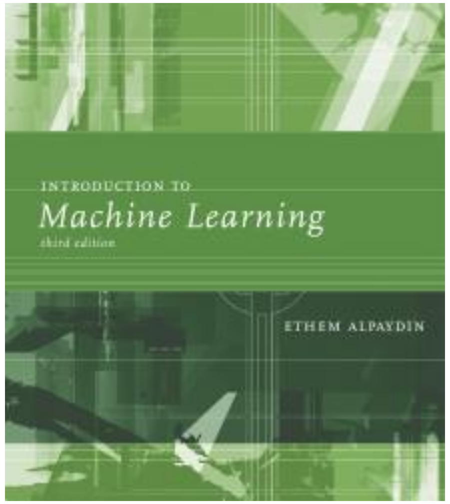
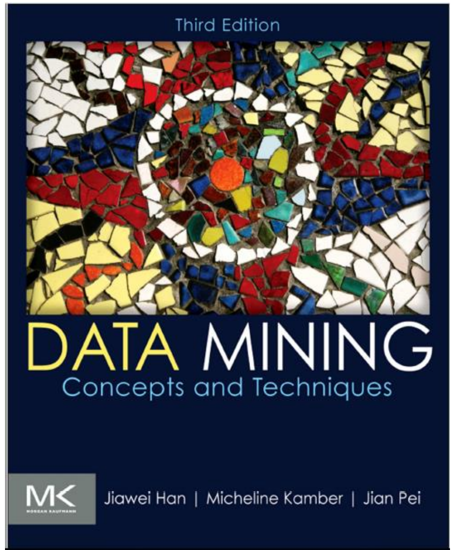


Parametric Classification

Lecture notes by Ethem Alpaydın
Introduction to Machine Learning (Boğaziçi Üniversitesi)



Parametric Classification

$$g_i(x) = p(x \mid C_i)P(C_i)$$

or

$$g_i(x) = \log p(x \mid C_i) + \log P(C_i)$$

$$p(x \mid C_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x-\mu_i)^2}{2\sigma_i^{\,2}}\right]$$

$$g_i(x) = -\frac{1}{2}\log 2\pi - \log \sigma_i - \frac{(x-\mu_i)^2}{2\sigma_i^{\,2}} + \log P(C_i)$$

- Given the sample $\mathcal{X} = \{x^t, r^t\}_{t=1}^N$

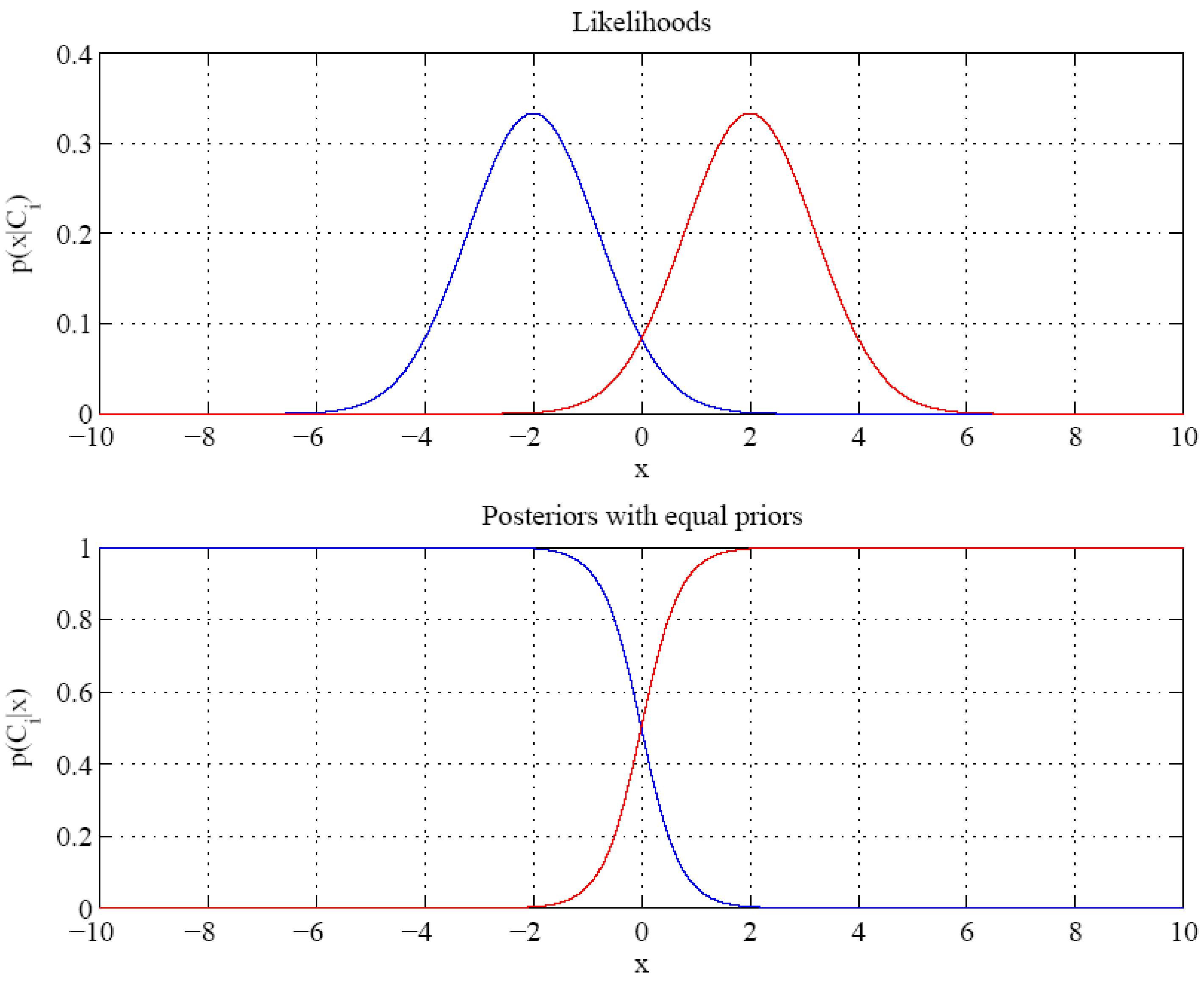
$$x \in \Re \qquad r_i^t = \begin{cases} 1 \text{ if } x^t \in C_i \\ 0 \text{ if } x^t \in C_j, j \neq i \end{cases}$$

