Machine Learning

Lecture 17b: Gaussian BC and Generative vs. Discriminative Classifier

Dr. Beilun Wang

Southeast University
School of Computer Science and Engineering

Course Content Plan

- ❑ Regression (supervised) — Y is a continuous
- ❑ Classification (supervised) — Y is a discrete
- ❑ Unsupervised models — NO Y
- ❑ Learning theory — About f()
- ❑ Graphical models — About interactions among X1,... Xp
- ❑ Reinforcement Learning — Learn program to Interact with its environment

Today: More Generative Bayes Classifiers

- Generative Bayes Classifier
- Naïve Bayes Classifier
- Gaussian Bayes Classifiers
  - Gaussian distribution
  - Naïve Gaussian BC
  - Not-naïve Gaussian BC → LDA, QDA
    - LDA: Linear Discriminant Analysis
    - QDA: Quadratic Discriminant Analysis
- Extra: Discriminative vs. Generative classifier

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

Generative Bayes Classifier

Task
↓
Representation
↓
Score Function
↓
Search/Optimization
↓
Models, Parameters

classification
↓
Prob. models p(X|C)

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

↓
EPE with 0-1 loss → MAP Rule
↓
Many options
↓
Prob. Models' Parameter

Gaussian Naive

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

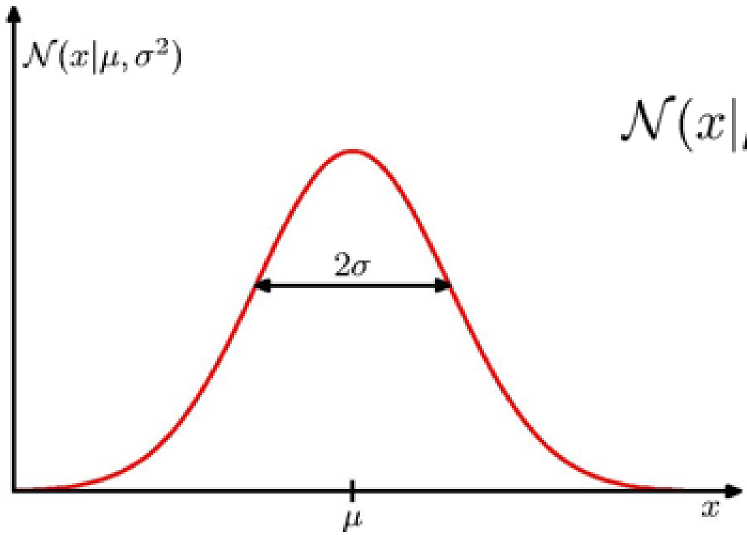
Bernoulli Naïve

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

Multinomial

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

Review: Single-Variate Gaussian Distribution



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

$$X \sim N\left(\mu,\sigma^2\right)$$

Multivariate Normal (Gaussian) PDFs

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

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

- 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

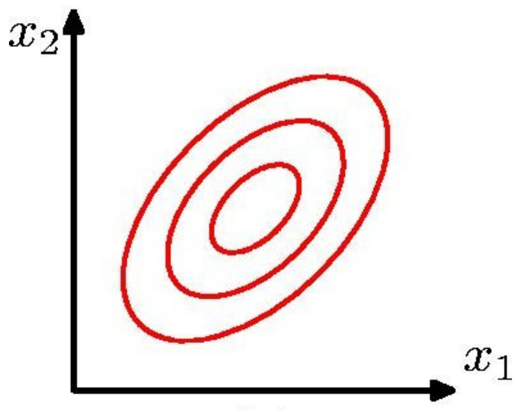
$$f(x_1, x_2) = \frac{1}{(2\pi)|\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1}(\vec{x}-\vec{\mu})}$$

$$\text{with} \quad \vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \text{and}$$

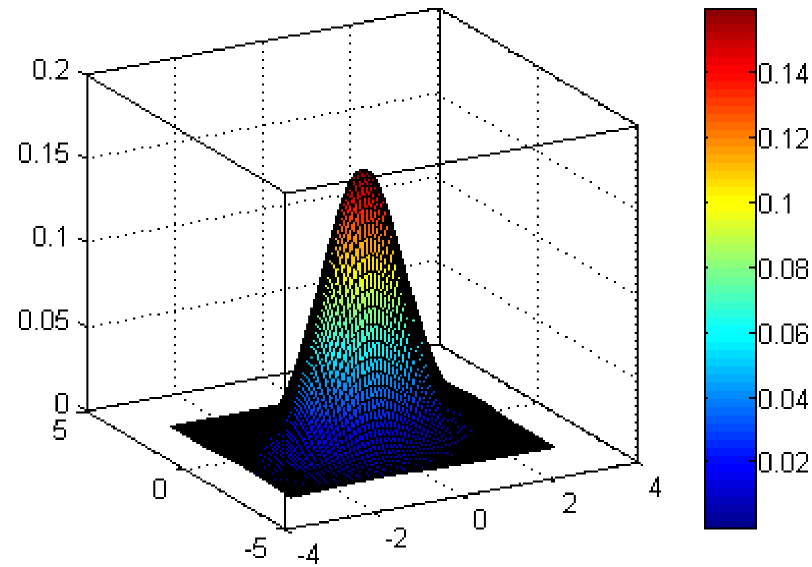
$$\underset{2\times 2}{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}_{2\times 2}$$

