Example:**

- ◆ The male distribution has a **higher variance** than the female distribution.



➤ The assumption might not be applicable.

- ◆ However, **it can significantly reduce the number of parameters**.

# Special Case: Same Variance

- What if assuming two Gaussians have the same variance?

$$-\frac{(x - \mu_1)^2}{2\sigma_1^2} - \ln \sqrt{2\pi}\sigma_1 + \ln p_1 \geq -\frac{(x - \mu_1)^2}{2\sigma_1^2} - \ln \sqrt{2\pi}\sigma_1 + \ln p_0$$



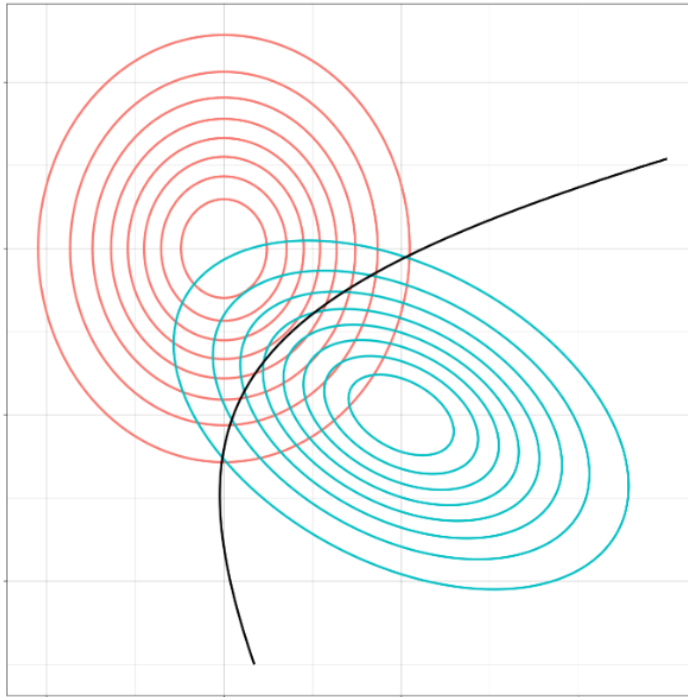
$$\sigma_0 = \sigma_1$$

$$-\frac{(x - \mu_1)^2}{2\sigma^2} - \ln \sqrt{2\pi}\sigma + \ln p_1 \geq -\frac{(x - \mu_0)^2}{2\sigma^2} - \ln \sqrt{2\pi}\sigma + \ln p_0$$

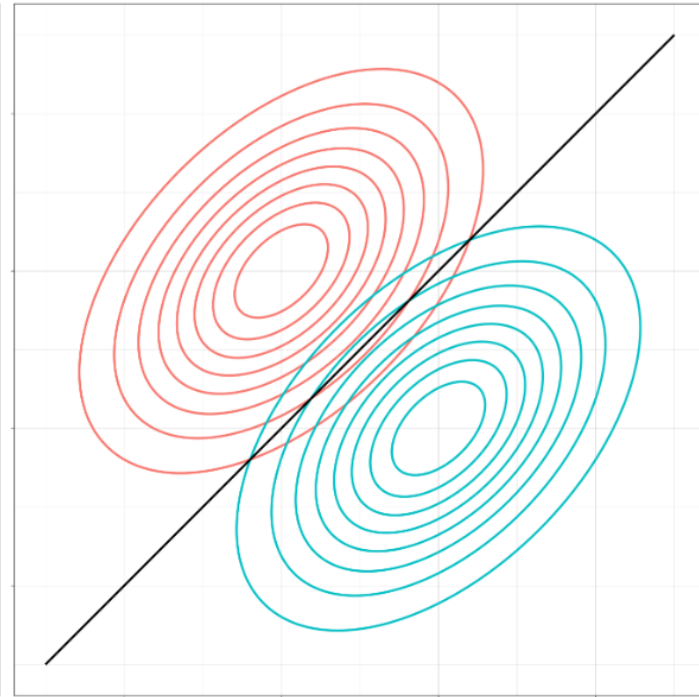
- We get a **linear decision boundary**:  $bx + c \geq 0$

# Visualizing Decision Boundaries

- Are all the covariance matrices modeled separately?
- ◆ If **separate**, the decision boundaries are **quadratic**.
  - ◆ If **shared**, the decision boundaries are **linear**.