- ML estimates are

$$\hat{P}(C_i) = \frac{\sum_t r_i^t}{N} \quad m_i = \frac{\sum_t x^t r_i^t}{\sum_t r_i^t} \quad s_i^2 = \frac{\sum_t \left(x^t - m_i\right)^2 r_i^t}{\sum_t r_i^t}$$

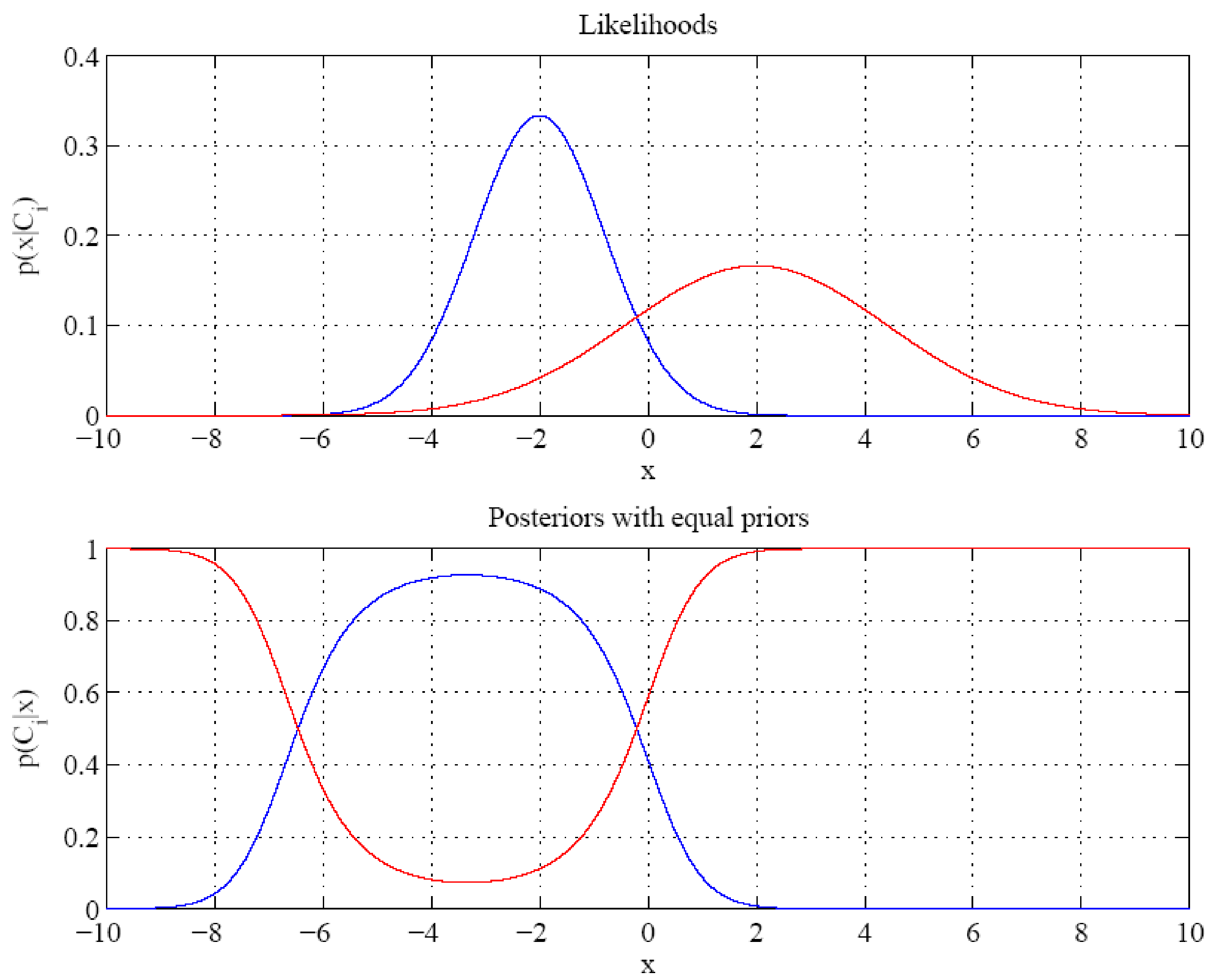
- Discriminant