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

Bi-Variate Gaussian Distribution



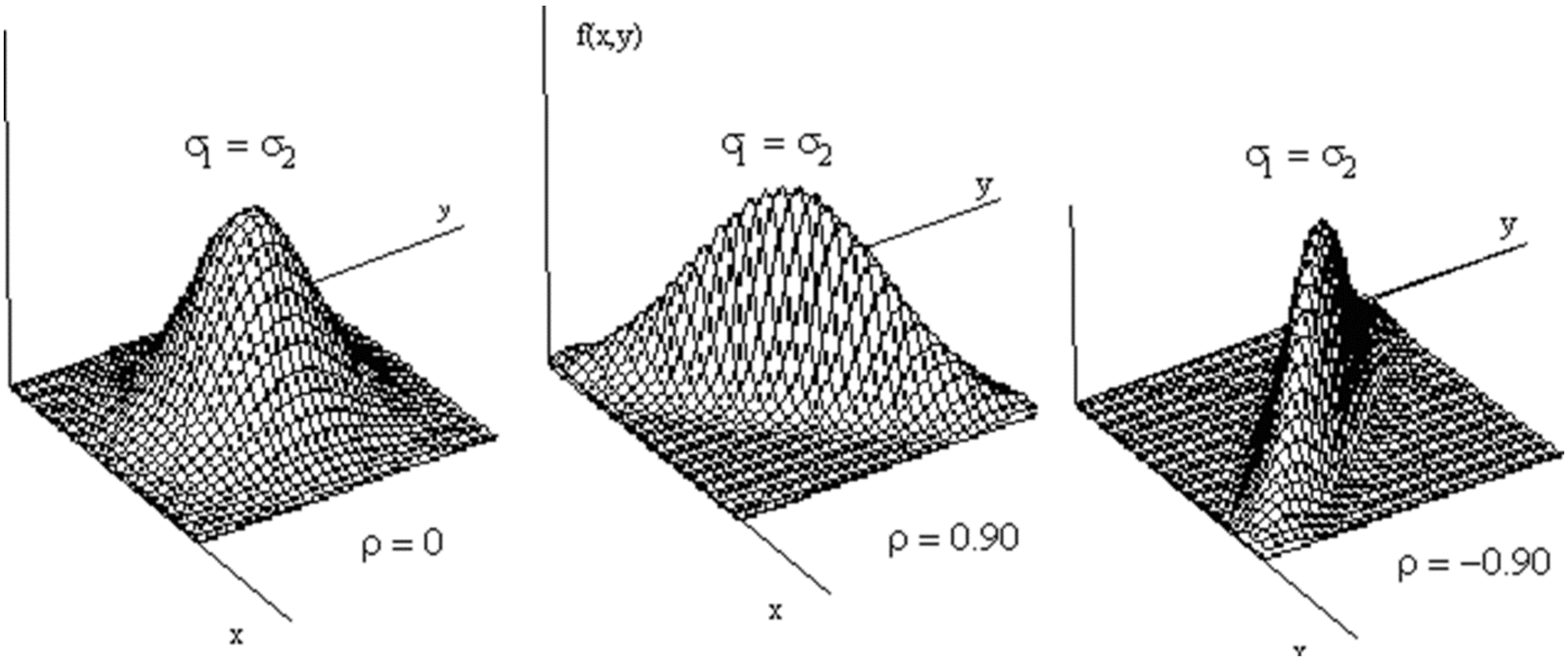
Bivariate normal PDF



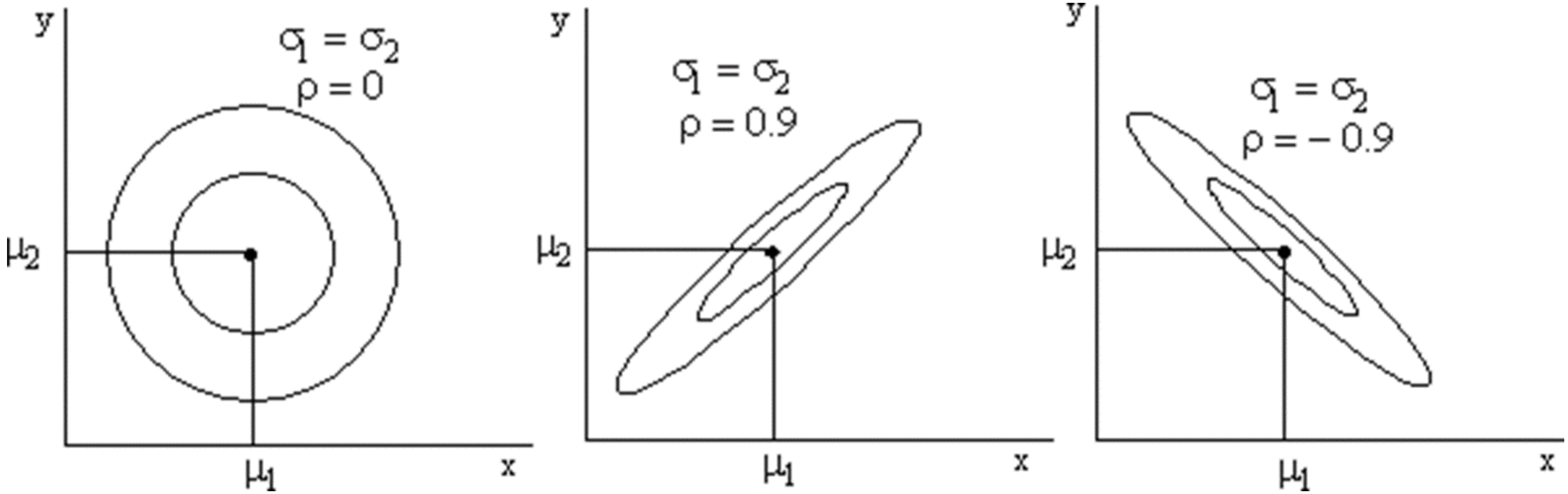
- Mean of normal PDF is at peak value. Contours of equal PDF form ellipses.