Quadratic decision boundary



Linear decision boundary

# Summary of GDA



➤ It is a generative approach, assuming the data modeled by

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

- ◆ Assume that  $p(\mathbf{x} | y)$  is a Gaussian distribution.
- ◆ The parameters are determined by maximum likelihood estimation.

➤ **Decision boundary**

- ◆ In general, it shows **quadratic functions**.
- ◆ **It is linear** under various assumptions about Gaussian covariance matrices.
  - Single arbitrary matrix, i.e., linear discriminant analysis
  - Single or multiple diagonal matrices



# Discussion on GDA

# Recap: Naïve Bayes

- Assume that each feature is **independent** give the class.

$$p(\mathbf{x} \mid y = k) = \prod_{i=1}^d p(x_i \mid y = k)$$

- When the likelihoods are Gaussian, how many parameters are necessary for Naïve Bayes classifiers?

- It is equivalent to assuming  $\Sigma_i$  is diagonal.

- ◆ The other parts in the covariance matrix are zero.

$$\begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_d^2 \end{bmatrix}$$

# Gaussian Naïve Bayes

➤ It assumes that the likelihoods are Gaussian.

$$p(x_i | y = k) = \frac{1}{\sqrt{2\pi}\sigma_{ik}} \exp \left[ \frac{-(x_i - \mu_{ik})^2}{2\sigma_{ik}^2} \right]$$

- ◆ It is just 1-dim Gaussian, one for each input dimension.
- ◆ Model the same as GDA with **diagonal covariance matrix**.

➤ Maximum likelihood estimate of parameters

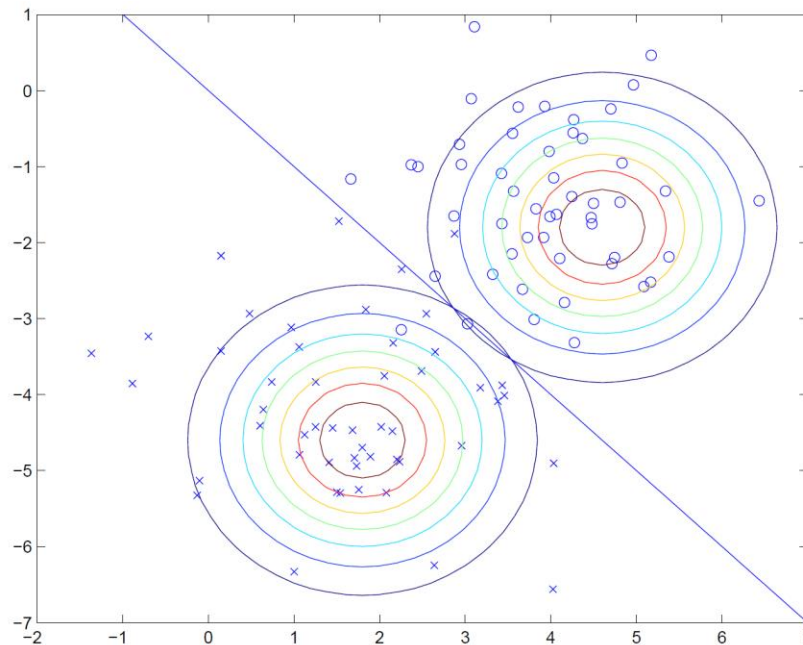
$$\mu_{ik} = \frac{\sum_{j=1}^n \mathbb{I}[y^{(j)} = k] \cdot x_i^{(j)}}{\sum_{j=1}^n \mathbb{I}[y^{(j)} = k]} = \frac{\sum_{j: y^{(j)} = k} x_i^{(j)}}{\text{\# of training samples in class } k}$$

$$\sigma_{ik}^2 = \frac{\sum_{j=1}^n \mathbb{I}[y^{(j)} = k] \cdot (x_i^{(j)} - \mu_{ik})^2}{\sum_{j=1}^n \mathbb{I}[y^{(j)} = k]} = \frac{\sum_{j: y^{(j)} = k} (x_i^{(j)} - \mu_{ik})^2}{\text{\# of training samples in class } k}$$



# Decision Boundary: Isotropic

- Same variance across all classes and input dimensions.
  - ◆ All class priors are equal.