$$g_i(x) = -\frac{1}{2}\log 2\pi - \log s_i - \frac{\left(x - m_i\right)^2}{2{s_i}^2} + \log \hat{P}(C_i)$$



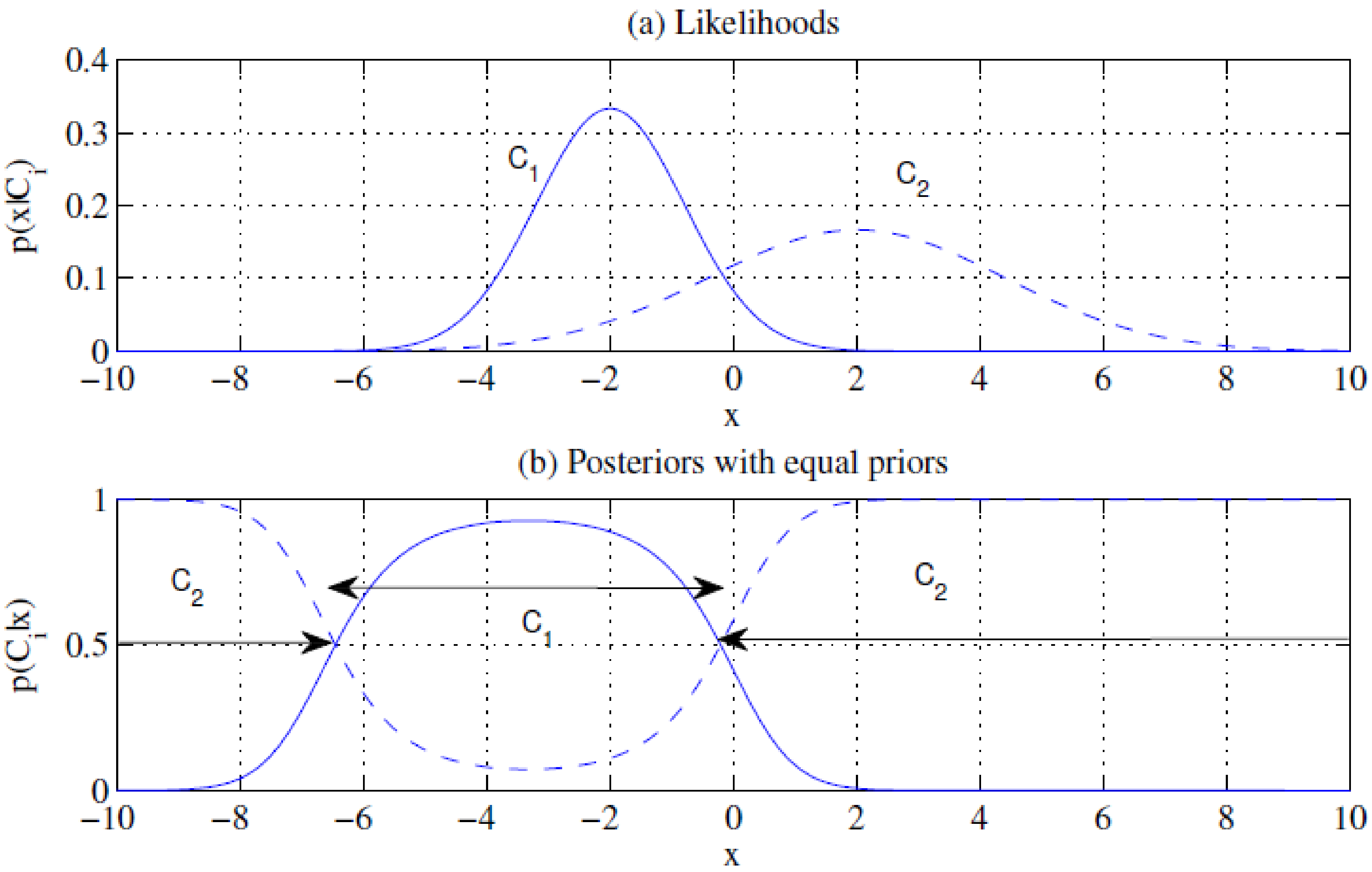
Likelihood functions and posteriors with equal priors for two classes when the input is one-dimensional.