- The covariance matrix captures linear dependencies among the variables

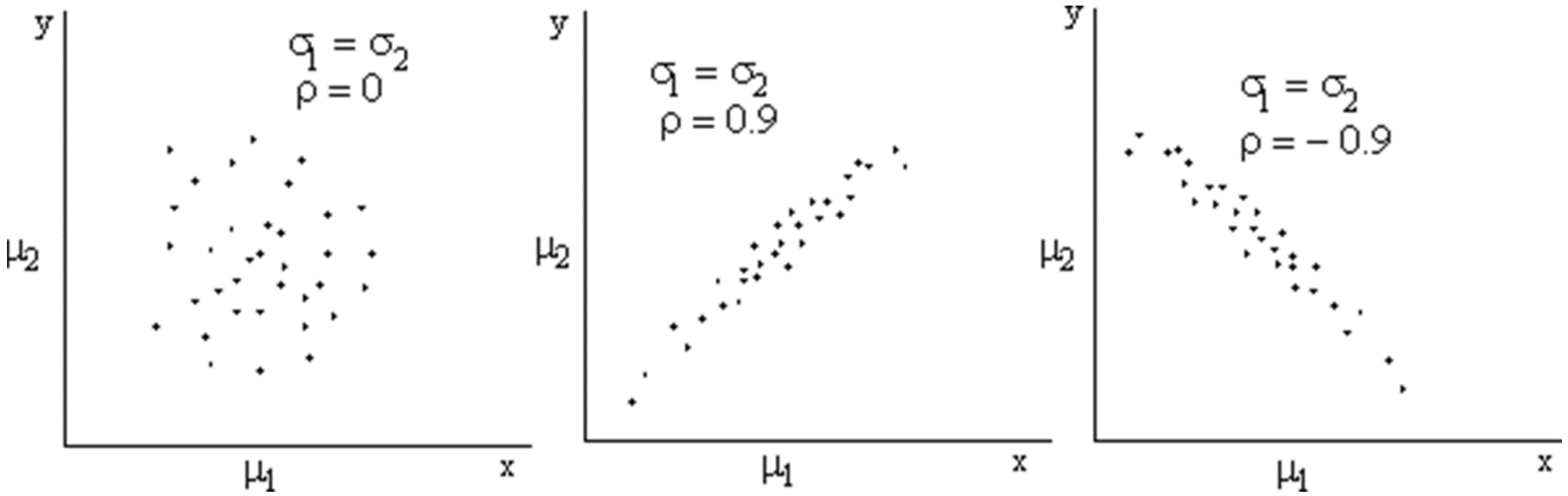
Surface Plots of the bivariate Normal distribution



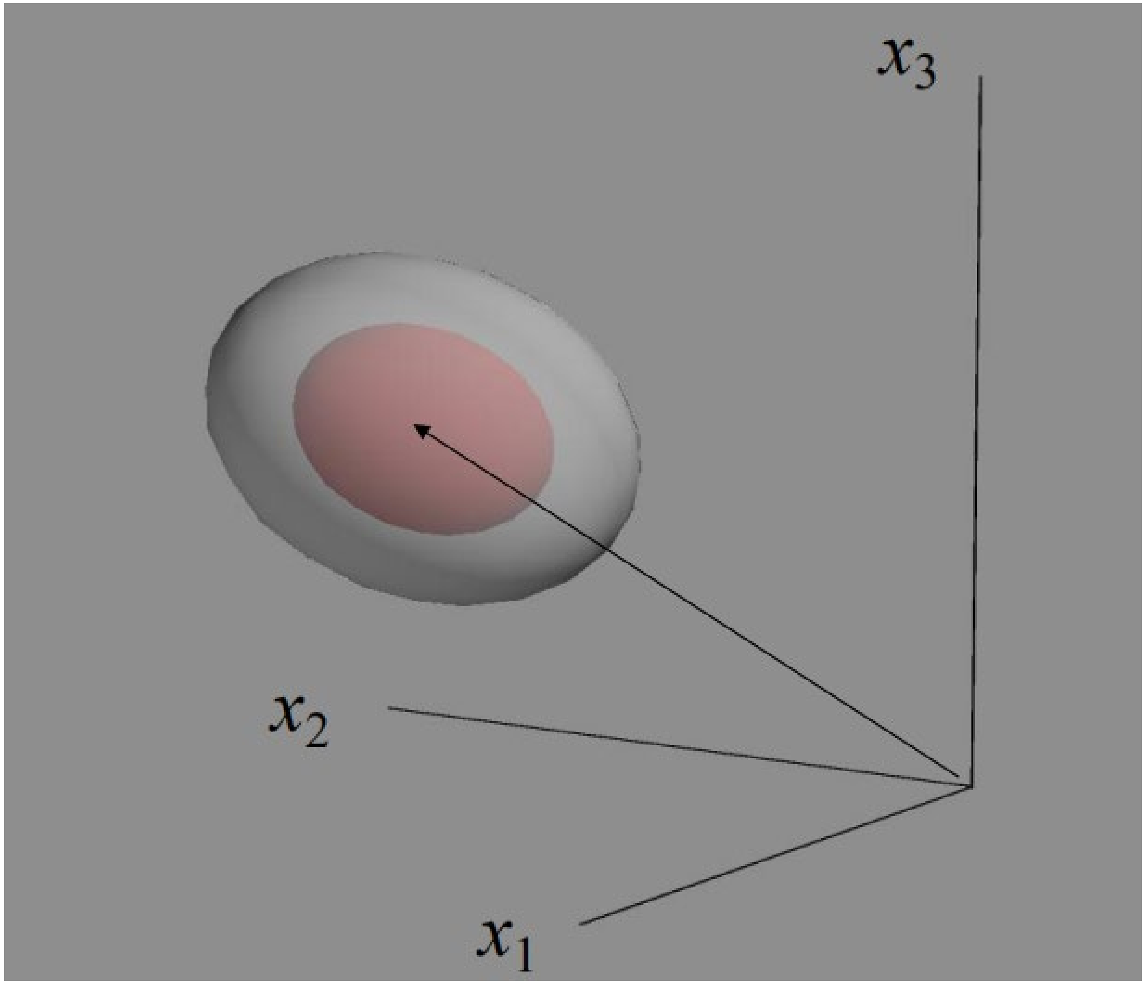
Contour Plots of the bivariate Normal distribution



Scatter Plots of the bivariate Normal distribution

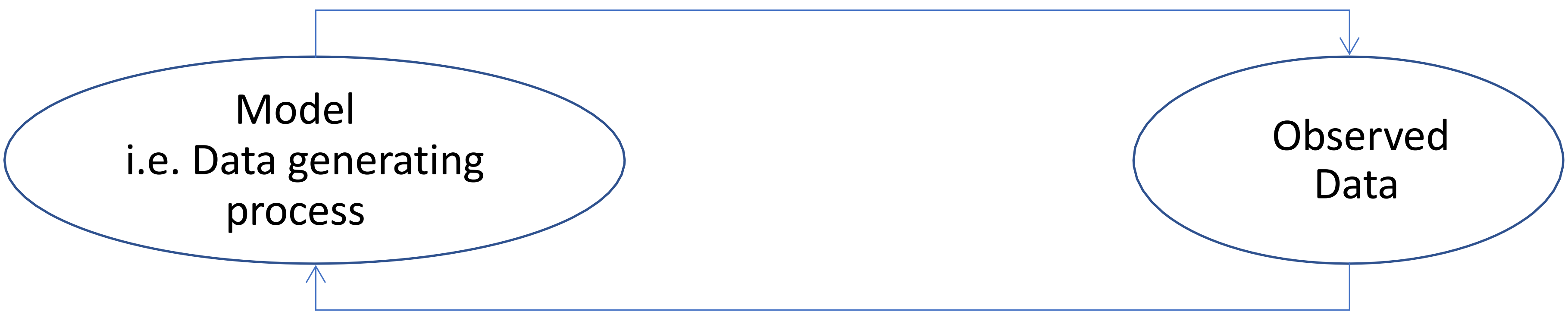


Trivariate Normal distribution



The Big Picture

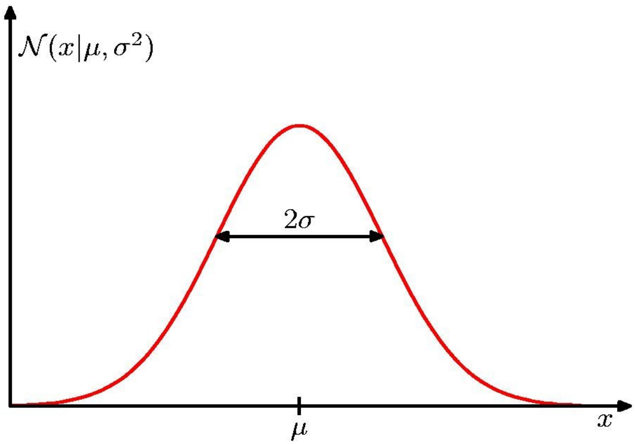
Probability



MLE Estimation / learning / statistics / Data mining

How to Estimate 1D Gaussian: MLE

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



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

How to Estimate p-D Gaussian: MLE

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

$$\vec{\mu} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_p \end{bmatrix} \qquad \Sigma_{p\times p} = \begin{bmatrix} var(X_1) & \dots & cov(X_1, X_p) \\ \vdots & \ddots & \vdots \\ cov(X_p, X_1) & \dots & var(X_p) \end{bmatrix}$$

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})\,(x_i - \hat{\mu})^T$$

Review: Generative BC

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

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

$P(\mathbf{x}|c_1)$ $P(\mathbf{x}|c_2)$ $P(\mathbf{x}|c_L)$

Generative Probabilistic Model for Class *1*

Generative Probabilistic Model for Class *2*

…

Generative Probabilistic Model for Class *L*

x_1 x_2 $\bullet\bullet\bullet$ x_p x_1 x_2 $\bullet\bullet\bullet$ x_p x_1 x_2 $\bullet\bullet\bullet$ x_p

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

Review: 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

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

Today: More Generative Bayes Classifiers

- Generative Bayes Classifier
- Naïve Bayes Classifier
- Gaussian Bayes Classifiers
  - Gaussian distribution
  - Naïve Gaussian BC
  - Not-naïve Gaussian BC → LDA, QDA
    - LDA: Linear Discriminant Analysis
    - QDA: Quadratic Discriminant Analysis
- Extra: Discriminative vs. Generative classifier

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

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

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

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 $C = c_i$

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: L different p-normal distributions and $P(C = c_i)$ $i = 1, \cdots, L$

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 $\mathrm{C} = c_i$

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: L different p-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

Gaussian Naïve Bayes Classifier

Naïve

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

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

Diagonal Matrix

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

Each class' covariance matrix is diagonal

11/6/19 Dr. Yanjun Qi / UVA CS

Today: More Generative Bayes Classifiers

- Generative Bayes Classifier
- Naïve Bayes Classifier
- Gaussian Bayes Classifiers
  - Gaussian distribution
  - Naïve Gaussian BC
  - Not-naïve Gaussian BC → LDA, QDA
    - LDA: Linear Discriminant Analysis
    - QDA: Quadratic Discriminant Analysis
- Extra: Discriminative vs. Generative classifier

Not-naïve Gaussian means?

Not Naïve

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

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

Naïve

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

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

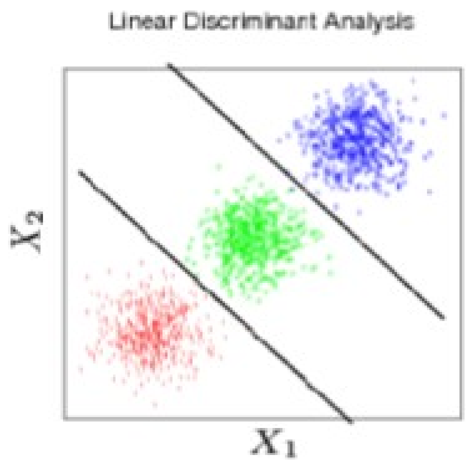
Diagonal Matrix $\Sigma\_c_k = \Lambda\_c_k$

Each class' covariance matrix is diagonal

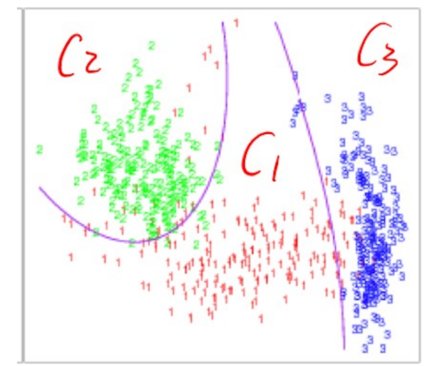
11/6/19 Dr. Yanjun Qi / UVA CS

Not-naïve Gaussian BC

- LDA: Linear Discriminant Analysis



- QDA: Quadratic Discriminant Analysis



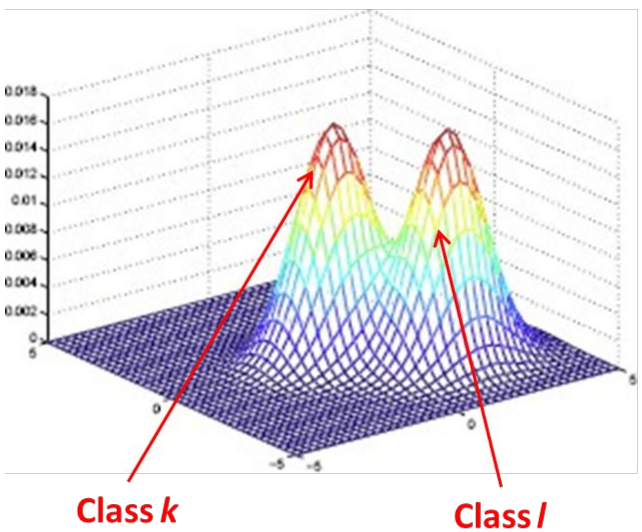
covariance matrix are the same across classes

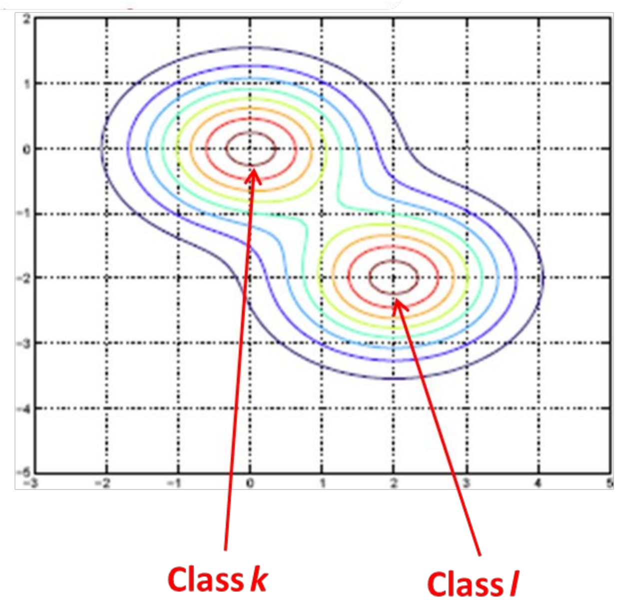
- LDA (Linear Discriminant Analysis)