- Classification only depends on **the distance to the mean.**

# GDA vs. Logistic Regression (LR)



- If assuming  $\Sigma_0 = \Sigma_1 = \Sigma$  in GDA, the conditional distribution for a linear decision boundary can be represented by

$$p(y | \mathbf{x}) = \sigma[bx + c] = \frac{1}{1 + \exp(-bx + c)}$$



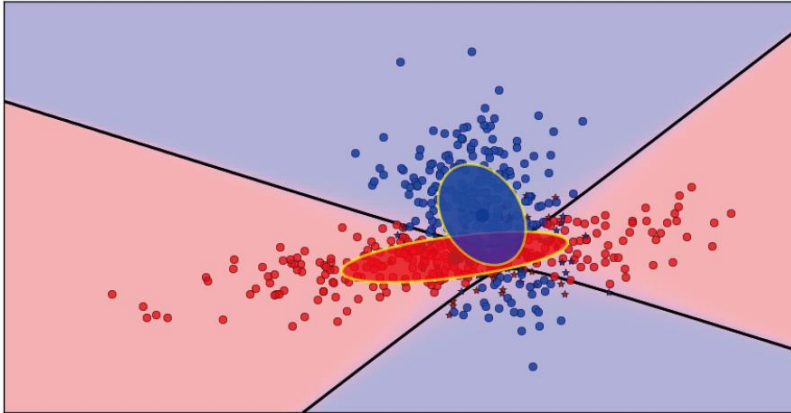
- It looks very similar to **logistic regression**.
  - ◆ Note that we only consider the **conditional probability, not the joint probability**.
- When should we prefer GDA to LR and vice versa?

# Various GDA vs. Logistic Regression



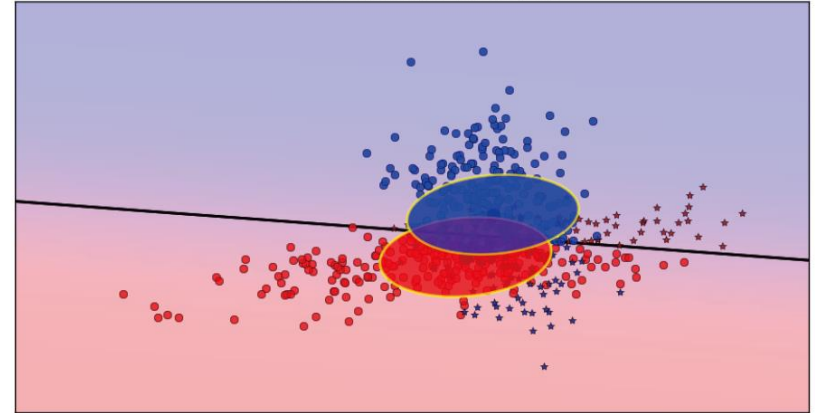
Covariance matrices are different.

Full Covariances (acc 0.805)

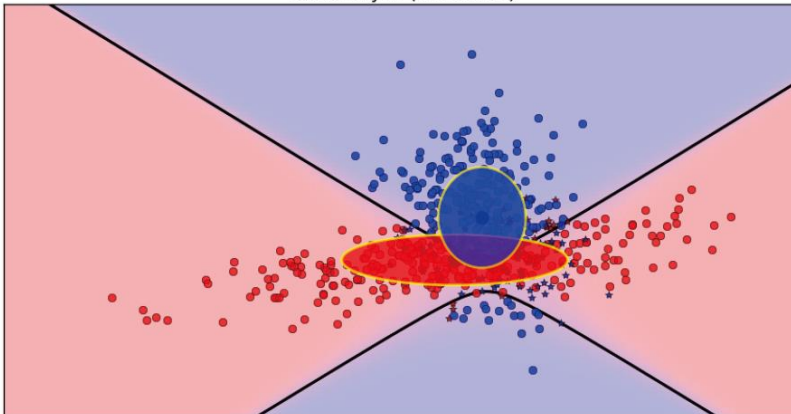


Covariance matrices are same.