Variances are equal and the posteriors intersect at one point, which is the threshold of decision.



Likelihood functions and posteriors with equal priors for two classes when the input is one-dimensional.

Variances are unequal and the posteriors intersect at two points


(a) Likelihoods
0.4
0.3
0.2
0.1
0
p(x|C_i)
C_1
C_2
–10 –8 –6 –4 –2 0 2 4 6 8 10
x
(b) Posteriors with equal priors
1
0.5
0
p(C_i|x)
C_2
C_1
C_2
–10 –8 –6 –4 –2 0 2 4 6 8 10
x

Multivariate Data

- Multiple measurements (sensors)
- *d* inputs/features/attributes: *d*-variate
- *N* instances/observations/examples

$$\mathbf{X} = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_d^1 \\ X_1^2 & X_2^2 & \cdots & X_d^2 \\ \vdots & & & \\ X_1^N & X_2^N & \cdots & X_d^N \end{bmatrix}$$

Multivariate Parameters

$$\text{Mean}: E[\mathbf{x}] = \boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]^T$$

$$\text{Covariance}: \sigma_{ij} \equiv \text{Cov}(X_i, X_j)$$

$$\text{Correlation}: \text{Corr}(X_i, X_j) \equiv \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

$$\Sigma \equiv \text{Cov}(\mathbf{X}) = E\left[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T\right] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & & & \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

Parameter Estimation

$$\text{Sample mean } \mathbf{m} : m_i = \frac{\sum_{t=1}^{N} x_i^t}{N}, i = 1,\ldots,d$$

$$\text{Covariance matrix } \mathbf{S} : s_{ij} = \frac{\sum_{t=1}^{N} \left(x_i^t - m_i\right)\left(x_j^t - m_j\right)}{N}$$

$$\text{Correlatio n matrix } \mathbf{R} : r_{ij} = \frac{s_{ij}}{s_i s_j}$$

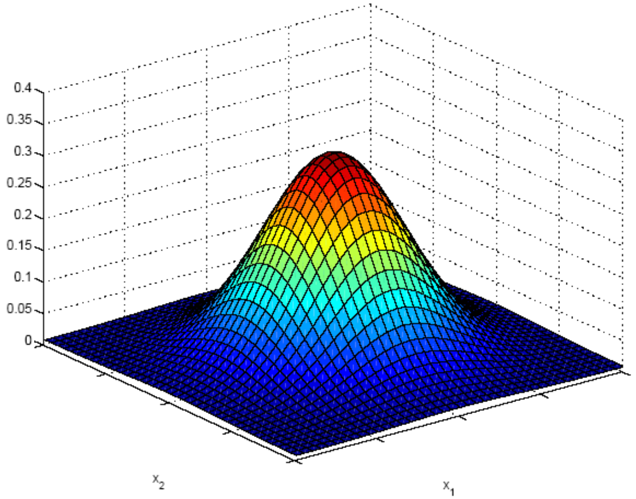
Estimation of Missing Values

- What to do if certain instances have missing attributes?
  - Ignore those instances: not a good idea if the sample is small
  - Use ‘missing’ as an attribute: may give information
  - Imputation: Fill in the missing value
    - Mean imputation: Use the most likely value (e.g., mean)
    - Imputation by regression: Predict based on other attributes