Linear Discriminant Analysis : $\Sigma_k = \Sigma, \ \forall k$

Each class' covariance matrix is the same

The Gaussian Distribution are shifted versions of each other





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

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

Decision Boundary Points satisfying:

$$P(C_i|X) = P\left(C_j \middle| X\right)$$

$$\frac{P(C_i|X)}{P(C_j|X)} = 1 \Rightarrow log\, \frac{P(C_k|K)}{P(C_k|K)} = 0$$

$$\arg\max_k P(C_k|X) = \arg\max_k P(X, C_k) = \arg\max_k P(X|C_k)P(C_k)$$

$$= \arg\max_k \log\{P(X|C_k)P(C_k)\}$$

$$= arg\max_k logP(x|C_k) + logP(C_k) \Rightarrow \pi_k$$

Decision Boundary Points

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

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

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

Decision Boundary Points of LDA classifier ➔

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

The above is derived from the following :

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

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

Decision Boundary Points of LDA classifier ➔

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

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

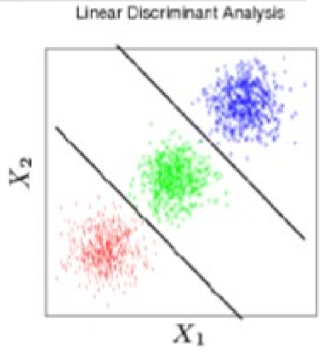
b

a

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

LDA Classification Rule

- Also called as Linear discriminant function



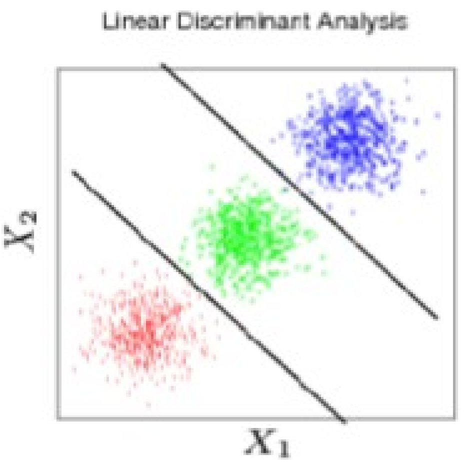
$$\underset{k}{\operatorname{argmax}} P(C_k \mid X) = \underset{k}{\operatorname{argmax}} P(X, C_k) = \underset{k}{\operatorname{argmax}} P(X \mid C_k) P(C_k)$$

$$\begin{aligned} &= \arg\max_k \Big[\cancel{-\log((2\pi)^{p/2}|\Sigma|^{1/2})} \\ &\quad -\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k) + \log(\pi_k) \Big] \\ &= \arg\max_k \boxed{\left[-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k) + \log(\pi_k) \right]} \end{aligned}$$

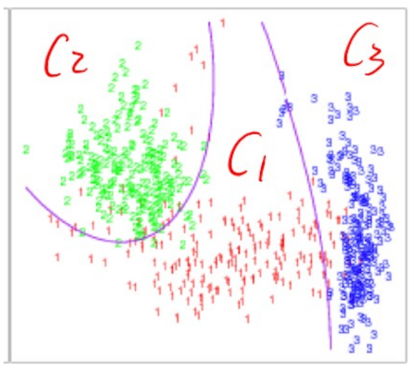
Linear Discriminant Function for LDA

Not-naïve Gaussian BC

- LDA: Linear Discriminant Analysis



- QDA: Quadratic Discriminant Analysis



If covariance matrix are not the same

- QDA (Quadratic Discriminant Analysis)

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

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

  - Classification rule:

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

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

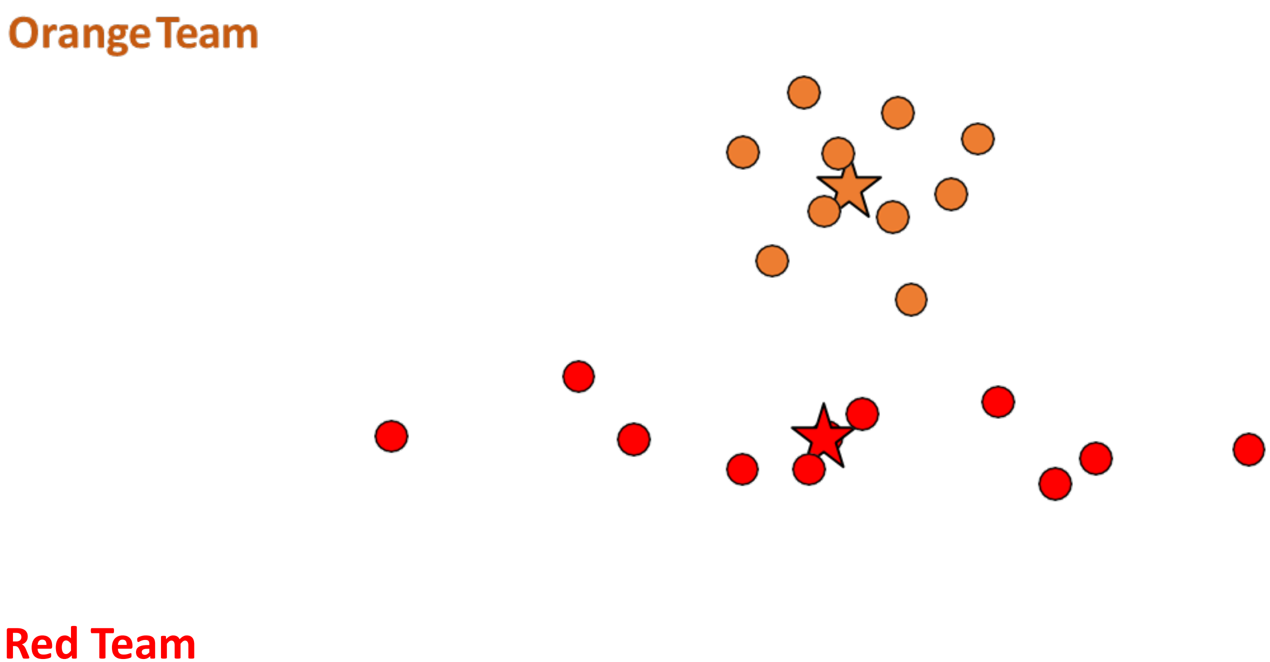
Regularized Discriminant Analysis

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

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

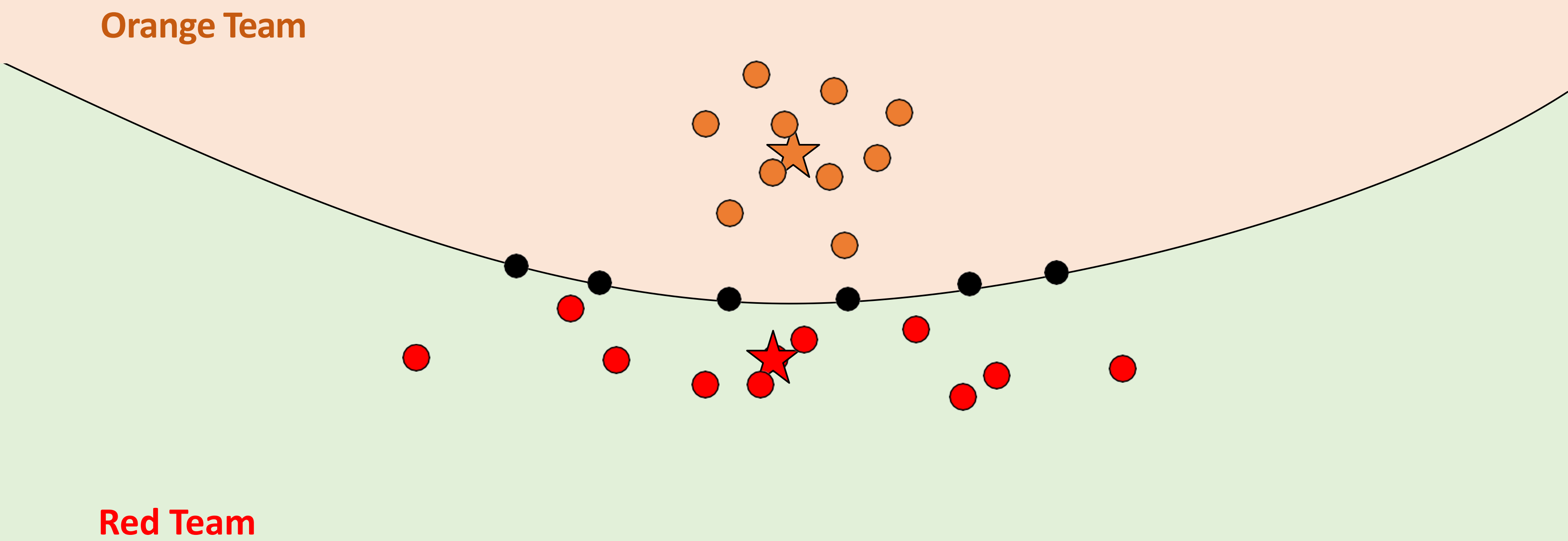
- The quadratic discriminant function $\delta_k(x)$ is defined using the shrunken covariance matrices $\hat{\Sigma}_k(\alpha)$.
- The parameter α controls the complexity of the model.

More: Decision Boundary of Gaussian naïve Bayes Classifiers?



Naïve Gaussian Bayes Classifier is not a linear classifier!

Gaussian Naïve Bayes Classifier



Naïve Gaussian Bayes Classifier is not a linear classifier!

Today: More Generative Bayes Classifiers

- Generative Bayes Classifier
- Naïve Bayes Classifier
- Gaussian Bayes Classifiers
  - Gaussian distribution
  - Naïve Gaussian BC
  - Not-naïve Gaussian BC → LDA, QDA
    - LDA: Linear Discriminant Analysis
    - QDA: Quadratic Discriminant Analysis
- Extra: Discriminative vs. Generative classifier