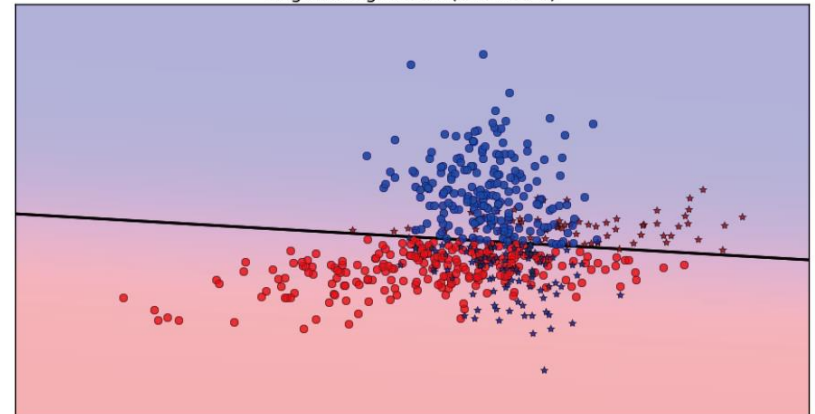
Shared Covariance (acc 0.717)



Naive Bayes (acc 0.780)



Logistic regression (acc 0.722)



Covariances are zero.

# Summary: GDA vs. LR

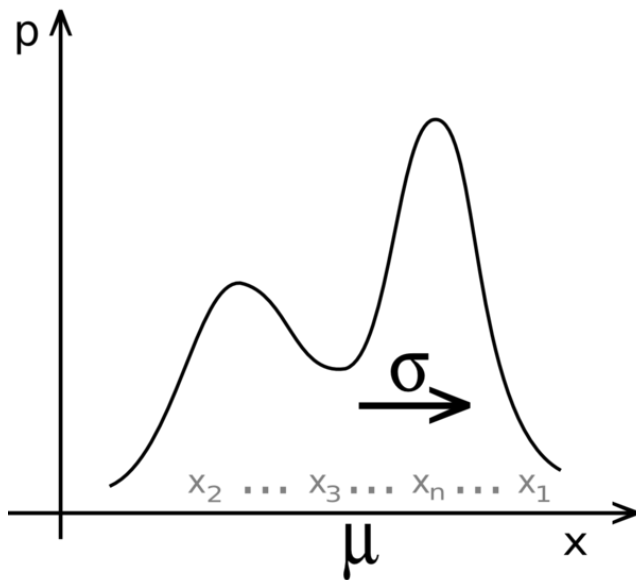
- **GDA makes a stronger assumption than LR.**
  - ◆ When data follows this assumption, GDA can be better than LR.
  - ◆ With shared covariance, it collapses to logistic regression similarly.
- **However, LR is more robust and less sensitive to incorrect modeling assumptions.**
  - ◆ When distributions are non-Gaussian, LR usually beats GDA.
- **The generative models tend to address a hard problem to solve an easy problem.**
  - ◆ It is useful if the assumption for data distribution holds.

# Q&A



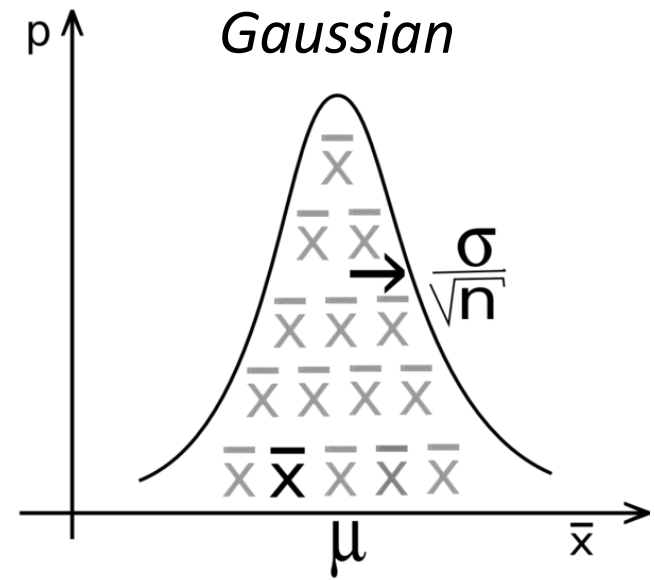
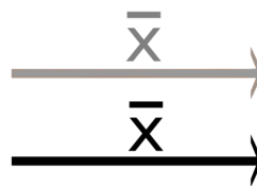
# Central Limit Theorem (CLT)

- The sampling distribution is approximately **normally distributed** if the sample size is large enough (e.g., 30).



Population  
Distribution

Samples of  
size  $n$



Sampling distribution of  
the mean