Multivariate Normal Distribution

$$\mathbf{x} \sim \mathcal{N}_d(\boldsymbol{\mu}, \Sigma)$$

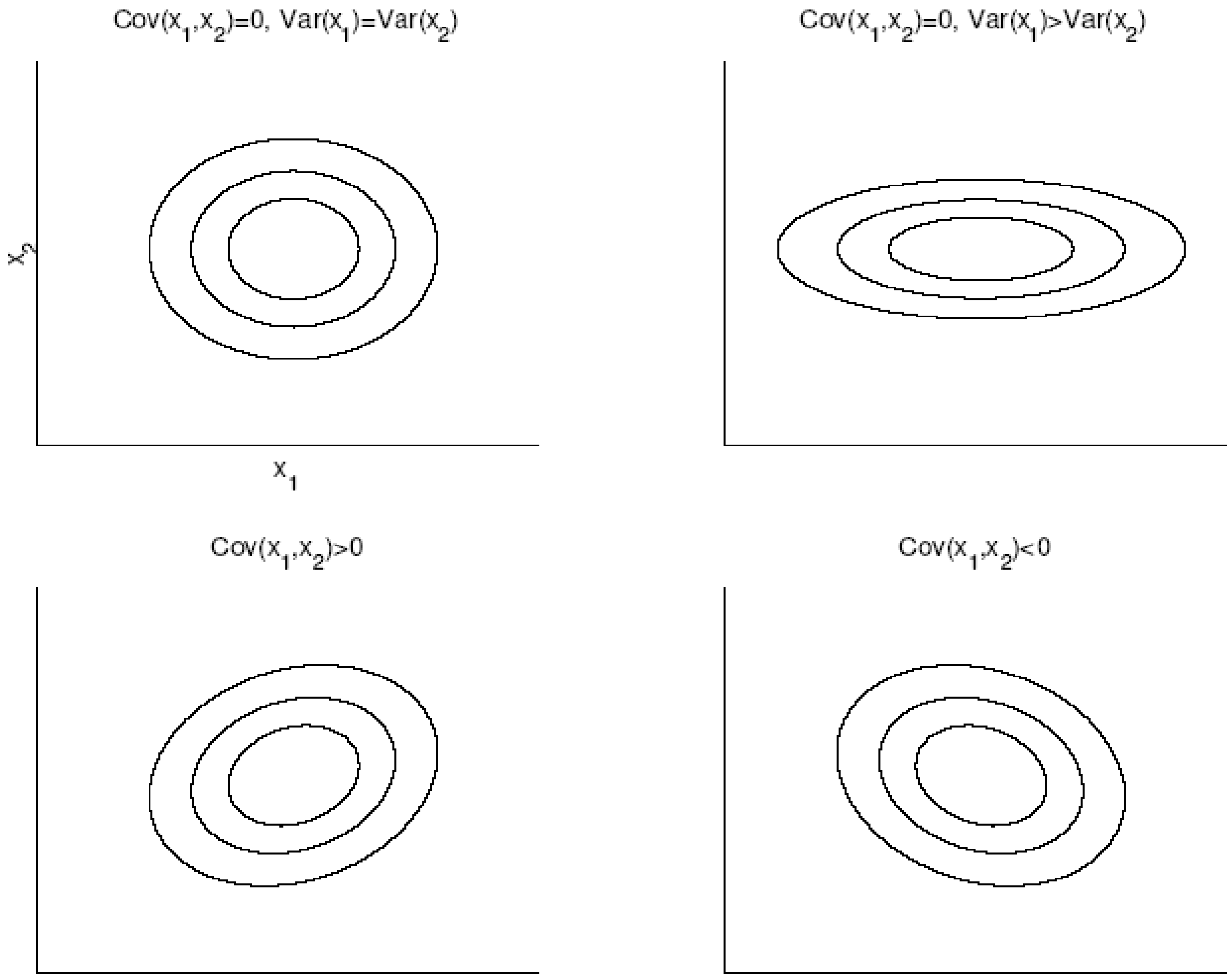
$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$



Mahalanobis distance: $(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})$ measures the distance from $\boldsymbol{x}$ to $\boldsymbol{\mu}$ in terms of $\boldsymbol{\Sigma}$ (normalizes for difference in variances and correlations)

$|\Sigma|$ denotes the determinant of the variance-covariance matrix Σ and Σ^{-1} is just the inverse of the variance-covariance matrix Σ

Bivariate Normal



Isoprobability contour plot of the bivariate normal distribution.

Its center is given by the mean, and its shape and orientation depend on the covariance matrix.

Independent Inputs: Naive Bayes

- If x_i are independent, offdiagonals of **∑** are 0, Mahalanobis distance reduces to weighted (by $1/\sigma_i$) Euclidean distance:

$$p(\mathbf{x}) = \prod_{i=1}^{d} p_i(x_i) = \frac{1}{(2\pi)^{d/2} \prod_{i=1}^{d} \sigma_i} \exp\left[-\frac{1}{2} \sum_{i=1}^{d} \left(\frac{x_i - \mu_i}{\sigma_i} \right)^2 \right]$$

- If variances are also equal, reduces to Euclidean distance

Parametric Classification

- If $p\ (\boldsymbol{x} \mid C_i) \sim N\ (\boldsymbol{\mu}_i, \Sigma_i)$

$$p(\mathbf{x} \mid C_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i)\right]$$

- Discriminant functions

$$\begin{aligned} g_i(\mathbf{x}) &= \log p(\mathbf{x} \mid C_i) + \log P(C_i) \\ &= -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i) + \log P(C_i) \end{aligned}$$

Estimation of Parameters

$$\hat{P}(C_i) = \frac{\sum_t r_i^t}{N}$$

$$\mathbf{m}_i = \frac{\sum_t r_i^t \mathbf{x}^t}{\sum_t r_i^t}$$

$$\mathbf{S}_i = \frac{\sum_t r_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T}{\sum_t r_i^t}$$

$$g_i(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{S}_i| - \frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T \mathbf{S}_i^{-1}(\mathbf{x}-\mathbf{m}_i) + \log \hat{P}(C_i)$$

