Introduction to Artificial Intelligence

Bayes Classification and Linear Discriminant Analysis

Lionel Fillatre

2021-2022

Outline of the Lecture

- Introduction
- Naïve Bayes: Principle
- Naïve Bayes: Discrete Case
- Naïve Bayes: Continuous Case
- Linear Discriminant Analysis
- Model Validation
- Conclusion

Introduction

Background

- There are three methods to establish a classifier

  a) Model a classification rule directly

  Examples: k-NN, decision trees, perceptron, SVM, Linear Discriminant Analysis

  b) Model the probability of class memberships given input data

  Example: logistic regression, perceptron with the cross-entropy cost

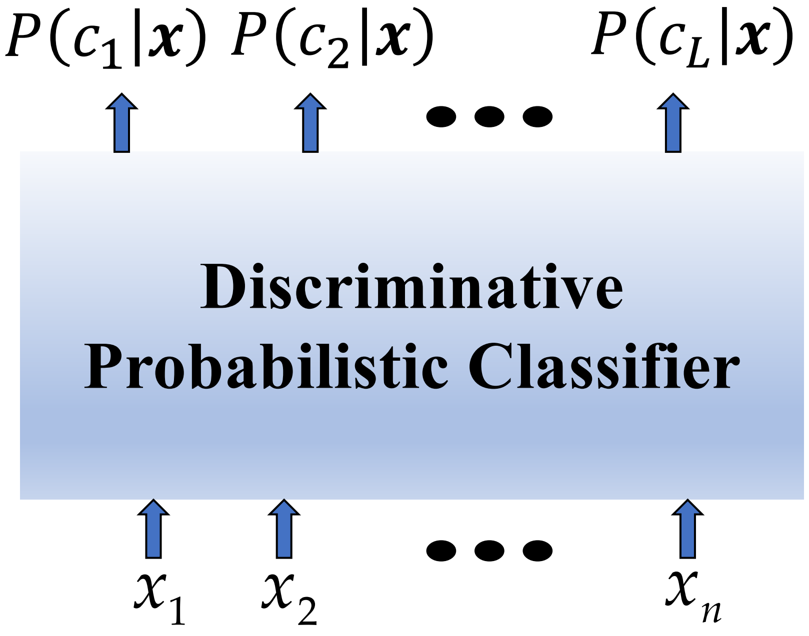
  c) Make a probabilistic model of data within each class

  Examples: naive Bayes, hypothesis testing

- *a*) and *b*) are examples of discriminative classification
- *c*) is an example of generative classification
- *b*) and *c*) are both examples of probabilistic classification

Probabilistic Classification

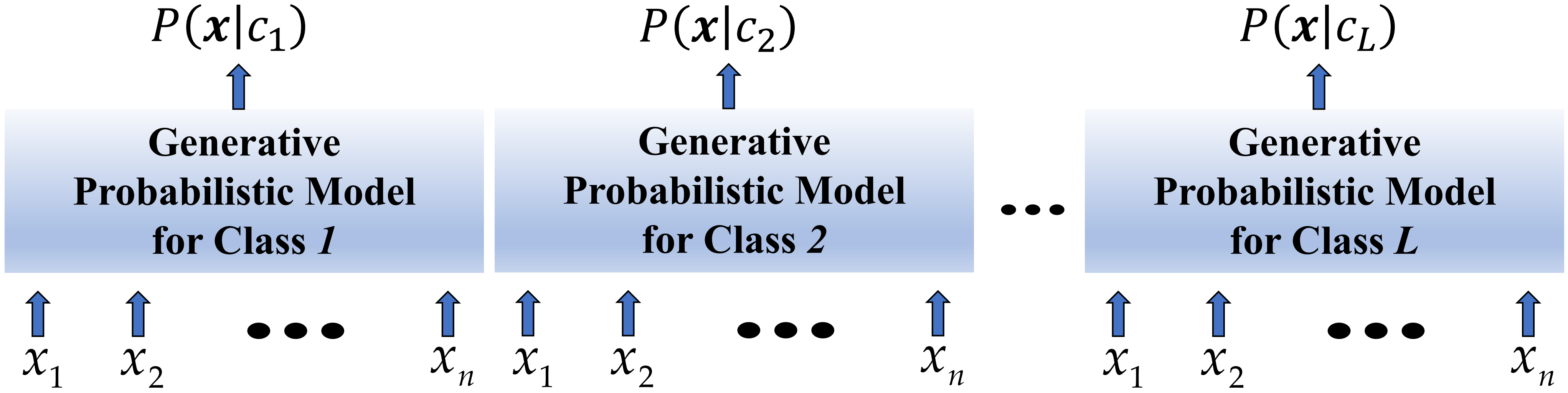
- Establishing a probabilistic model for classification
  - Discriminative model: $P(C|\boldsymbol{X}), \mathrm{C} \in \{c_1, \dots, c_L\}, \boldsymbol{X} = (X_1, \dots, X_n)$



$\boldsymbol{x} = (x_1, \dots, x_n)$: data, features, etc.

Probabilistic Classification

- Establishing a probabilistic model for classification
  - Generative model: $P(\boldsymbol{X}|C), \mathrm{C} \in \{c_1, \dots, c_L\}, \boldsymbol{X} = (X_1, \dots, X_n)$



$$\boldsymbol{x} = (x_1, \dots, x_n)$$

Naïve Bayes: Principle

Probability Basics

- Prior, conditional and joint probability for random variables X and C
  - Prior probability: $P(C)$
  - Conditional probability: $P(\boldsymbol{X}|C), P(C|\boldsymbol{X})$
  - Joint probability: $V = (\boldsymbol{X}, C)$, $P(V)$
  - Relationship: $P(\boldsymbol{X}, C) = P(\boldsymbol{X}|C)P(C) = P(C|\boldsymbol{X})P(\boldsymbol{X})$
  - Independence: $P(\boldsymbol{X}|C) = P(\boldsymbol{X})$, $P(C|\boldsymbol{X}) = P(C)$, $P(\boldsymbol{X}, C) = P(\boldsymbol{X})P(C)$

- Bayesian Rule
  - $P(C|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C)\times P(C)}{P(X)}$
  - Posterior = $\frac{\text{Likelihood}\times\text{Prior}}{\text{Evidence}}$

Probability Basics

- We have two independent six-sided dice. When they are tolled, it could end up with the following occurrence: (*A*) dice 1 lands on side "3", (*B*) dice 2 lands on side "1", and (*C*) Two dice sum to eight.

  1) $P(A) = \frac{1}{6}$

  2) $P(B) = \frac{1}{6}$

  3) $P(C) = \frac{5}{36}$

  4) $P(A, B) = \frac{1}{36}$

  5) $P(A|B) = \frac{1}{6}$

  6) $P(A, C) = \frac{1}{36}$

  7) $P(C|A) = \frac{1}{6}$

  8) *Is* $P(A, C)$ *equal to* $P(A)P(C)$?

MAP Rule

- MAP classification rule minimizes the Bayes decision error
  - **MAP**: **M**aximum **A** **P**osterior
  - Assign $\boldsymbol{x}$ to c^*, i.e., $Z = c^*$, if

$$P(C = c^* | \boldsymbol{X} = \boldsymbol{x}) > P(C = c | \boldsymbol{X} = \boldsymbol{x}), \qquad c^* \neq \mathrm{c}, \qquad \mathrm{c} \in \{c_1, \ldots, c_L\}$$

- Generative classification with the MAP rule
  - Apply Bayesian rule to calculate posterior probabilities for all c_i

$$P(C = c_i | \boldsymbol{X} = \boldsymbol{x}) = \frac{P(\boldsymbol{X} = \boldsymbol{x} | C = c_i) P(C = c_i)}{P(\boldsymbol{X} = \boldsymbol{x})} \propto P(\boldsymbol{X} = \boldsymbol{x} | C = c_i) P(C = c_i)$$

  - Then apply the MAP rule

Naïve Bayes

- Difficulty: learning the joint probability $P(X_1, \dots, X_n|C)$

$$P(C = c_i|\boldsymbol{X} = \boldsymbol{x}) \propto P(\boldsymbol{X} = \boldsymbol{x}|C = c_i)P(C = c_i) = P(X_1 = x_1, \dots, X_n = x_n|C = c_i)P(C = c_i)$$

- Naïve Bayes classification
  - **Assumption**: all input features are **conditionally independent**!

$$\begin{aligned} P(X_1, \dots, X_n|C) &= P(X_1|X_2, \dots, X_n, C)P(X_2, \dots, X_n|C) \\ &= P(X_1|\, C)P(X_2, \dots, X_n|C) \\ &= \cdots \\ &= P(X_1|\, C) \cdots P(X_n|\, C) \end{aligned}$$

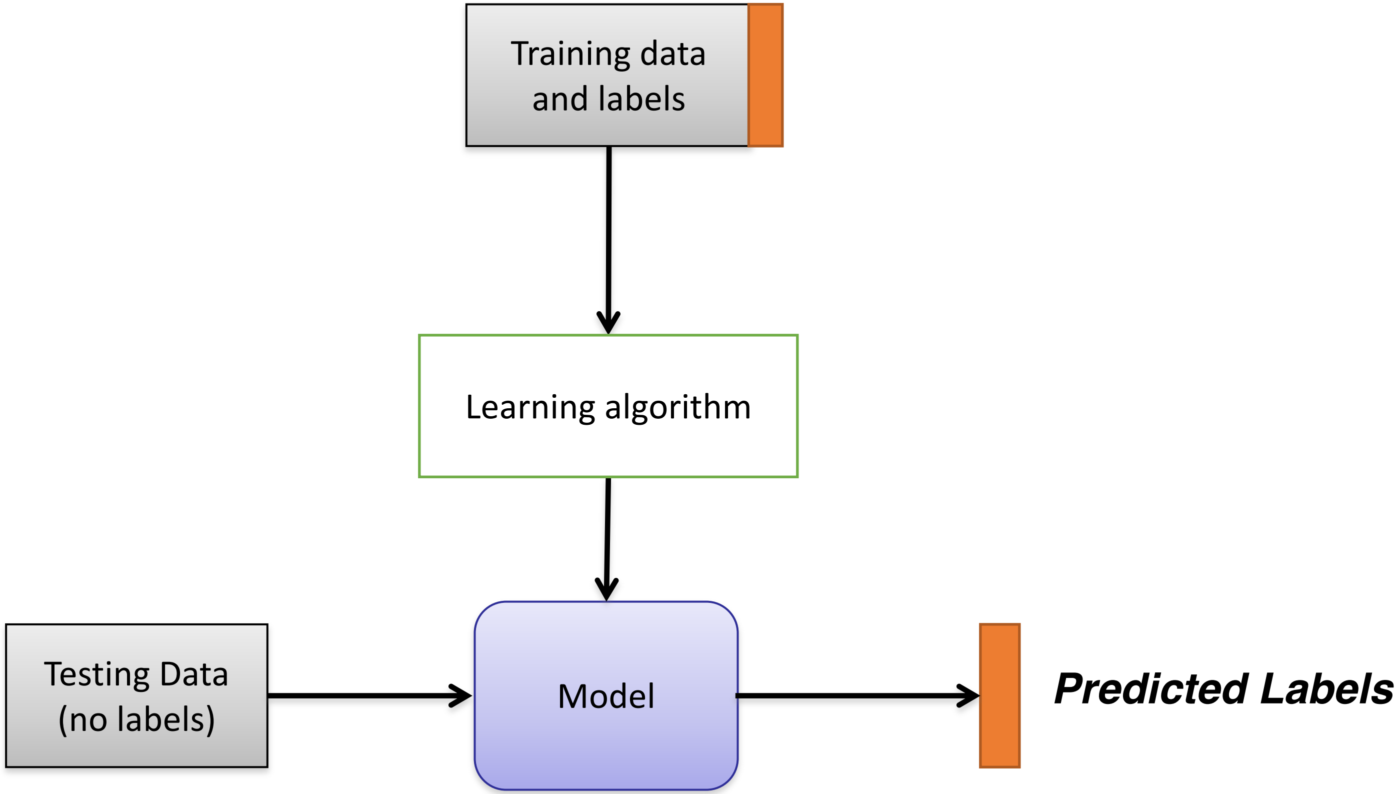
  - MAP classification rule

$$c^* = \underset{c \in \{c_1, \dots, c_L\}}{\operatorname{argmax}} \; P(X_1 = x_1|\, C = c) \cdots P(X_n = x_n|\, C = c)P(C = c)$$

    - In case of tie-breaking (several maxima), choose randomly the class

The Supervised Learning Pipeline

$$S = \{(\boldsymbol{X}_1, y_1), \dots, (\boldsymbol{X}_N, y_N)\}, y_i \in \{c_1, \dots, c_L\}$$



$$\boldsymbol{X} = (X_1, \dots, X_n)$$

Naïve Bayes: Discrete Case

$X_j \in \left\{a_{j1}, \dots, a_{jK_j}\right\}$ with K_j possible values for all j

- Given a training set $S = \{(\boldsymbol{X}_1, y_1),\dots, (\boldsymbol{X}_N, y_N)\}, y_i \in \{c_1, \dots, c_L\}$ with $\boldsymbol{X} = (X_1, \dots, X_n)$

- **Learning phase**
  - For each c_i, calculate an estimate $\hat{P}(C = c_i)$ of $P(C = c_i)$ from S
  - For every feature value a_{jk} of each feature X_j, calculate an estimate $\hat{P}\left(X_j = a_{jk} \middle| C = c_i\right)$ of $P\left(X_j = a_{jk} \middle| C = c_i\right)$ from S
  - Finally, we get $n \times L$ conditional probabilistic (generative) models and L estimates $\hat{P}(C = c_i)$

- **Test phase**
  - Given an unknown instance $\boldsymbol{x}' = (x_1', \dots, x_n')$
  - Assign the label c^* to $\boldsymbol{x}'$ such that

$$c^* = \underset{c \in \{c_1,\dots,c_L\}}{\operatorname{argmax}} \hat{P}(X_1 = x_1' |\, C = c) \cdots \hat{P}(X_n = x_n' |\, C = c)\hat{P}(C = c)$$

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

Example: Learning Phase

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

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

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

Example: Test Phase

- Given a new instance, predict its label

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

- Look up tables achieved in the learning phrase

P(Outlook=*Sunny*|Play=*Yes*) = 2/9
P(Temperature=*Cool*|Play=*Yes*) = 3/9
P(Humidity=*High*|Play=*Yes*) = 3/9
P(Wind=*Strong*|Play=*Yes*) = 3/9
P(Play=*Yes*) = 9/14

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

- Decision making with the MAP rule

P(*Yes*| $\boldsymbol{x}'$) ≈ [P(*Sunny*|Yes)P(*Cool*|*Yes*)P(*High*|Yes)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053
P(*No*| $\boldsymbol{x}'$) ≈ [P(*Sunny*|No) P(*Cool*|No)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

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

Naïve Bayes: Continuous Case

$X_j \in \mathbb{R}$ for all j

- Given a training set $S = \{(\boldsymbol{X}_1, y_1),\ldots, (\boldsymbol{X}_N, y_N)\}, y_i \in \{c_1, \ldots, c_L\}$
- **Learning phase**
  - For each c_i, calculate an estimate $\hat{P}(C = c_i)$ of $P(C = c_i)$ from S
  - Conditional probability of each feature X_j often modeled with the normal distribution

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

    - μ_{ji}: mean of feature X_j of examples for which $C = c_i$
    - σ_{ji}: standard deviation of feature X_j of examples for which $C = c_i$
  - For every feature X_j, calculate some estimates $\hat{\mu}_{ji}$ of μ_{ji} and $\hat{\sigma}_{ji}^2$ of σ_{ji}^2 from S
  - Finally, we get $n \times L$ conditional probabilistic (generative) models and L estimates $\hat{P}(C = c_i)$
- **Test phase**
  - Given an unknown instance $\boldsymbol{x}' = (x_1', \ldots, x_n')$
  - Calculate $\hat{P}\left(X_j = x_j' \middle| C = c_i\right)$ for all j and all i
  - Assign the label c^* to $\boldsymbol{x}'$ such that $c^* = \underset{c \in \{c_1,\ldots,c_L\}}{\operatorname{argmax}} \hat{P}(X_1 = x_1' \mid C = c) \cdots \hat{P}(X_n = x_1' \mid C = c)\hat{P}(C = c)$

Example: Monitoring a System Temperature

- Two classes:
  - Yes: the system works well
  - No: the system has a failure
- Temperature is naturally a continuous value X:

  Yes: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8

  No: 27.3, 30.1, 17.4, 29.5, 15.1

- **Training Phase**: Estimate mean and variance for each class:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i\,, \quad \widehat{\sigma^2} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \hat{\mu})^2 \quad \text{give} \quad \begin{cases} \hat{\mu}_{\text{Yes}} = 21.64, \quad \hat{\sigma}_{\text{Yes}} = 2.35 \\ \hat{\mu}_{\text{No}} = 23.88, \quad \hat{\sigma}_{\text{No}} = 7.09 \end{cases}$$

- **Learning Phase**: output two Gaussian models for $\hat{P}(\text{temperature}|C)$ where $x' \in \mathbb{R}$ is the temperature to be tested

$$\hat{P}(x'|Yes) = \frac{1}{2.35\sqrt{2\pi}}\exp\left(-\frac{(x'-21.64)^2}{2\times 2.35^2}\right) = \frac{1}{2.35\sqrt{2\pi}}\exp\left(-\frac{(x'-21.64)^2}{11.09}\right)$$

$$\hat{P}(x'|No) = \frac{1}{7.09\sqrt{2\pi}}\exp\left(-\frac{(x'-23.88)^2}{2\times 7.09^2}\right) = \frac{1}{7.09\sqrt{2\pi}}\exp\left(-\frac{(x'-23.88)^2}{50.25}\right)$$

Linear Discriminant Analysis

Assumptions of LDA

- Predictor variable $X \in \mathbb{R}^n$ for each group is normally distributed.
- 5 underlying assumptions
  - There are L groups
  - Conditional density of X in group $C = k$ is
    $$f_k(x) = P(X = x|C = k)$$
  - π_k is the prior probability of group k such that $\sum_{k=1}^{L} \pi_k = 1$
  - Each group follows Multivariate Normal Distribution $\mathcal{N}(\mu_k, \Sigma_k)$
  - Common covariance matrix: $\Sigma_k = \Sigma$ for any $k = 1, \cdots, L$

Notations

- $p_k(X) = P(C = k|X = x)$: the *posterior* probability that an observation posterior $X = x$ belongs to the kth class.

- Objective: classifies an observation to class for which $p_k(X)$ is largest

- **Bayes' Theorem:**

$$P(C = k|X = x) = \frac{P(X = x|C = k)P(C = k)}{P(X = x)} = \frac{\pi_k f_k(x)}{\sum_{i=1}^{L} \pi_i f_i(x)}$$

LDA for $n = 1$ (single predictor)

- $f_k(x) == \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma^2}\right)$

- $p_k(x) = P(C = k|X = x) = \dfrac{\pi_k \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma^2}\right)}{\sum_{i=1}^{L} \pi_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma^2}\right)}$

- The Bayes classifier involves assigning an observation $X = x$ to class k for which $p_k(x)$ is largest.

- Taking the log of $p_k(x)$ and rearranging the terms, it is not hard to show that this is equivalent to assigning the observation to the class for which the value in $\delta_k(x)$ is largest:

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

- Linear log-odds function implies decision boundary between classes k and j —the set where $P(C = k|X = x)$ and $P(C = j|X = x)$—is linear in x; in n dimensions a hyperplane. This is of course true for any pair of classes, so all decision boundaries are linear.

Proof of LDA for $n = 1$

- The Bayes classifier involves assigning an observation $X = x$ to class k for which $p_k(x)$ is largest. We get

$$p_k(x) \geq p_j(x)$$

$$\frac{\pi_k \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma^2}\right)}{\sum_{i=1}^{L} \pi_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma^2}\right)} \geq \frac{\pi_j \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(x-\mu_j\right)^2}{2\sigma^2}\right)}{\sum_{i=1}^{L} \pi_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma^2}\right)}$$

$$\pi_k \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma^2}\right) \geq \pi_j \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(x-\mu_j\right)^2}{2\sigma^2}\right)$$

$$\pi_k \exp\left(-\frac{(x-\mu_k)^2}{2\sigma^2}\right) \geq \pi_j \exp\left(-\frac{\left(x-\mu_j\right)^2}{2\sigma^2}\right)$$

$$\log \pi_k - \frac{(x-\mu_k)^2}{2\sigma^2} \geq \log \pi_j - \frac{\left(x-\mu_j\right)^2}{2\sigma^2}$$

$$\log \pi_k - \frac{x^2 - 2x\mu_k + \mu_k^2}{2\sigma^2} \geq \log \pi_j - \frac{x^2 - 2x\mu_j + \mu_j^2}{2\sigma^2}$$

$$x \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k \geq x \frac{\mu_j}{\sigma^2} - \frac{\mu_j^2}{2\sigma^2} + \log \pi_j$$

$$\delta_k(x) \geq \delta_j(x)$$

Apply LDA

- LDA starts by assuming that each class has a **normal** distribution with a **common** variance
- The mean and variance are estimated
- Finally, Bayes' theorem is used to compute $p_k(x)$ and observation is assigned to class with maximum prob. among all k probabilities

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

- $\delta_k(x)$ is called the discriminant function

Use Training Data set for Estimation

- Mean μ_k could be estimated by average of all training observations from k^{th} class:

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

- Variance σ^2 could be estimated as weighted average of variances of all L classes:

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

- And, π_k is estimated as proportion of training observations that belong to k^{th} class:

$$\hat{\pi}_k = \frac{n_k}{n}$$

- Estimated discriminant function:

$$\hat{\delta}_k(x) = x\,\frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log \hat{\pi}_k$$

LDA for $n = 1$ and $L = 2$

- If $L = 2$ and $\pi_1 = \pi_2 = {}^1\!/_2$, then the Bayes classifier assigns an observation to class 1 if

$$\delta_1(x) = x\frac{\mu_1}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} + \log\pi_1 > \delta_2(x) = x\frac{\mu_2}{\sigma^2} - \frac{\mu_2^2}{2\sigma^2} + \log\pi_2$$

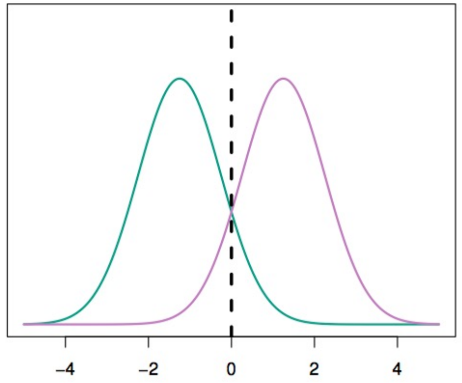
$$x(\mu_1 - \mu_2) > \frac{\mu_1^2 - \mu_2^2}{2}$$

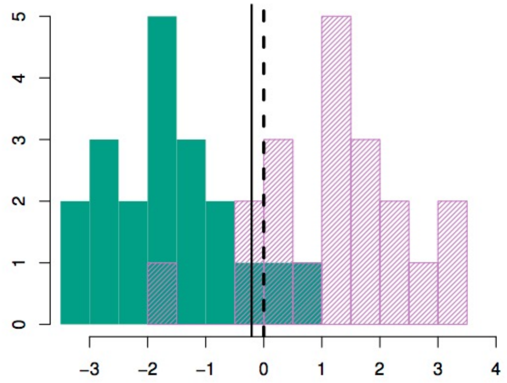
- The Bayes decision boundary corresponds to the points x where

$$x(\mu_1 - \mu_2) = \frac{\mu_1^2 - \mu_2^2}{2} = \frac{(\mu_1 - \mu_2)(\mu_1 + \mu_2)}{2}$$

$$x = \frac{\mu_1 + \mu_2}{2}$$

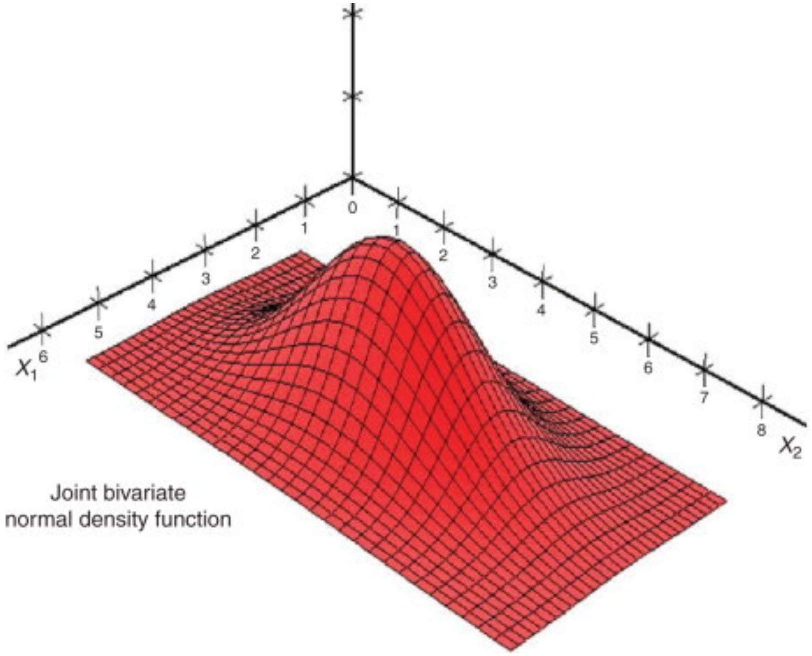
- 20 observations were drawn from each of the two classes
- The dashed vertical line is the Bayes decision boundary





An Example When $n > 1$

- If $X = (X_1, X_2, \dots, X_n)$ is multidimensional ($n > 1$), we use exactly same approach except density function $f(x)$ is modeled using multivariate normal probability density function (pdf)



LDA for $n > 1$ (Multiple predictors)

- Multivariate normal pdf: $f(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$

- Bayes classifier assigns an observation $X = x$ to the class for which the $P(C = k|X = x)$ is largest.

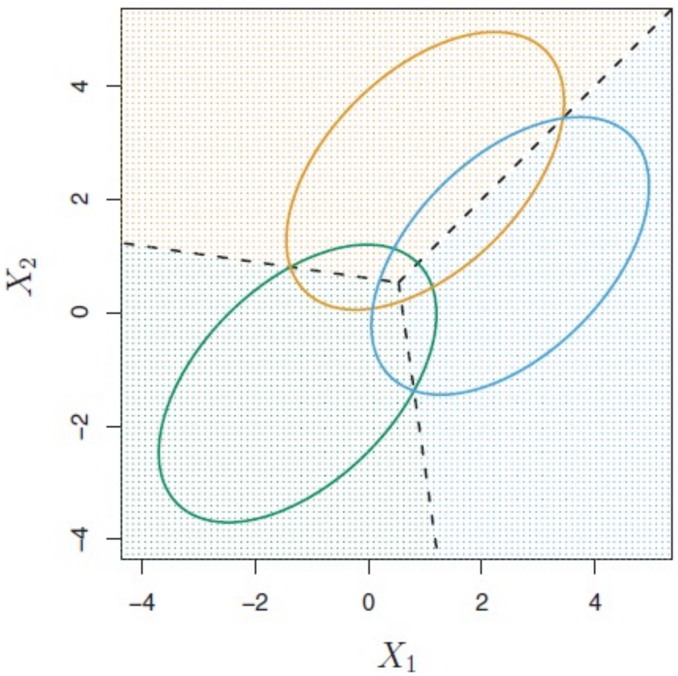
- Taking the log of $P(C = k|X = x)$ and rearranging the terms, it is not hard to show that this is equivalent to assigning the observation to the class for which the value in $\delta_k(x)$ is largest:

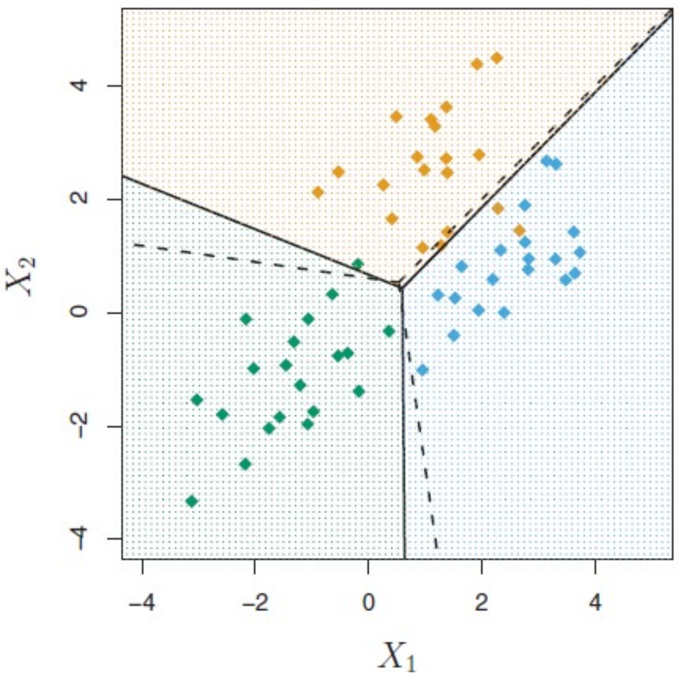
$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

- The LDA decision rule depends on x only through a linear combination of its elements (e.g. the decision boundaries are linear). This is the reason for the word linear in LDA.

LDA for $n > 1$ (Multiple predictors)

- Two predictors ($n = 2$); three classes ($L = 3$)
- 20 observations were generated from each class
- Three ellipses represent regions that contain 95% of prob. for each of three classes.
- Dashed lines: Bayes boundaries; Solid lines: LDA boundaries

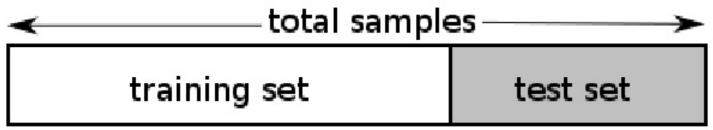




Model Validation

Holdout and Cross-Validation Methods

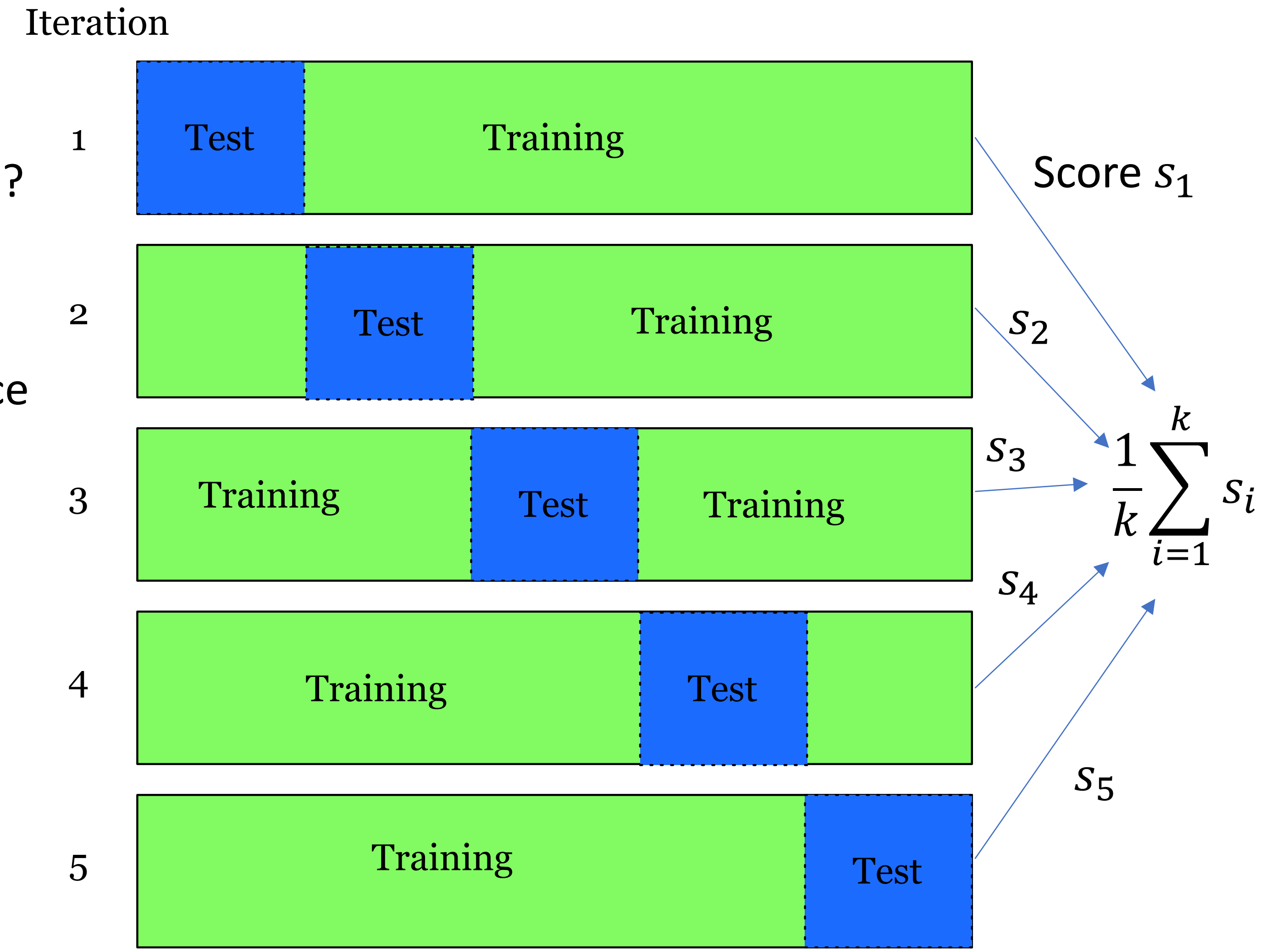
- Holdout method
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Random sampling: a variation of holdout
    - Repeat holdout k times, accuracy = avg. of the accuracies obtained



- Cross-validation (k-fold, where $k = 10$ is most popular)
  - k-fold: randomly partition the data into k mutually exclusive subsets D_1,…, D_k, each approximately equal size. At i-th iteration, use D_i as test set and others as training set
  - Leave-one-out: k folds where k = # of tuples (the test set contains only one tuple)

k-Fold Cross-Validation

- Break up data into k folds
  - Equal positive and negative inside each fold?
- For each fold
  - Choose the fold as a temporary test set
  - Train on $k-1$ folds, compute performance on the test fold
- Report average performance of the k runs



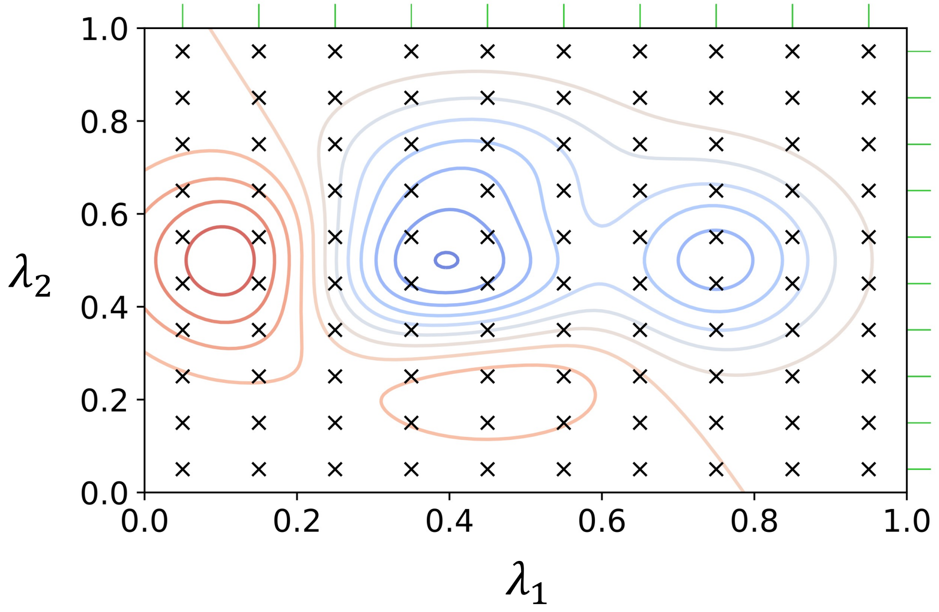
Hyperparameter optimization

- A learning algorithm generally depends on few hyperparameters
  - Examples: estimates of the mean, estimates of the variance, class priors, regularization tradeoff, etc.

- The performance of the algorithm depends on these hyperparameters

- Grid search is the standard approach to tune these hyperparameters

- Exemple :
  - Grid search across different values of two hyperparameters λ_1 and λ_2.
  - For each hyperparameter, 10 different values are considered.
  - Blue contours indicate regions with strong results (model accuracy), whereas red ones show regions with poor results.



Conclusion

Conclusion

- Naïve Bayes is based on the **conditional independence assumption** between attributes
  - Training is very easy and fast; just requiring considering each attribute in each class separately
  - Test is straightforward; just looking up tables or calculating conditional probabilities with estimated distributions

- Linear discriminant analysis is based on a **strong assumption** (multivariate normal distributions) but it works well in practice
  - Training is easy and fast
  - Linear decision rule easy to interpret