Discriminative vs. Generative

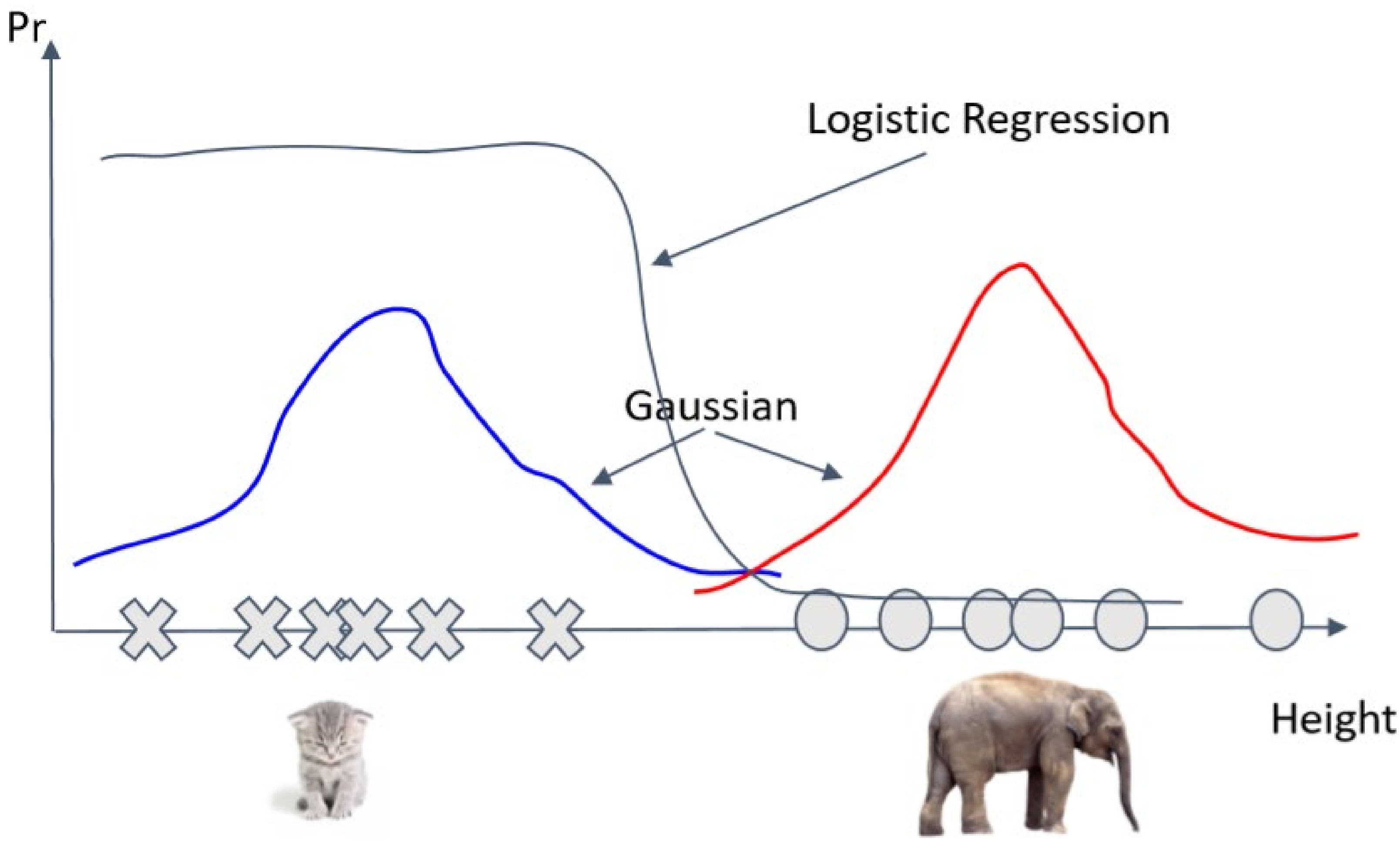
- Generative approach
  - Model the joint distribution p(X, C) using p(X | C = c_k) and p(C = c_k)

- Discriminative approach
  - Model the conditional distribution p(c| X) directly

e.g.

$$P(C = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 * X))}}$$

Discriminative vs. Generative



LDA vs. Logistic Regression

LDA (Generative model)

- Assumes Gaussian class-conditional densities and a common covariance
- Model parameters are estimated by maximizing the full log likelihood, parameters for each class are estimated independently of other classes, K p + $\frac{p(p+1)}{2} + (K-1)$ parameters
- Makes use of marginal density information $\Pr(x)$
- Easier to train, low variance, more efficient if model is correct
- Higher asymptotic error, but converges faster

Logistic Regression (Discriminative model)

- Assumes class-conditional densities are members of the (same) exponential family distribution
- Model parameters are estimated by maximizing the conditional log likelihood, simultaneous consideration of all other classes, $(K-1)(p+1)$parameters
- Ignores marginal density information $\Pr(x)$
- Harder to train, robust to uncertainty about the data generation process
- Lower asymptotic error, but converges more slowly

LDA vs. Logistic Regression

- Discriminative classifier (Logistic Regression)
  - Smaller asymptotic error
  - Slow convergence ~ O(p)

- Generative classifier (Naive Bayes)
  - Larger asymptotic error
  - Can handle missing data (EM)
  - Fast convergence ~ O(lg(p))

the speed at which a convergent sequence approaches its limit is called the rate of convergence.

Summary: Discriminative vs. Generative

- Empirically, generative classifiers approach their asymptotic error faster than discriminative ones
  - Good for small training set
  - Handle missing data well (EM)

- Empirically, discriminative classifiers have lower asymptotic error than generative ones
  - Good for larger training set

References

- https://qiyanjun.github.io/2019f-UVA-CS6316-MachineLearning/
- Prof. Tan, Steinbach, Kumar’s “Introduction to Data Mining” slide
- Prof. Andrew Moore’s slides
- Prof. Eric Xing’s slides
- Prof. KeChen NB slides qHastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Thanks for listening