Different $\mathbf{S}_i$

- Quadratic discriminant

$$g_i(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{S}_i| - \frac{1}{2}\left(\mathbf{x}^T\mathbf{S}_i^{-1}\mathbf{x} - 2\mathbf{x}^T\mathbf{S}_i^{-1}\mathbf{m}_i + \mathbf{m}_i^T\mathbf{S}_i^{-1}\mathbf{m}_i\right) + \log\hat{P}(C_i)$$

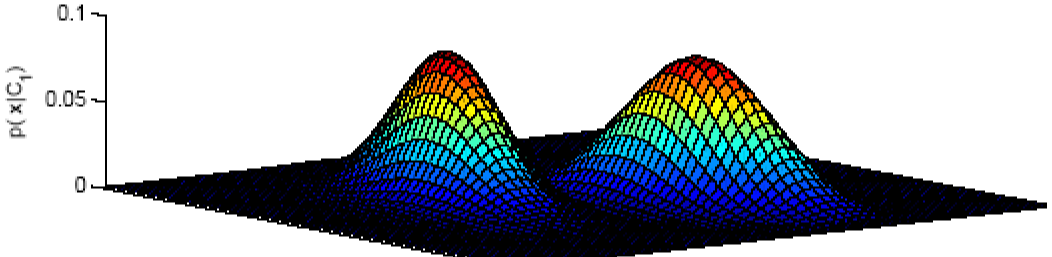
$$= \mathbf{x}^T\mathbf{W}_i\mathbf{x} + \mathbf{w}_i^T\mathbf{x} + w_{i0}$$

where

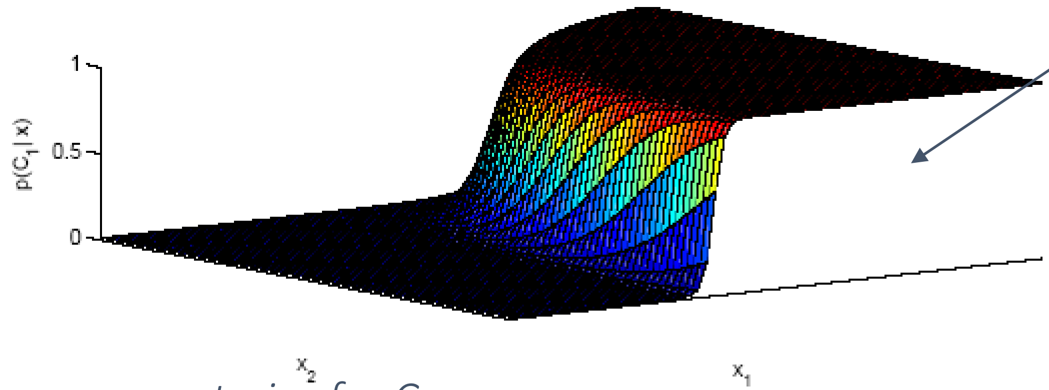
$$\mathbf{W}_i = -\frac{1}{2}\mathbf{S}_i^{-1}$$

$$\mathbf{w}_i = \mathbf{S}_i^{-1}\mathbf{m}_i$$

$$w_{i0} = -\frac{1}{2}\mathbf{m}_i^T\mathbf{S}_i^{-1}\mathbf{m}_i - \frac{1}{2}\log|\mathbf{S}_i| + \log\hat{P}(C_i)$$

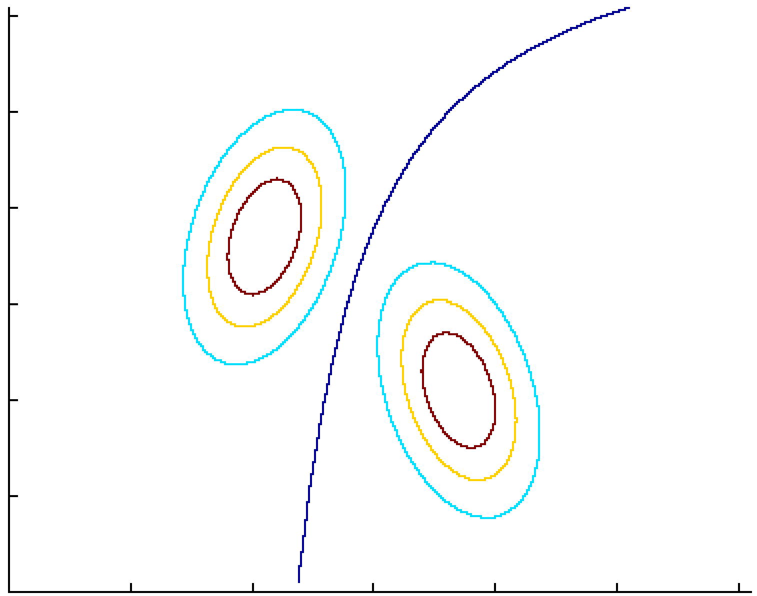


likelihoods



posterior for C_1

discriminant:
$P(C_1|\boldsymbol{x}) = 0.5$



Classes have different covariance matrices.

