INFSCI 2915: Machine Learning
Classification – Multiple Classes, LDA & QDA

Mai Abdelhakim

School of Computing and Information

610 IS Building

Spring 2018

Discriminant Analysis for Classification

- Linear Discriminant Analysis
- Quadratic Discriminant Analysis

Discriminant Analysis

- **Approach**: Model the **distribution of each class** separately, then use Bayes Rule

- Recall Bayes Classifier: Assign a new observation with features x_0 to class k that has largest Pr (Y= k|X= x_0)

Pr( ) and P( ) both stands for probability; the sign "|" reflects conditional (information is given)

- Both Linear and Quadratic Discriminant analysis **use normal (Gaussian) distribution to model features in each class** ➔ P(X|y) is Gaussian

- More popular than logistic regression when there is more than 2 classes

Bayes Theorem

$$\Pr(Y=k|X=x) = \frac{\Pr(X=x|Y=k)\cdot\Pr(Y=k)}{\Pr(X=x)}$$

X contains Features (e.g., number of words in an email). Y is the class label (e.g. Y=0 means not spam, Y=1 means spam email).

- Another way to write the equation above is:

$$\Pr(Y=k|X=x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K}\pi_l f_l(x)}$$

$K: number\ of\ classes$

$k: is\ a\ class\ label\ \{1,2,\ldots K\}$

$f_k(x) = \Pr(X=x|Y=k)$: is the density of x given that it is an observation from class k

$\pi_k = \Pr(Y=k)$: is the prior probability of class k

Linear Discriminant Analysis

- Recall that in **logistic regression** we assumed that: $\Pr(Y = k|X = x) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$

- However, LDA assumes that the density function of the **features** in each class is **normal (i.e. Gaussian)**

  - Density functions of the **features** in each class are assumed to **have the same variance** (in one feature case) or **covariance** (in multiple features case)

Linear Discriminant Analysis (LDA) with One Feature

- Assume multiple classes, and one feature (p=1)
- With one feature, the Gaussian density is given by:

$$f_k(x) = \Pr(X = x|Y = k) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2}$$

μ_k *is the mean of* x *in class* k *(class specific mean) and* σ_k^2 *is the variance of* x *in class* k

- In LDA, we assume **equal variance in all classes** $\sigma_k^2 = \sigma^2$
- Using training data, we estimate μ_k, σ_k, π_k for each class k to get:

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Note: This term is not important while comparing classes, as it is the same for all classes

Use Training Data for Estimation

- The mean can be estimated by the average of all training observations from the k^{th} class

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

- The variance can be estimated as the weighted average of variances of all k classes

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

- Prior of class k is estimated as the training observations n_k that belong to the k^{th} class divided by the total number of observations *n*

$$\hat{\pi}_k = n_k/n.$$

- **Build classifier:** have an estimation for $\Pr(Y = k|X = x)$

- **Prediction**: assign X to class k with largest $\Pr(Y = k|X = x)$

  - Equivalent to **finding the class** that gives the **largest discriminant score** given by:

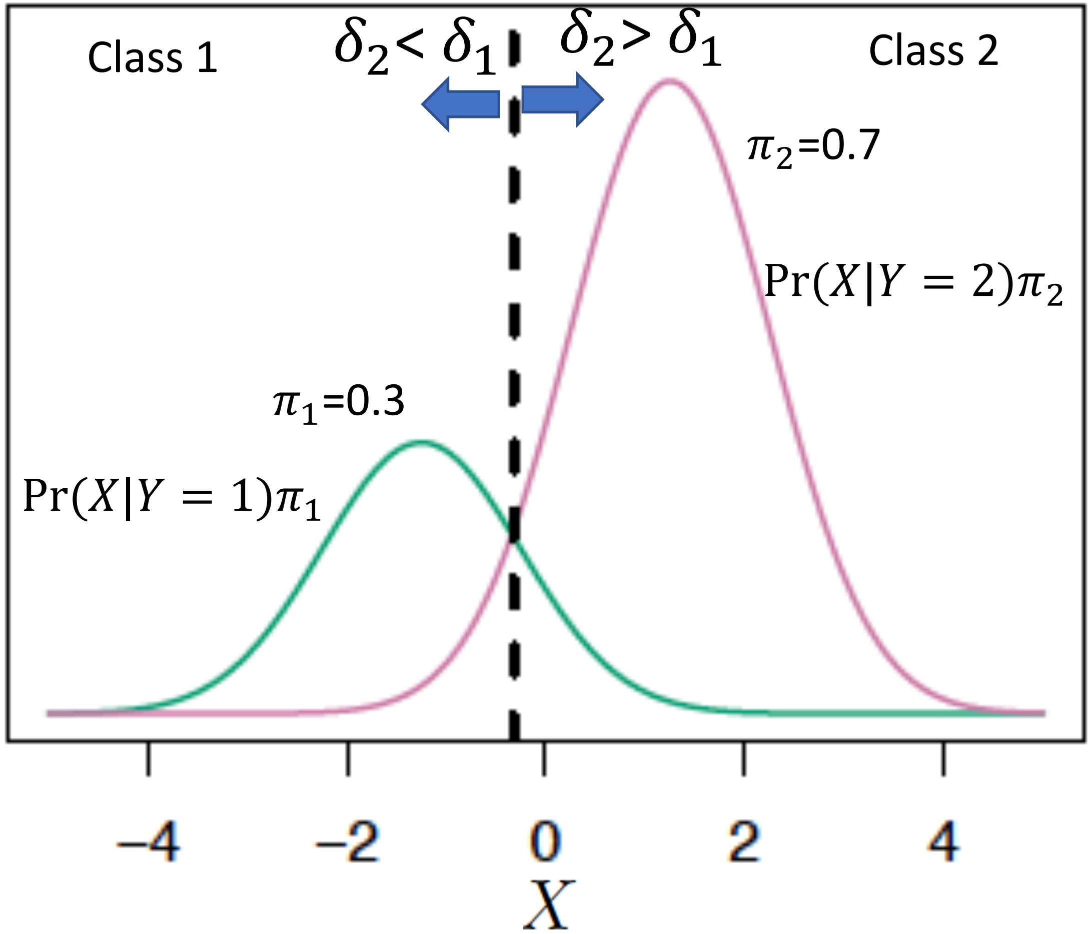
$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

  You can get this by taking log and discarding the terms that do not depend on k. Try it!

- The discriminant score is **linear function of x →** thus called **Linear Discriminant** Analysis

One Feature and Two Classes Example

Classification: new observation is assigned to class with highest $\Pr(Y = k|X = x)$



$$\pi_k = \Pr(Y = k)$$

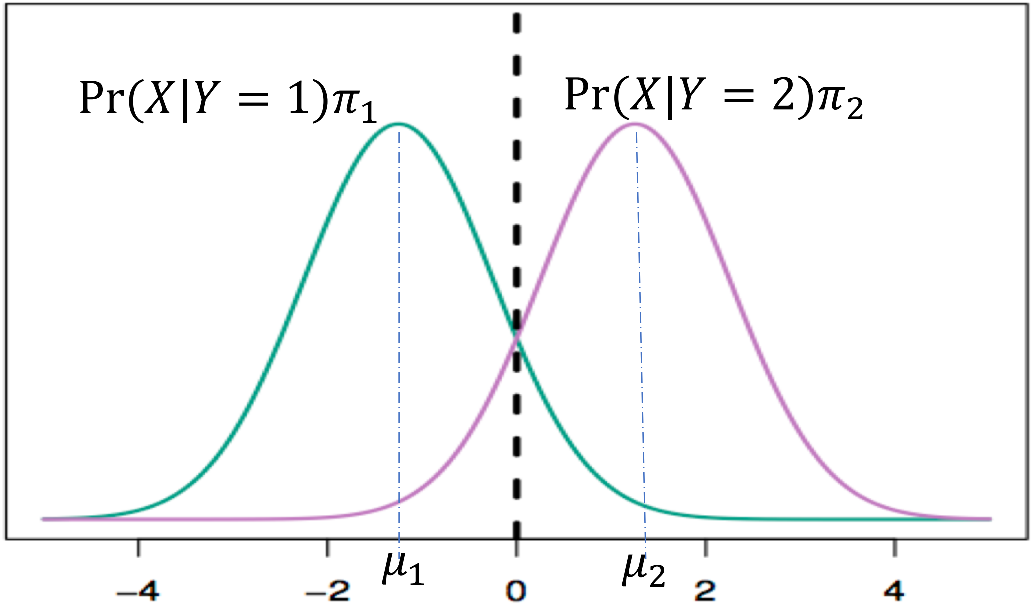
The priors can be estimated by finding the fraction of training samples that belong to each class

Special Case: Two classes (K=1,2), equal priors, one feature

- Assume equal priors $\pi_1 = \pi_2$=0.5, we can get a decision boundary as follows:

Try it!

Maximize $\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$ $\Rightarrow$ $x = \frac{\mu_1 + \mu_2}{2}$



This means that, for $\mu_2 > \mu_1$:

$$X \underset{\text{Class 1}}{\overset{\text{Class 2}}{\gtrless}} \frac{\mu_1+\mu_2}{2}$$

i.e.,

choose class 1 if X $< \frac{\mu_1+\mu_2}{2}$

and class 2 if X $> \frac{\mu_1+\mu_2}{2}$

Linear Discriminant Analysis When P>1

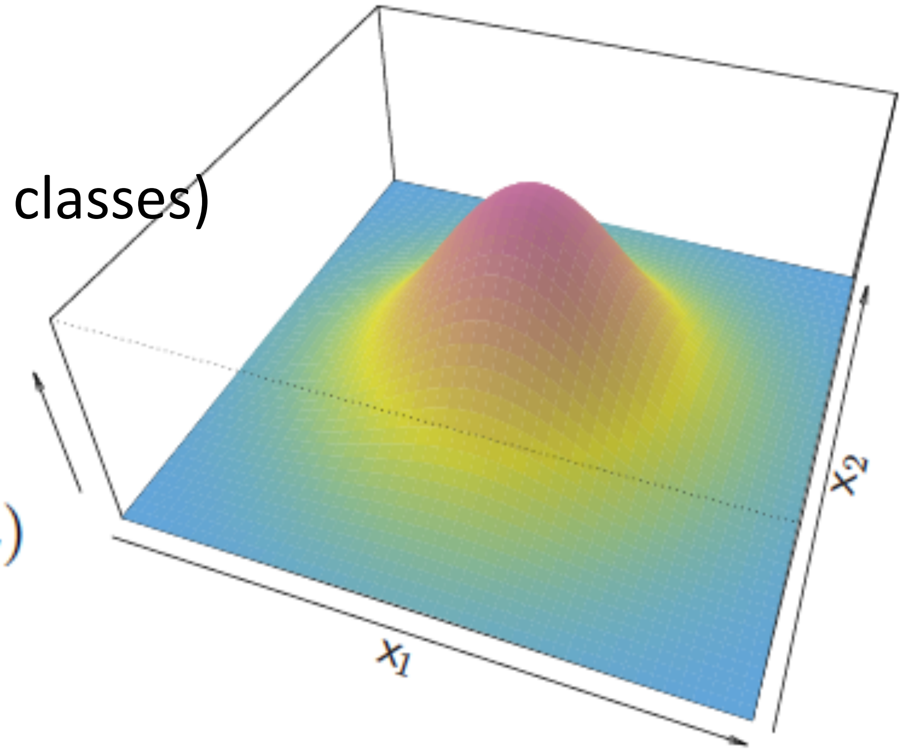
- If X is contains multiple features(p > 1), same approach is used except that the density function *f(x)* is modeled using the multivariate normal density, i.e.,

  $X \sim \mathcal{N}(\mu_k, \Sigma)$,

  - μ_k is the px1 mean vector
  - Σ is the pxp covariance matrix (LDA assume same for all classes)

- The multivariate normal density of mean μ *and covariance* Σ is given by

$$f(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)}$$



Linear Discriminant Analysis When P>1

- In this case, the discriminant functions will take the form

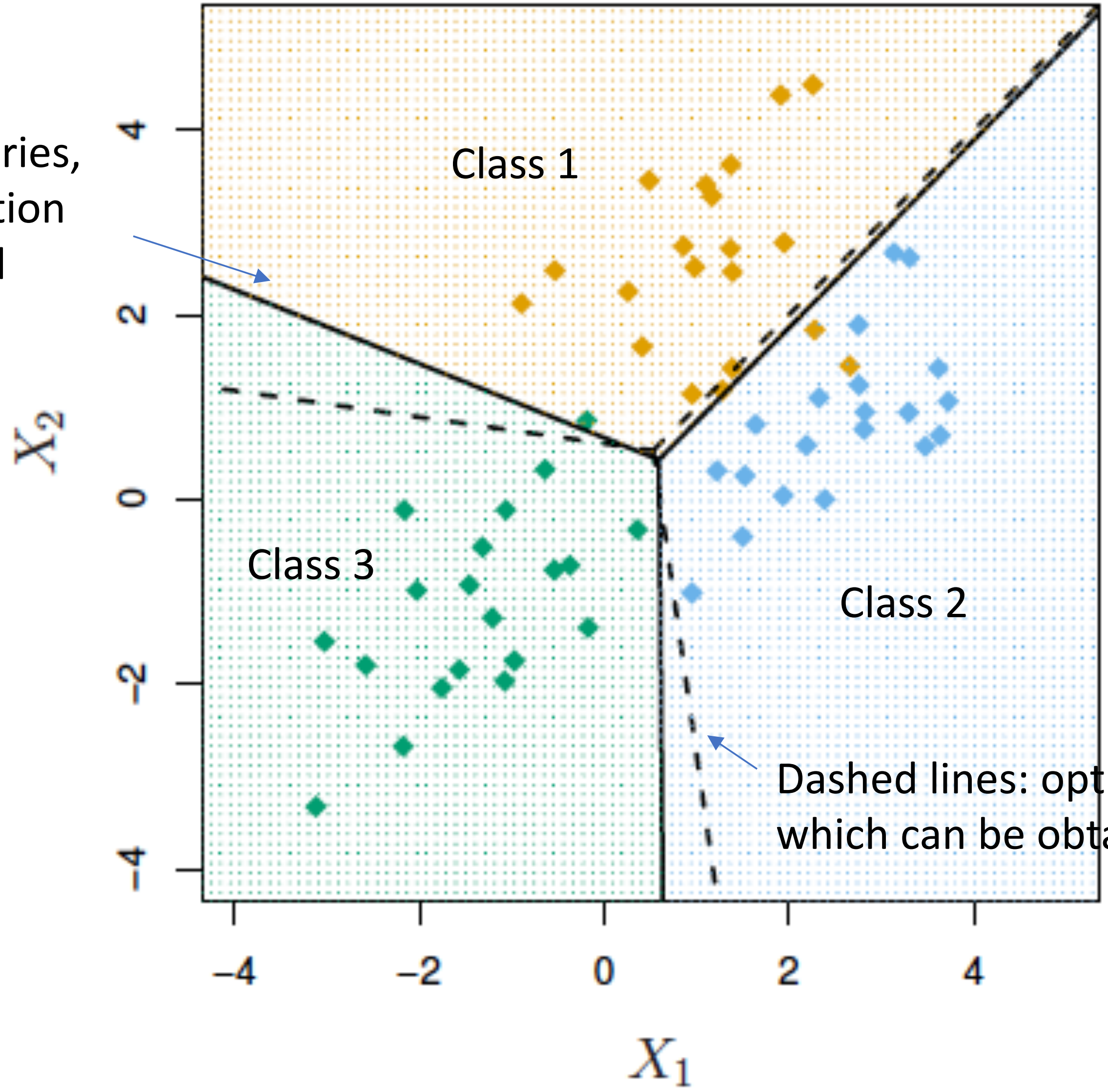
$$\delta_k(x) = x^T \boldsymbol{\Sigma}^{-1} \mu_k - \frac{1}{2}\mu_k^T \boldsymbol{\Sigma}^{-1} \mu_k + \log \pi_k$$

- Still LINEAR with feature vector x

Example: classify Iris flower to one of three possible species (K=3):
Setosa (Y=1), Versicolor (Y=2), Virginica (Y=3)
Using features: X=[sepal length, sepal width, petal length, petal width]
Build classifier: From the training data, we estimate mean and covariance matrix of this feature vector when Y=1, 2 and 3 = > find discriminant functions / decision boundaries
Classify: Apply the discriminant function to classify new observations

Illustration: Assume 3 classes (K=3), and 2 features

Solid lines: LDA decision boundaries, obtained with Gaussian distribution with estimated priors, mean and covariance



Dashed lines: optimal Bayes decision boundary, which can be obtained if actual distributions are known

Observation points are drawn from multivariate Gaussian distribution with p=2 and common covariance matrix

Quadratic Discriminant Analysis (QDA)

- In QDA, the density $f_k(x)$ is assumed to also be Gaussian for each class, but each **class has a different covariance matrix $\boldsymbol{\Sigma_k}$**

- In this case we get discriminate functions that are **quadratic** with x (hence the name)

$$\begin{aligned}\delta_k(x) &= -\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log\pi_k \\ &= -\frac{1}{2}x^T\mathbf{\Sigma}_k^{-1}x + x^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log\pi_k\end{aligned}$$

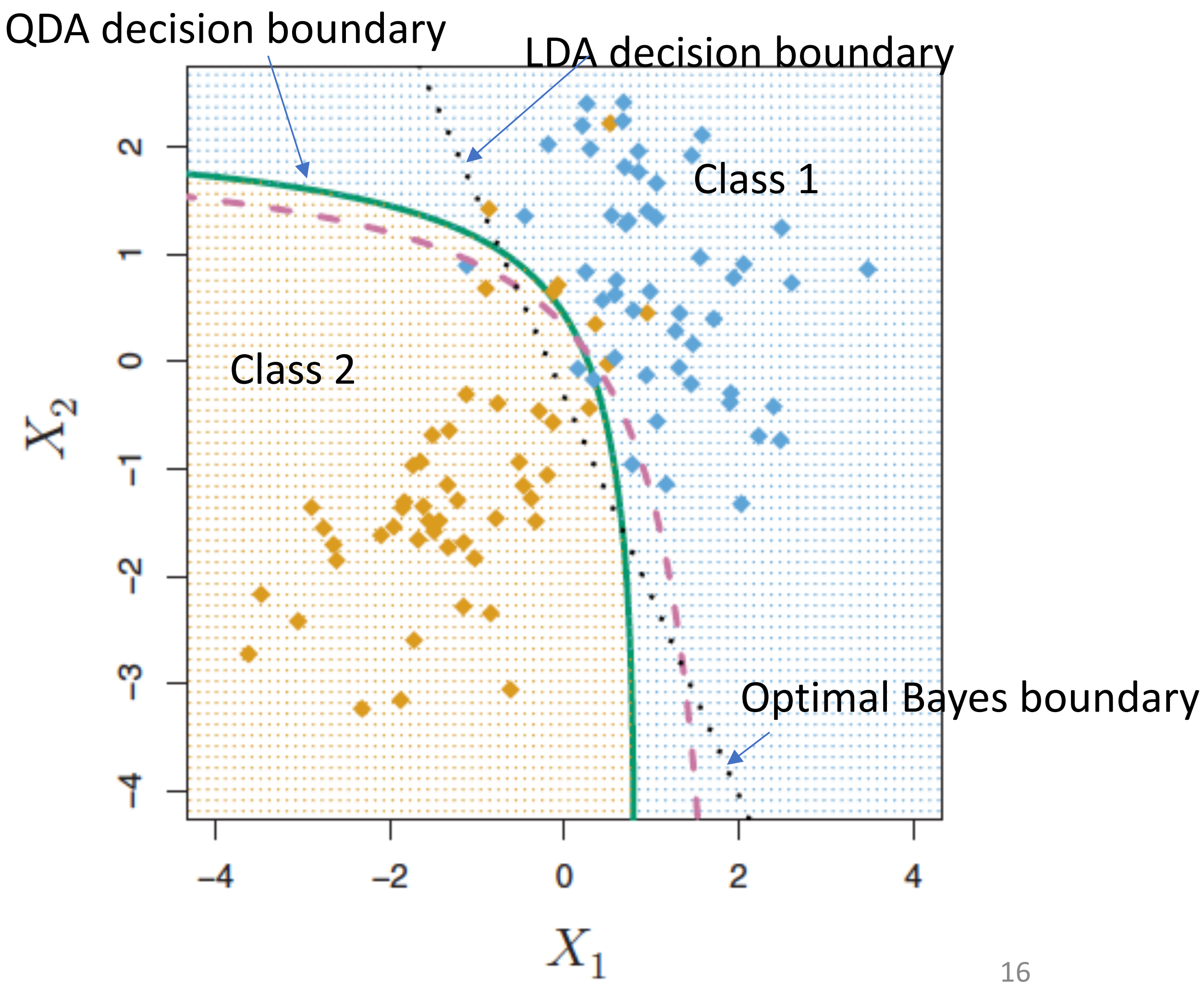
- If p=2, how a

Quadratic Discriminant Analysis – Quadratic Decision Boundary

- Here, 2 features, and 2 classes
  - Two classes have different covariance in this example, hence QDA is better
- QDA, the covariance matrix is estimated for each class
  - Each matrix has p.(p+1)/2 parameters
  - Need large training data to avoid overfitting (high variance)

Bias-variance trade-off!

- LDA is simpler, but could lead to high bias
  - Estimate single covariance matrix



Naïve Bayes Classifier

- Assumes:
  - Gaussian densities (same as LDA and QDA)
  - each class has its **own covariance** (same as QDA),
  - <u>Covariance matrix of each class is diagonal matrix</u> (<u>features are statistically independent</u>)
    - Now we need to estimate only p parameters for each covariance matrix

- Due to the statistical independence, we have

$$f_k(x) = \prod_{j=1}^{p} f_{jk}(x_j)$$

Density function of feature j in class k

LDA and QDA in Python

- Linear Discriminant Analysis (LDA)

from **sklearn.discriminant_analysis** import **LinearDiscriminantAnalysis**

LDAmodelFitted = LinearDiscriminantAnalysis().fit(X_train, Y_train)

- Quadratic Discriminant Analysis

from **sklearn.discriminant_analysis** import **QuadraticDiscriminantAnalysis**

QDAmodelFitted = **QuadraticDiscriminantAnalysis().**fit(X_train, Y_train)

Exercise

Find the accuracy of LDA and QDA classifiers when they are applied to classify breast cancer as malignant or benign using the sklearn data set.