Likelihood densities and the posterior probability for one of the classes

Class distributions are indicated by isoprobability contours and the discriminant is drawn.

Common Covariance Matrix **S**

- Shared common sample covariance **S**

$$\mathbf{S} = \sum_i \hat{P}(C_i)\mathbf{S}_i$$

- Discriminant reduces to

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_i)^T \mathbf{S}^{-1}(\boldsymbol{x} - \boldsymbol{m}_i) + \log \hat{P}(C_i)$$

which is a linear discriminant

$$g_i(\mathbf{x}) = \mathbf{w}_i{}^T \mathbf{x} + w_{i0}$$

where

$$\mathbf{w}_i = \mathbf{S}^{-1}\mathbf{m}_i \quad w_{i0} = -\frac{1}{2}\mathbf{m}_i{}^T \mathbf{S}^{-1}\mathbf{m}_i + \log \hat{P}(C_i)$$

Diagonal S

- When x_j $j = 1,..d$, are independent, ∑ is diagonal

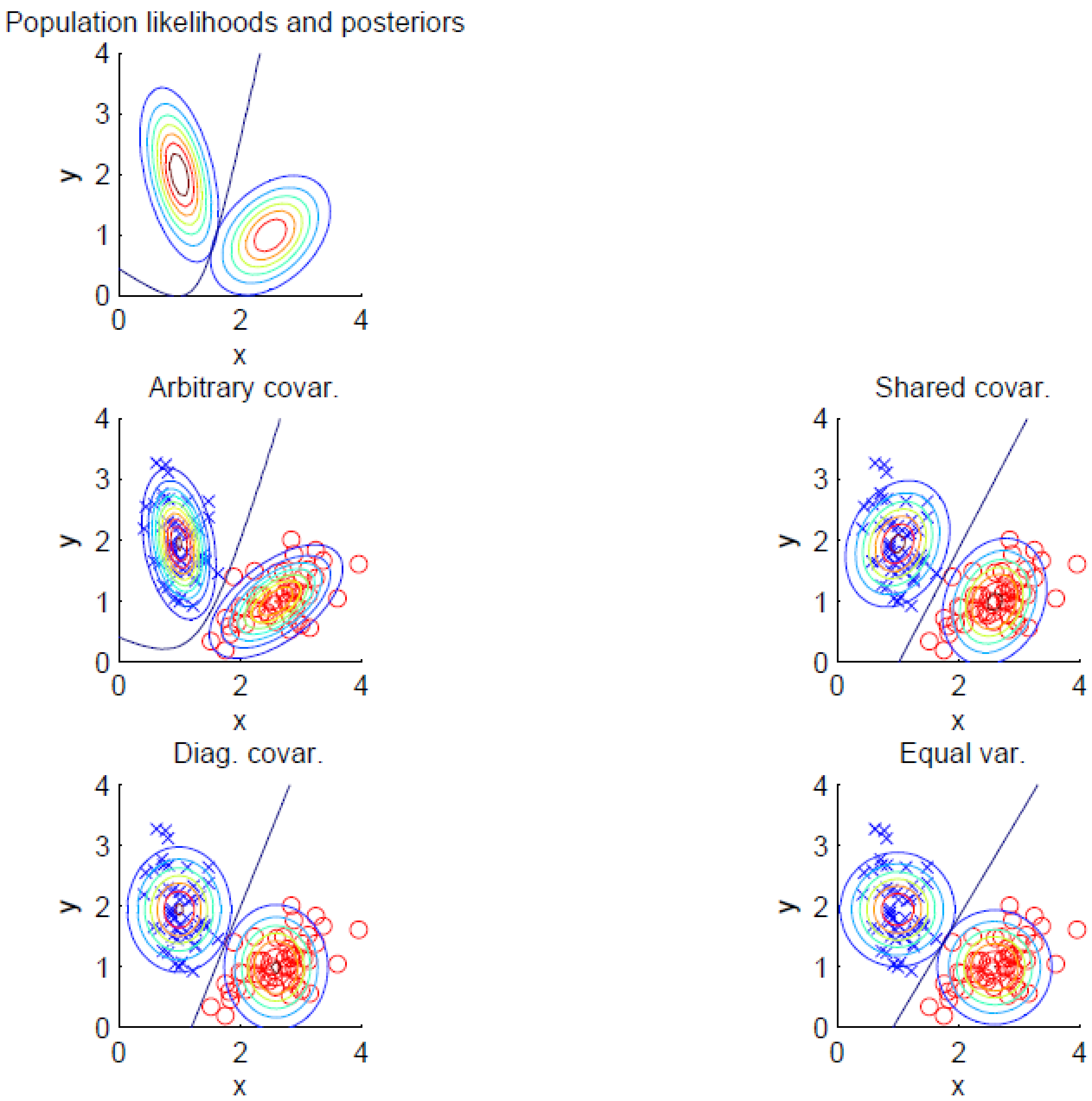
  $p\,(\boldsymbol{x}|C_i) = \prod_j p\,(x_j\,|C_i)$ (Naive Bayes' assumption)

$$g_i(\mathbf{x}) = -\frac{1}{2}\sum_{j=1}^{d}\left(\frac{x_j^t - m_{ij}}{s_j}\right)^2 + \log \hat{P}(C_i)$$

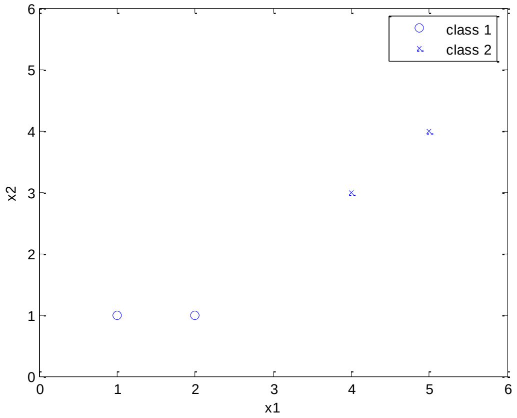
Classify based on weighted Euclidean distance (in s_j units) to the nearest mean

Model Selection

Different cases of the covariance matrices fitted to the same data lead to different boundaries.



Example



- Consider the labeled data points:

$$X = \left\{ \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 4 \\ 3 \end{bmatrix}, 2 \right), \left(\begin{bmatrix} 5 \\ 4 \end{bmatrix}, 2 \right) \right\}$$

- Assuming that inputs are normally distributed with class covariance matrices as follows:

$$S_1 = S_2 = s^2 I = s^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Compute the discriminant functions for both classes, $g_1(x)$ and $g_2(x)$

Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$$

- This greatly reduces the computation cost: Only counts the class distribution

Naïve Bayes Classifier

- If A_k is categorical, $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i, D}|$ (# of tuples of C_i in D)

- If A_k is continous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

Naïve Bayes Classifier: Training Dataset

Table 8.1 Class-Labeled Training Tuples from the *AllElectronics* Customer Database

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Naïve Bayes Classifier - Example

- **Predicting a class label using naive Bayesian classification.** We wish to predict the class label of a tuple using naive Bayesian classification, given the training data in Table 8.1.
- The data tuples are described by the attributes *age*, *income*, *student*, and *credit rating*.
- The class label attribute, *buys computer*, has two distinct values (namely, {*yes, no*}). Let $C1$ correspond to the class *buys computer* D *yes* and $C2$ correspond to *buys computer* D *no.*
- The tuple we wish to classify is

X = (*age = youth, income = medium, student = yes, credit rating = fair*)

Naïve Bayes Classifier - Example

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643

  P(buys_computer = "no") = 5/14= 0.357

- Compute $P(X|C_i)$ for each class

  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222

  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6

  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444

  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4

  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667

  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2

  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667

  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044

P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028

P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be **non-zero**. Otherwise, the predicted prob. will be zero
- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)
- Use **Laplacian correction** (or Laplacian estimator)
  - *Adding 1 to each case*

    Prob(income = low) = 1/1003

    Prob(income = medium) = 991/1003

    Prob(income = high) = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Avoiding the Zero-Probability Problem

Laplace

$c: number\ of\ classesi$

$N_c: number\ of\ instances\ in\ the\ class,$

$N_{ic}: number\ of\ instances\ having\ attribute\ value\ A_i\ in\ class\ c$

$$\mathrm{P}(A_i|C) = \frac{N_{ic} + 1}{N_c + c}$$

Example

We wish to predict the class label of a tuple using naive Bayesian classification given the following training data. The data tuples are described by the attributes genre and price. The class label attribute, class, has two distinct values, (namely, recommended and not recommended). Given a new instance, predict its label.

X = (genre = self-help, price = medium)

Book	Genre	Price	Class
B1	Romance	Low	Recommended
B2	Romance	Medium	Recommended
B3	Thriller	Low	Recommended
B4	Thriller	Low	Recommended
B5	Self-Help	High	Not Recommended
B6	Romance	High	Not Recommended
B7	Self-Help	High	Not Recommended

Example

$P(recommended) = \frac{4}{7}, \quad P(not\ recommended) = \frac{3}{7}$

$$P(recommended|X) \\ = P(self - help|recommended) * P(medium\ |recommended) \\ * P(recommended) = \frac{1}{7} * \frac{2}{7} * \frac{4}{7} = 0.023$$

$$P(not\ recommended|X) \\ = P(self - help|not\ recommended) * P(medium\ |\ not\ recommended) \\ * P(not\ recommended) = \frac{3}{6} * \frac{1}{6} * \frac{3}{7} = 0.036$$

Class is $not\ recommended$

Naïve Bayes Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.

      Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier