Probabilistic classifiers

CE-477: Machine Learning - CS-828: Theory of Machine Learning
Sharif University of Technology
Fall 2024

Fatemeh Seyyedsalehi

References of the lecture are mentioned in the last slide

Topics

- Probabilistic approach
  - Bayes decision theory
  - Generative models
    - Gaussian Bayes classifier
    - Naïve Bayes
  - Discriminative models
    - Logistic regression

Classification problem: probabilistic view

- Each feature as a random variable

- Class label also as a random variable

- We observe the feature values for a random sample and we intend to find its class label
  - Evidence: feature vector $\boldsymbol{x}$
  - Query: class label

Definitions

- Posterior probability: $p(\mathcal{C}_k|\boldsymbol{x})$

- Likelihood or class conditional probability: $p(\boldsymbol{x}|\mathcal{C}_k)$

- Prior probability: $p(\mathcal{C}_k)$

$p(\boldsymbol{x})$: pdf of feature vector $\boldsymbol{x}$ ($p(\boldsymbol{x}) = \sum_{k=1}^{K} p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)$)

$p(\boldsymbol{x}|\mathcal{C}_k)$: pdf of feature vector $\boldsymbol{x}$ for samples of class $\mathcal{C}_k$

$p(\mathcal{C}_k)$: probability of the label be $\mathcal{C}_k$

Bayes decision rule

$K = 2$

If $P(\mathcal{C}_1|\boldsymbol{x}) > P(\mathcal{C}_2|\boldsymbol{x})$ decide $\mathcal{C}_1$
otherwise decide $\mathcal{C}_2$

$$p(error|\boldsymbol{x}) = \begin{cases} p(C_2|\boldsymbol{x}) & \text{if we decide } \mathcal{C}_1 \\ P(C_1|\boldsymbol{x}) & \text{if we decide } \mathcal{C}_2 \end{cases}$$

- If we use Bayes decision rule:

$$P(error|\boldsymbol{x}) = \min\{P(\mathcal{C}_1|\boldsymbol{x}), P(\mathcal{C}_2|\boldsymbol{x})\}$$

Using Bayes rule, for each $\boldsymbol{x}$, $P(error|\boldsymbol{x})$ is as small as possible and thus this rule minimizes the probability of error

Optimal classifier

- The optimal decision is the one that minimizes the expected number of mistakes

- We show that Bayes classifier is an optimal classifier

Bayes decision rule
Minimizing misclassification rate

- Decision regions: $\mathcal{R}_k = \{\boldsymbol{x} | \alpha(\boldsymbol{x}) = k\}$
  - All points in $\mathcal{R}_k$ are assigned to class $\mathcal{C}_k$

$K = 2$

$$p(error) = E_{\boldsymbol{x},y}[I(\alpha(\boldsymbol{x}) \neq y)]$$

$$= p(\boldsymbol{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\boldsymbol{x} \in \mathcal{R}_2, \mathcal{C}_1)$$

$$= \int_{\mathcal{R}_1} p(\boldsymbol{x}, \mathcal{C}_2)\, d\boldsymbol{x} + \int_{\mathcal{R}_2} p(\boldsymbol{x}, \mathcal{C}_1)\, d\boldsymbol{x}$$

$$= \int_{\mathcal{R}_1} p(\mathcal{C}_2 | \boldsymbol{x}) p(\boldsymbol{x})\, d\boldsymbol{x} + \int_{\mathcal{R}_2} p(\mathcal{C}_1 | \boldsymbol{x}) p(\boldsymbol{x})\, d\boldsymbol{x}$$

Choose class with highest $p(\mathcal{C}_k | \boldsymbol{x})$ as $\alpha(\boldsymbol{x})$

Bayes minimum error

- Bayes minimum error classifier:

$$\min_{\alpha(.)} E_{\boldsymbol{x},y}[I(\alpha(\boldsymbol{x}) \neq y)]$$

Zero-one loss

  - If we know the probabilities in advance then the above optimization problem will be solved easily.
    - $\alpha(\boldsymbol{x}) = \underset{y}{\operatorname{argmax}}\, p(y|\boldsymbol{x})$

- In practice, we can estimate $p(y|\boldsymbol{x})$ based on a set of training samples $\mathcal{D}$

Bayes theorem

- Bayes' theorem

Posterior, Likelihood, Prior

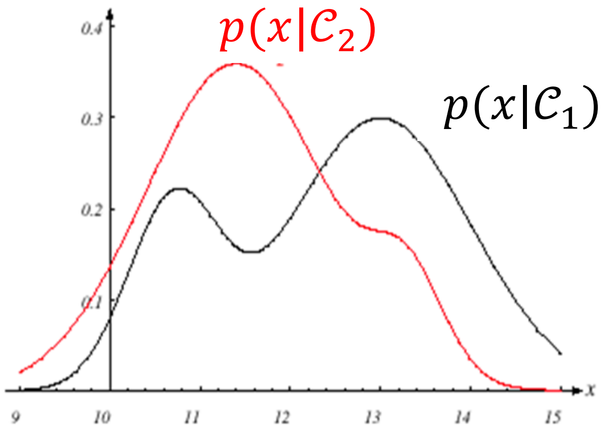
$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\boldsymbol{x})}$$

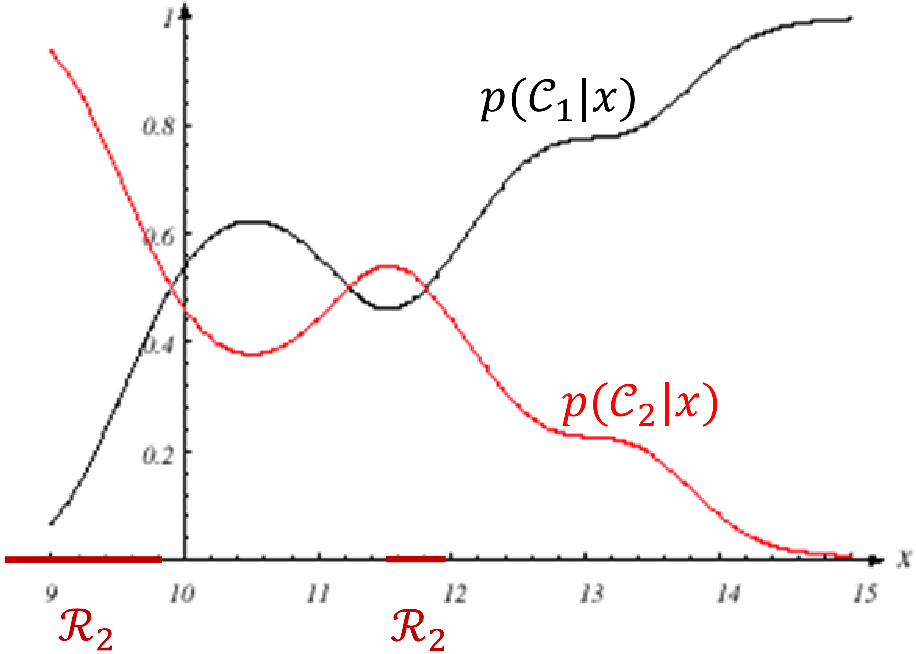
- Posterior probability: $p(\mathcal{C}_k|\boldsymbol{x})$
- Likelihood or class conditional probability: $p(\boldsymbol{x}|\mathcal{C}_k)$
- Prior probability: $p(\mathcal{C}_k)$

$p(\boldsymbol{x})$: pdf of feature vector $\boldsymbol{x}$ ($p(\boldsymbol{x}) = \sum_{k=1}^{K} p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)$)
$p(\boldsymbol{x}|\mathcal{C}_k)$: pdf of feature vector $\boldsymbol{x}$ for samples of class $\mathcal{C}_k$
$p(\mathcal{C}_k)$: probability of the label be $\mathcal{C}_k$

Bayes decision rule: example

- Bayes decision: Choose the class with highest $p(\mathcal{C}_k|\boldsymbol{x})$





$$p(\mathcal{C}_1) = \frac{2}{3}$$

$$p(\mathcal{C}_2) = \frac{1}{3}$$

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\boldsymbol{x})}$$

$$p(\boldsymbol{x}) = p(\mathcal{C}_1)p(\boldsymbol{x}|\mathcal{C}_1) + p(\mathcal{C}_2)p(\boldsymbol{x}|\mathcal{C}_2)$$

Bayesian decision rule

- If $P(\mathcal{C}_1|\boldsymbol{x}) > P(\mathcal{C}_2|\boldsymbol{x})$ decide $\mathcal{C}_1$
  otherwise decide $\mathcal{C}_2$

Equivalent

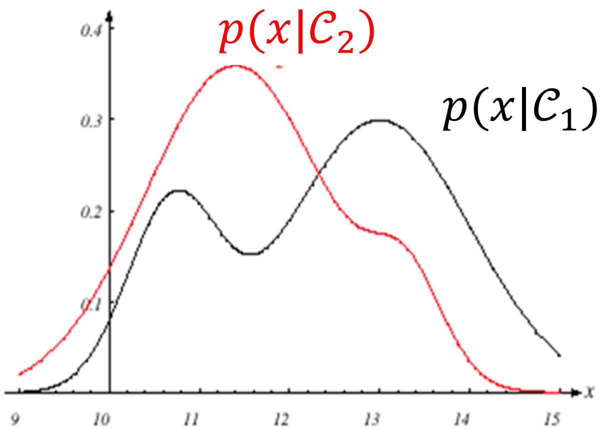
- If $\frac{p(\boldsymbol{x}|\mathcal{C}_1)P(\mathcal{C}_1)}{p(\boldsymbol{x})} > \frac{p(\boldsymbol{x}|\mathcal{C}_2)P(\mathcal{C}_2)}{p(\boldsymbol{x})}$ decide $\mathcal{C}_1$
  otherwise decide $\mathcal{C}_2$

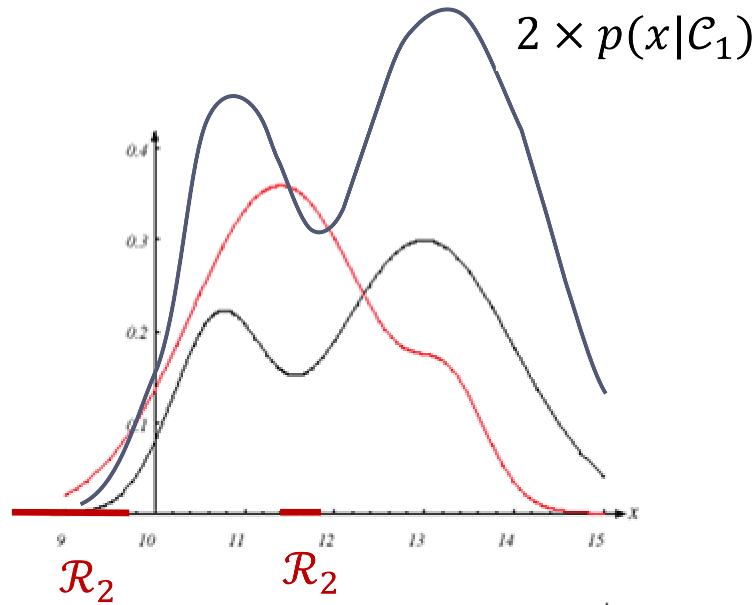
Equivalent

- If $p(\boldsymbol{x}|\mathcal{C}_1)P(\mathcal{C}_1) > p(\boldsymbol{x}|\mathcal{C}_2)P(\mathcal{C}_2)$ decide $\mathcal{C}_1$
  otherwise decide $\mathcal{C}_2$

Bayes decision rule: example

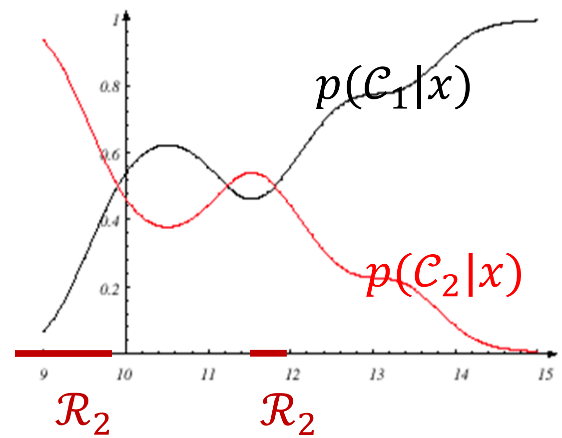
- Bayes decision: Choose the class with highest $p(\mathcal{C}_k|\boldsymbol{x})$





$$p(\mathcal{C}_1) = \frac{2}{3}$$

$$p(\mathcal{C}_2) = \frac{1}{3}$$



Bayes Classier

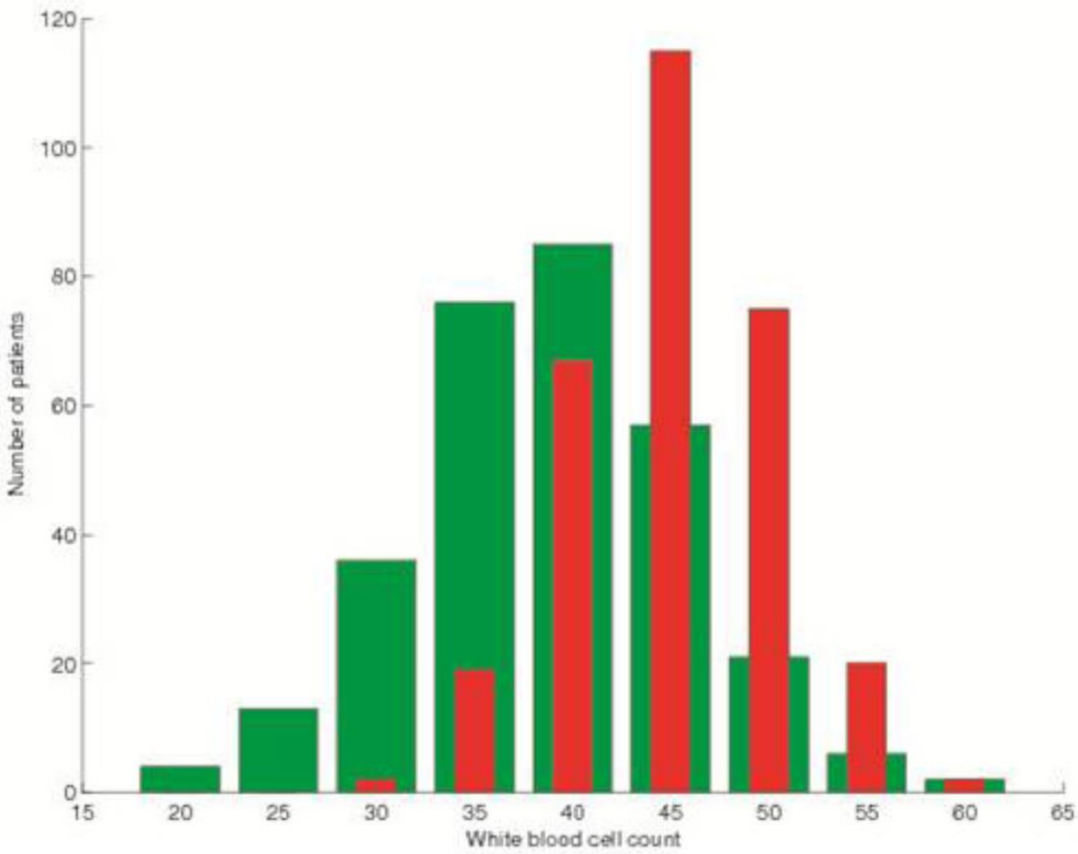
- Simple Bayes classifier: estimate posterior probability of each class

- What should the decision criterion be?
  - Choose class with highest $p(\mathcal{C}_k|\boldsymbol{x})$

- The optimal decision is the one that minimizes the expected number of mistakes

Diabetes example

- white blood cell count



This example has been adopted from Sanja Fidler's slides, University of Toronto, CSC411

Diabetes example

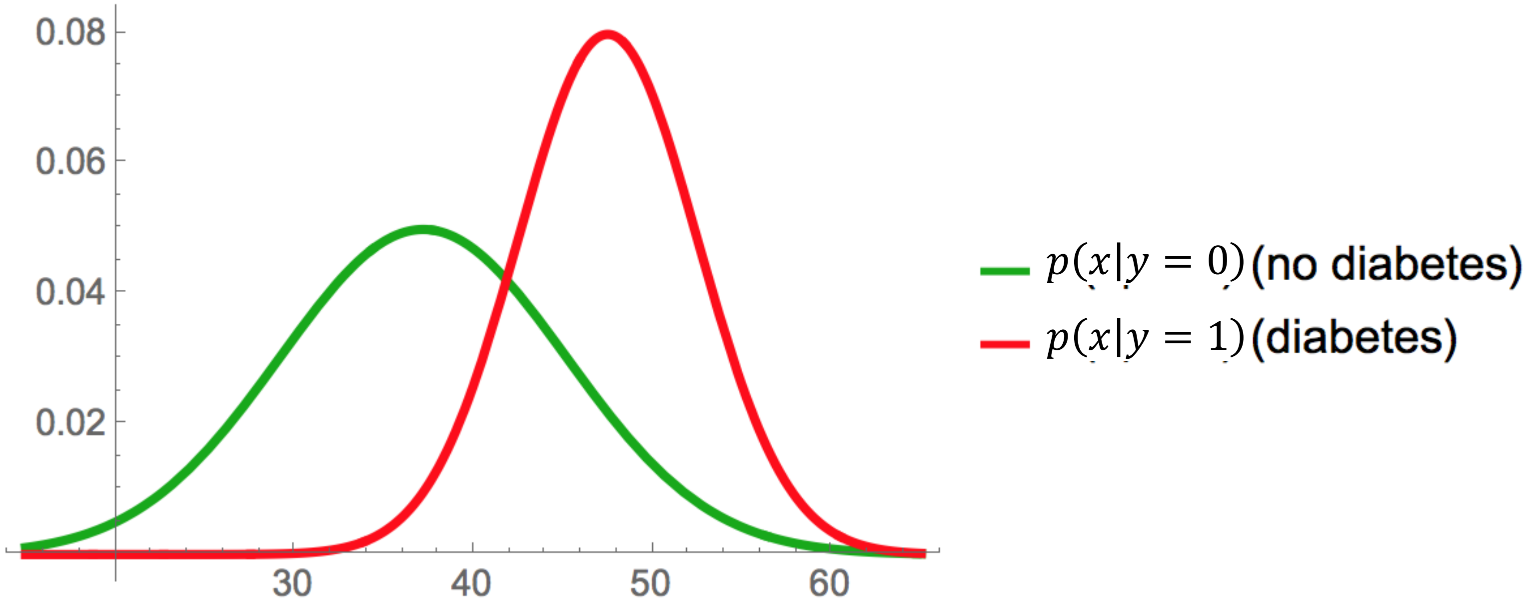
- Doctor has a prior $p(y = 1) = 0.2$
  - Prior: In the absence of any observation, what do I know about the probability of the classes?

- A patient comes in with white blood cell count x

- Does the patient have diabetes $p(y = 1|x)$?
  - given a new observation, we still need to compute the posterior

Diabetes example

$$p(x = 40|y = 0)P(y = 0) >^? p(x = 40|y = 1)P(y = 1)$$



This example has been adopted from Sanja Fidler's slides, University of Toronto, CSC411

Estimate probability densities from data

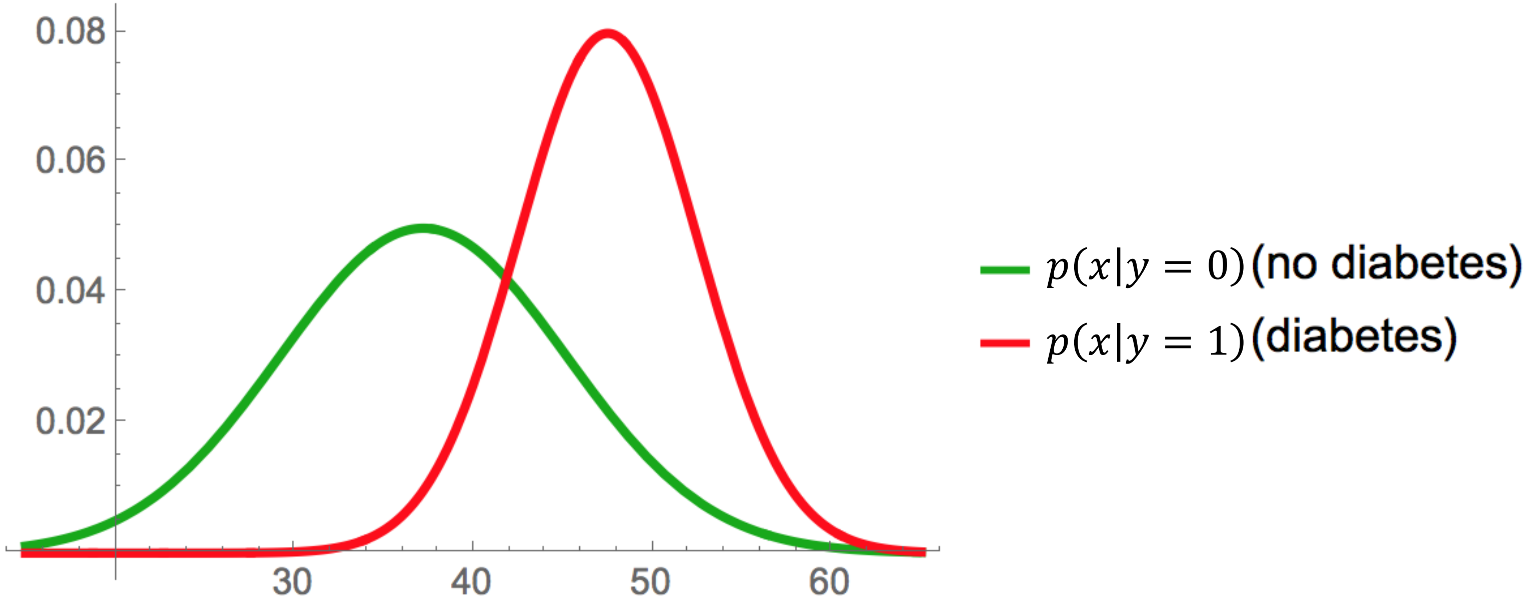
- If we assume Gaussian distributions for $p(x|y=0)$ and $p(x|y=1)$

- Recall that for samples $\{x^{(1)}, \dots, x^{(N)}\}$, if we assume a Gaussian distribution, the MLE estimates will be

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

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

Diabetes example



$$p(x|y=1) = N(\mu_1, \sigma_1^2)$$

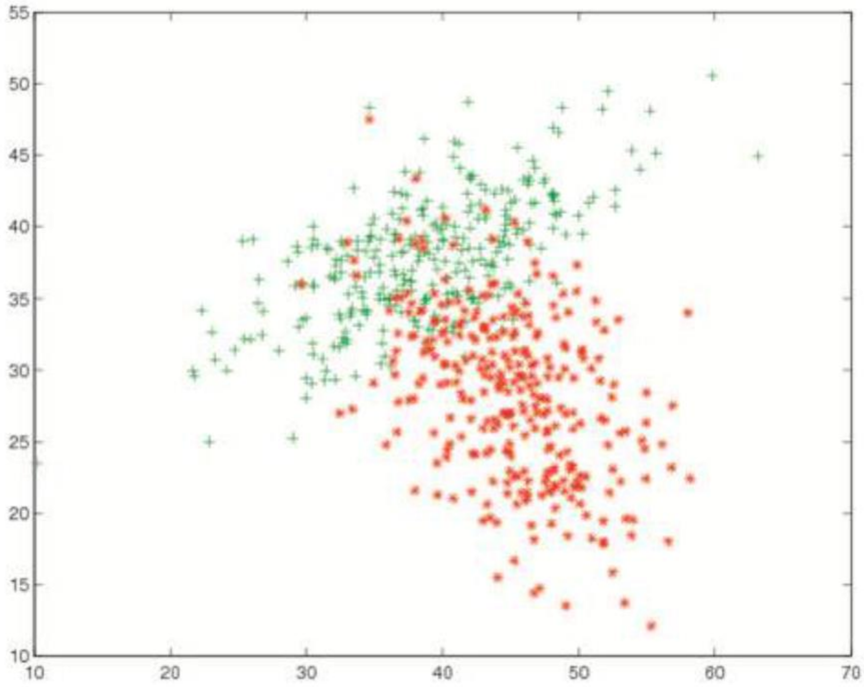
$$\mu_1 = \frac{\sum_{n:\, y^{(n)}=1} x^{(n)}}{\sum_{n:\, y^{(n)}=1} 1} = \frac{\sum_{n:\, y^{(n)}=1} x^{(n)}}{N_1}$$

$$\sigma_1^2 = \frac{\sum_{n:\, y^{(n)}=1} \left(x^{(n)} - \mu_1\right)^2}{N_1}$$

This example has been adopted from Sanja Fidler's slides, University of Toronto, CSC411

Diabetes example

- Add a second observation: Plasma glucose value



This example has been adopted from Sanja Fidler's slides, University of Toronto, CSC411

Generative approach for this example

- Multivariate Gaussian distributions for $p(x|\mathcal{C}_k)$:

$$p(\boldsymbol{x}|y=k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\}$$

$$k = 1,2$$

- Prior distribution $p(y)$:
  - $p(y=1) = \pi, \quad p(y=0) = 1-\pi$

MLE for multivariate Gaussian

- For samples $\{x^{(1)}, \dots, x^{(N)}\}$, if we assume a multivariate Gaussian distribution, the MLE estimates will be:

$$\boldsymbol{\mu} = \frac{\sum_{n=1}^{N} \boldsymbol{x}^{(n)}}{N}$$

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} \left(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}\right)\left(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}\right)^{T}$$

Generative approach: example

$y \in \{0,1\}$

Maximum likelihood estimation ($D = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}$):

- $\pi = \frac{N_1}{N}$
- $\boldsymbol{\mu}_1 = \frac{\sum_{n=1}^{N} y^{(n)} \boldsymbol{x}^{(n)}}{N_1}, \boldsymbol{\mu}_2 = \frac{\sum_{n=1}^{N} (1-y^{(n)}) \boldsymbol{x}^{(n)}}{N_2}$
- $\boldsymbol{\Sigma}_1 = \frac{1}{N_1} \sum_{n=1}^{N} y^{(n)} \left(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}\right)\left(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}\right)^T$
- $\boldsymbol{\Sigma}_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - y^{(n)}) \left(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}\right)\left(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}\right)^T$

$$N_1 = \sum_{n=1}^{N} y^{(n)}$$

$$N_2 = N - N_1$$

Decision boundary for Gaussian Bayes classifier

$$p(\mathcal{C}_1|\boldsymbol{x}) = p(\mathcal{C}_2|\boldsymbol{x})$$

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\boldsymbol{x})}$$

$$\ln p(\mathcal{C}_1|\boldsymbol{x}) = \ln p(\mathcal{C}_2|\boldsymbol{x})$$

$$\ln p(\boldsymbol{x}|\mathcal{C}_1) + \ln p(\mathcal{C}_1) - \ln p(\boldsymbol{x})$$
$$= \ln p(\boldsymbol{x}|\mathcal{C}_2) + \ln p(\mathcal{C}_2) - \ln p(\boldsymbol{x})$$

Decision boundary for Gaussian Bayes classifier

$$p(\mathcal{C}_1|\boldsymbol{x}) = p(\mathcal{C}_2|\boldsymbol{x})$$

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\boldsymbol{x})}$$

$$\ln p(\mathcal{C}_1|\boldsymbol{x}) = \ln p(\mathcal{C}_2|\boldsymbol{x})$$

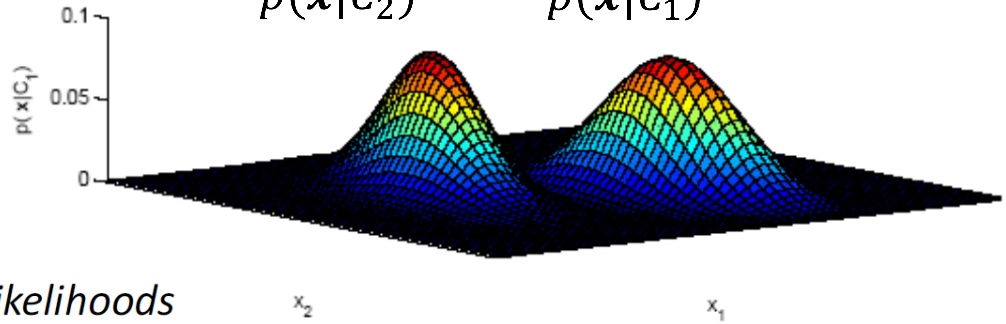
$$\ln p(\boldsymbol{x}|\mathcal{C}_1) + \ln p(\mathcal{C}_1) - \ln p(\boldsymbol{x}) \\ = \ln p(\boldsymbol{x}|\mathcal{C}_2) + \ln p(\mathcal{C}_2) - \ln p(\boldsymbol{x})$$

$$\ln p(\boldsymbol{x}|\mathcal{C}_1) + \ln p(\mathcal{C}_1) = \ln p(\boldsymbol{x}|\mathcal{C}_2) + \ln p(\mathcal{C}_2)$$

$$\ln p(\boldsymbol{x}|\mathcal{C}_k) \\ = -\frac{d}{2}\ln 2\pi - \frac{1}{2}\ln\left|\boldsymbol{\Sigma}_k\right| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)$$

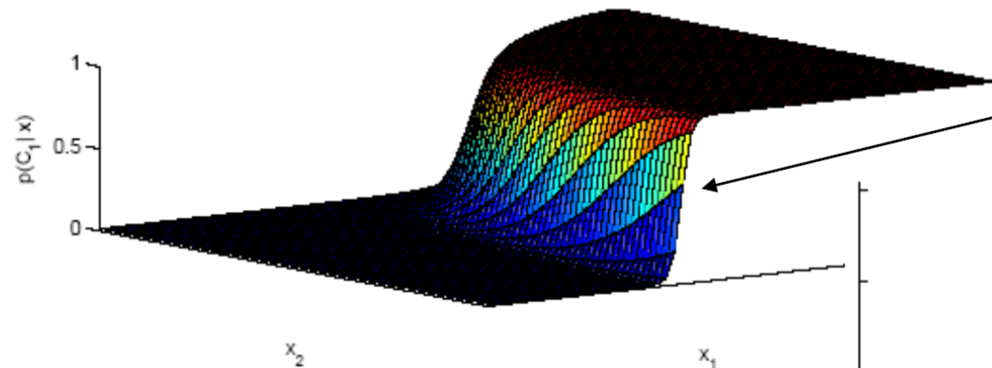
Decision boundary

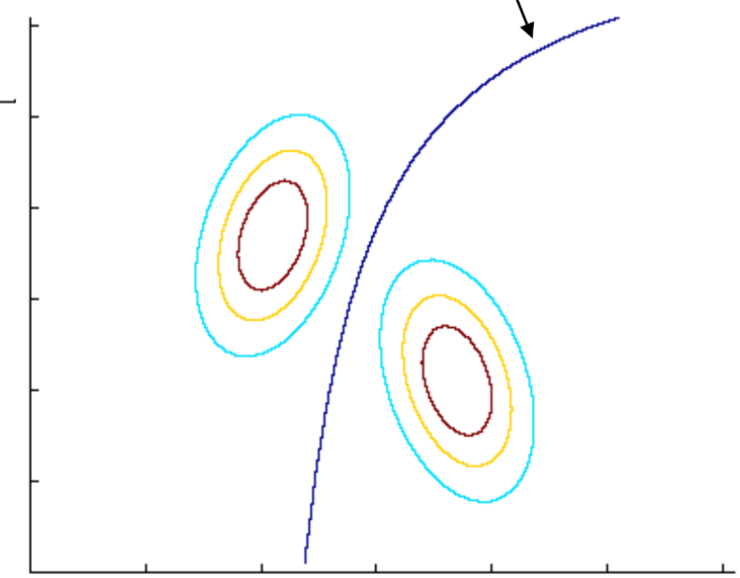
$p(\boldsymbol{x}|C_2)$ $p(\boldsymbol{x}|C_1)$



likelihoods

discriminant:
$p(C_1|\boldsymbol{x})=p(C_2|\boldsymbol{x})$





$p(C_1|\boldsymbol{x})$

[4]

Gaussian discriminant analysis

- When the likelihood probability is distributed according to a multivariate normal distribution.
- The general model is:

$$C_k \sim Bernouli(\phi) = \phi^y (1-\phi)^{(1-y)}$$

$$p(\boldsymbol{x}|y=k) \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\ )$$
$$= \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\}$$

Gaussian discriminant analysis

- The log-likelihood of the data:
  - $k = 2$

$$\ell(\phi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\ , \boldsymbol{\Sigma}_2\ ) = \log \prod_{i=1}^{N} p(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}; \phi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\ , \boldsymbol{\Sigma}_2\ )$$

- By maximizing with respect to the parameters

$$\phi = \frac{N_1}{N}$$

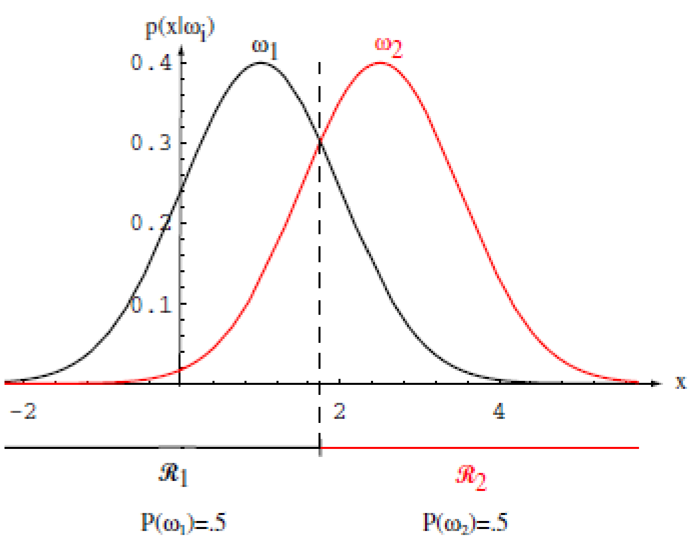
$$\boldsymbol{\mu}_1 = \frac{\sum_{i=1}^{N_1} y^{(i)} \boldsymbol{x}^{(i)}}{N_1}, \boldsymbol{\mu}_2 = \frac{\sum_{i=1}^{N_2} (1 - y^{(i)}) \boldsymbol{x}^{(i)}}{N_2}$$

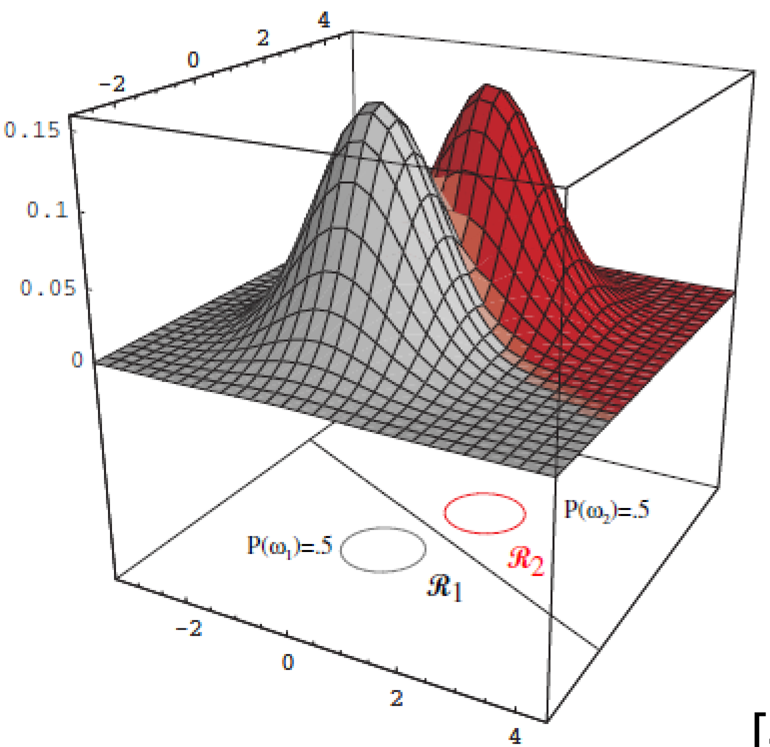
$$\boldsymbol{\Sigma}_1 = \frac{1}{N_1} \sum_{i=1}^{N_1} y^{(i)} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}\right) \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}\right)^T$$

$$\boldsymbol{\Sigma}_2 = \frac{1}{N_2} \sum_{i=1}^{N_2} (1 - y^{(i)}) \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}\right) \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}\right)^T$$

Gaussian discriminant analysis

- A special case: $\boldsymbol{\Sigma}_k = \sigma^2 I$
  - The decision boundary is a Hyperplane which is orthogonal to the line between the means.

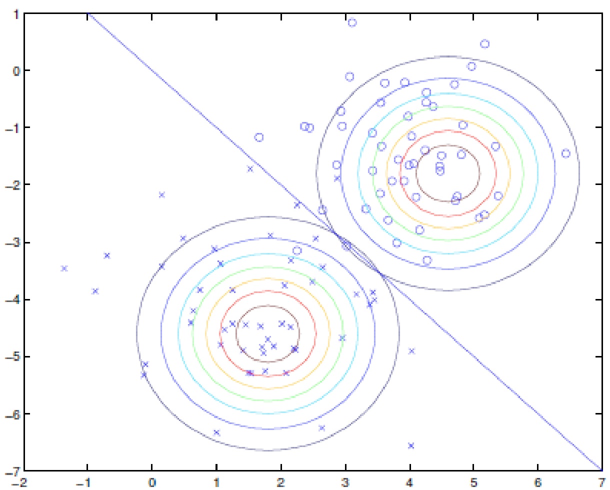


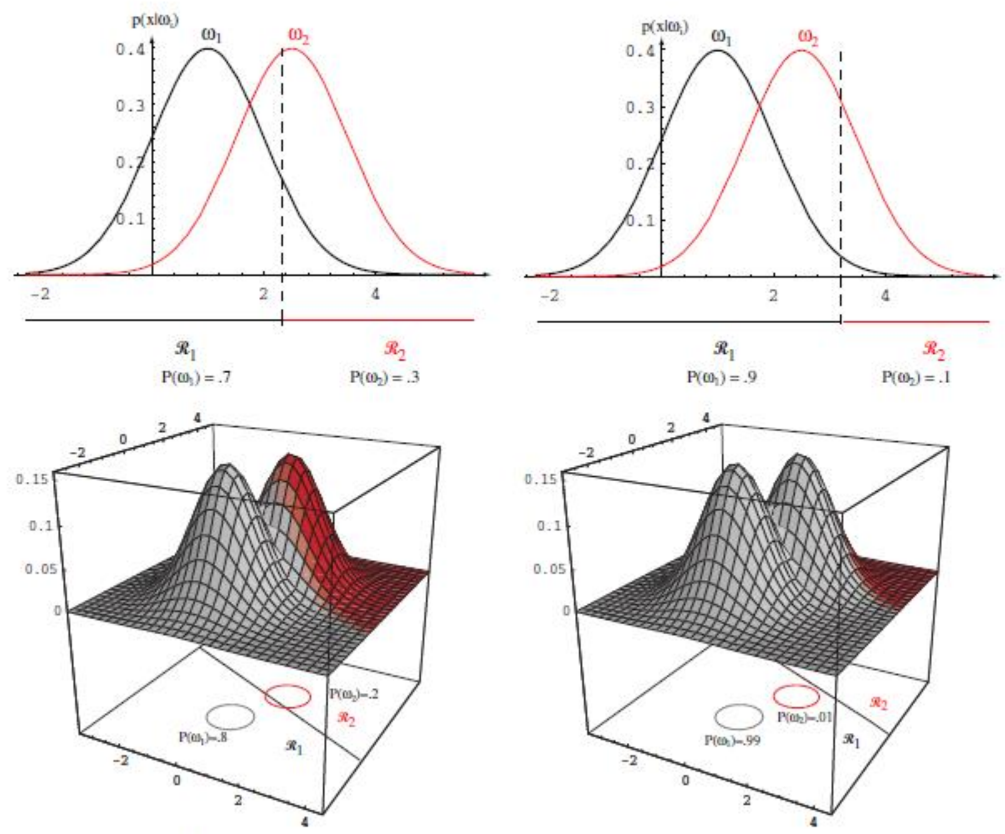


[3]

Gaussian discriminant analysis

- A special case: $\boldsymbol{\Sigma}_k = \sigma^2 I$
  - Features are independent random variables

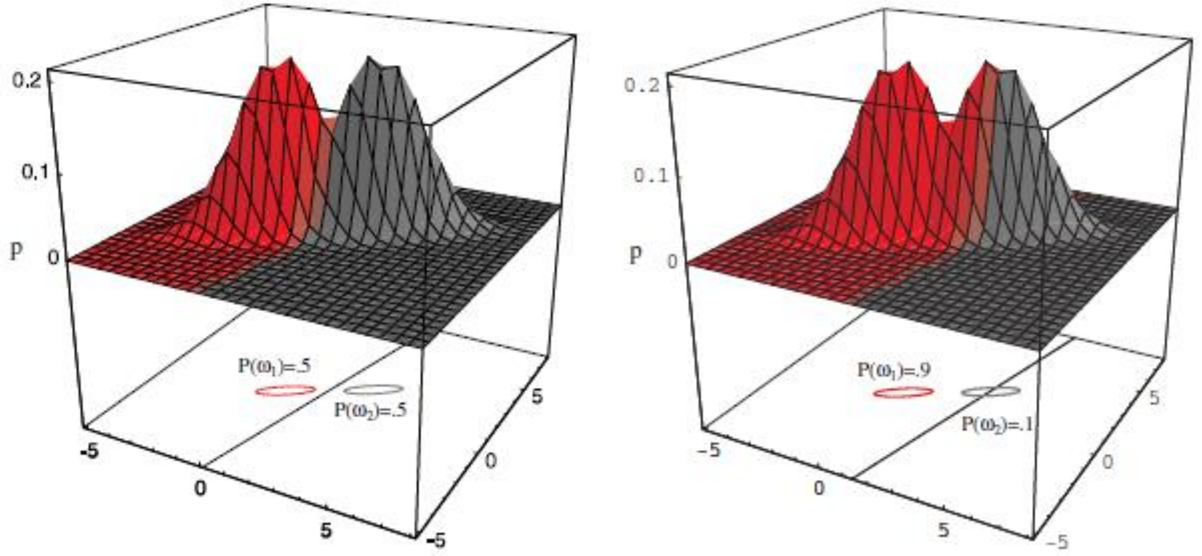




[3]

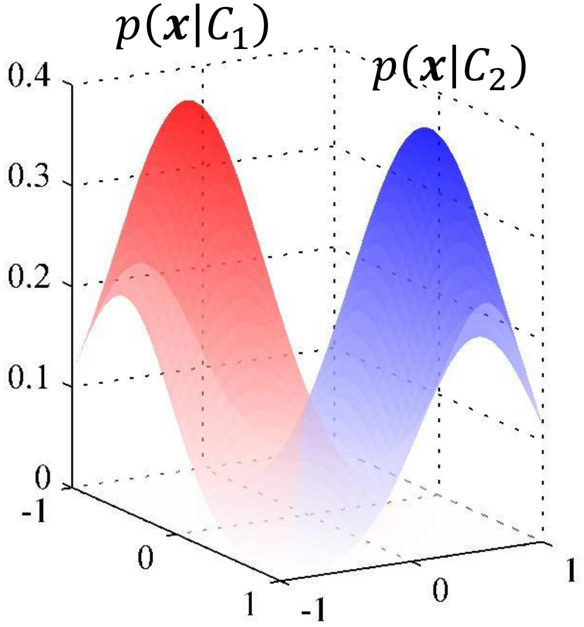
Gaussian discriminant analysis

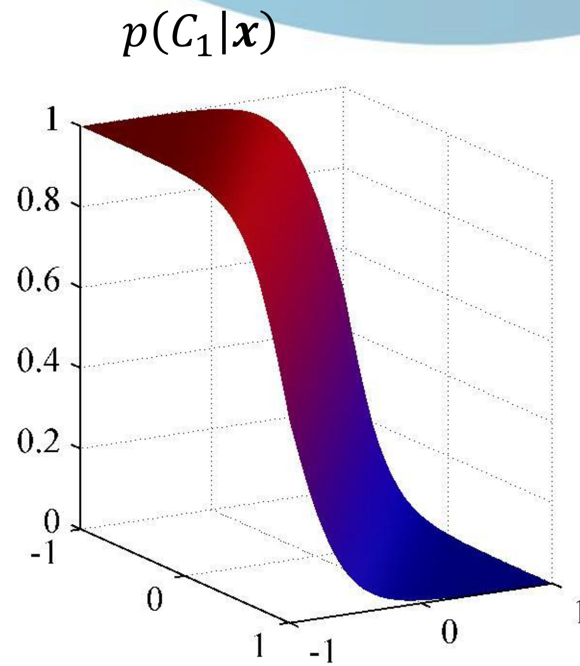
- A special case: $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$
  - Shared covariance matrix
    - Equivalent to LDA decision boundary
  - The decision surface is a Hyperplane, but is not necessarily orthogonal to the line between the means.



[3]

Class conditional densities vs. posterior





[I]

$$p(\mathcal{C}_1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

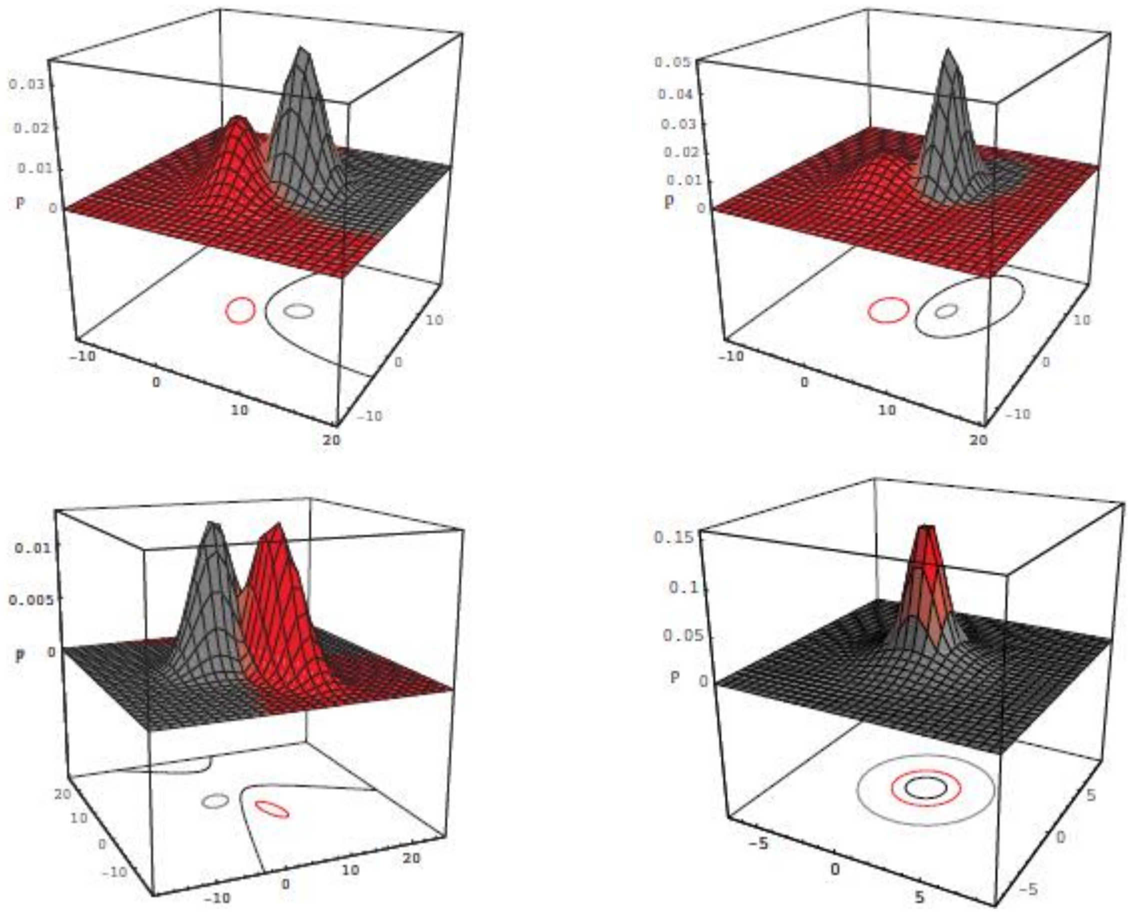
$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

$$\sigma(z) = \frac{1}{1 + \exp(z)}$$

Gaussian discriminant analysis

- A special case: $\boldsymbol{\Sigma}_k = arbitrary$
  - Decision boundaries are hyperquadrics



[3]

Naïve Bayes classifier

- Generative methods
  - High number of parameters

- Naïve Bayes assumption: Conditional independence of features

$$p(\boldsymbol{x}|C_k) = p(x_1|C_k) \times p(x_2|C_k) \times \cdots \times p(x_d|C_k)$$

Naïve Bayes classifier

- In the decision phase, it finds the label of $\boldsymbol{x}$ according to:

$$\underset{k=1,\dots,K}{\operatorname{argmax}}\, p(C_k|\boldsymbol{x})$$

$$\underset{k=1,\dots,K}{\operatorname{argmax}}\, p(C_k)\prod_{i=1}^{N} p(x_i|C_k)$$

$$p(\boldsymbol{x}|C_k) = p(x_1|C_k) \times p(x_2|C_k) \times \cdots \times p(x_d|C_k)$$

$$p(C_k|\boldsymbol{x}) \propto p(C_k)\prod_{i=1}^{n} p(x_i|C_k)$$

Naïve Bayes: discrete example

$$H = Yes \quad \equiv h$$
$$H = No \quad \equiv \bar{h}$$

Diabetes (D)	Smoke (S)	Heart Disease (H)
Y	N	Y
Y	N	N
N	Y	N
N	Y	N
N	N	N
N	Y	Y
N	N	N
N	Y	Y
N	N	N
Y	N	N

- $p(h) = 0.3$
- $p(d|h) = \frac{1}{3}$
- $p(s|h) = \frac{2}{3}$
- $p(d|\bar{h}) = \frac{2}{7}$
- $p(s|\bar{h}) = \frac{2}{7}$
- Decision on $\boldsymbol{x} = [d, \bar{s}]$ (a person that has diabetes but does not smoke):
  - $p(h|\boldsymbol{x}) \propto p(h)p(d|h)p(\bar{s}|h) = 1/14$
  - $p(\bar{h}|\boldsymbol{x}) \propto p(\bar{h})p(d|\bar{h})p(\bar{s}|\bar{h}) = 1/6$
  - Thus decide $H = No$

Naïve Bayes classifier

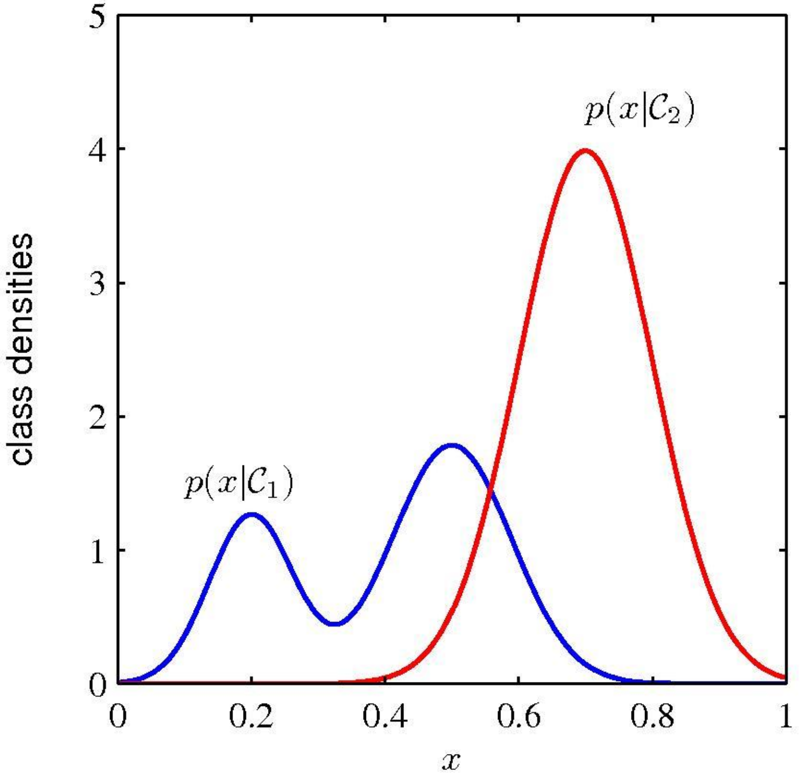
- Finds d univariate distributions $p(x_1|C_k), \cdots, p(x_d|C_k)$ instead of finding one multi-variate distribution $p(\boldsymbol{x}|C_k)$
  - Example 1: For Gaussian class-conditional density $p(\boldsymbol{x}|C_k)$, it finds $d + d$ (mean and sigma parameters on different dimensions) instead of $d + \frac{d(d+1)}{2}$ parameters
  - Example 2: For Bernoulli class-conditional density $p(\boldsymbol{x}|C_k)$, it finds d (mean parameters on different dimensions) instead of $2^{\mathrm{d}} - 1$ parameters

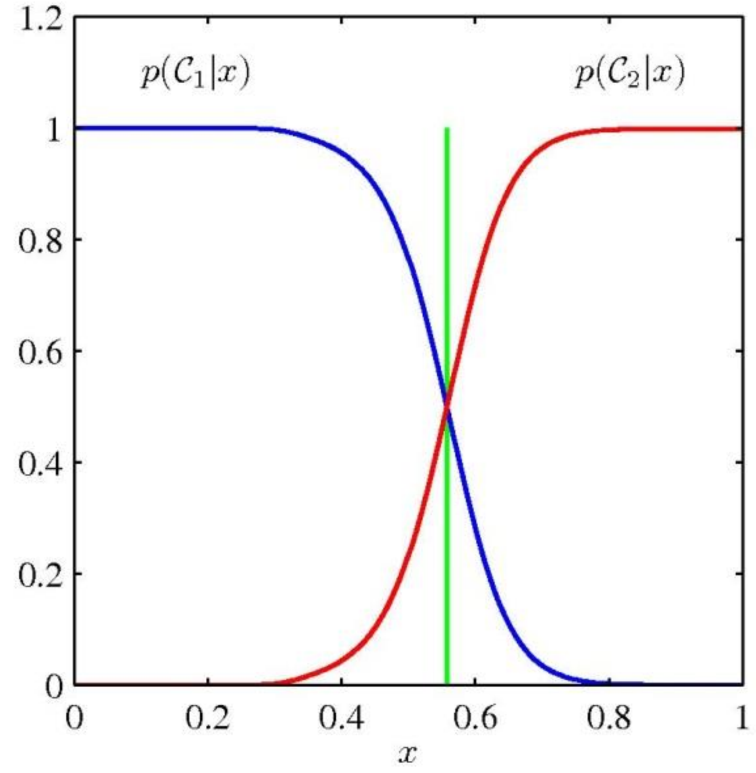
- It first estimates the class conditional densities $p(x_1|C_k), \cdots, p(x_d|C_k)$ and the prior probability $p(C_k)$ for each class $(k = 1, \ldots, K)$ based on the training set.

Probabilistic classifiers

- Probabilistic classification approaches can be divided in two main categories:
  - Generative
    - Estimate pdf $p(\boldsymbol{x}, \mathcal{C}_k)$ for each class $\mathcal{C}_k$ and then use it to find $p(\mathcal{C}_k|\boldsymbol{x})$
      - Or alternatively estimate both pdf $p(\boldsymbol{x}|\mathcal{C}_k)$ and $p(\mathcal{C}_k)$ to find $p(\mathcal{C}_k|\boldsymbol{x})$
  - Discriminative
    - Directly estimate $p(\mathcal{C}_k|\boldsymbol{x})$ for each class $\mathcal{C}_k$

Discriminative vs. generative approach





[I]

Discriminative approach: logistic regression

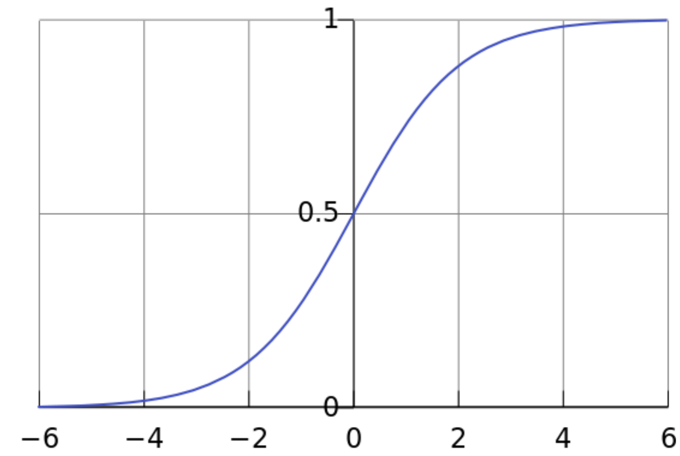
- $h(\boldsymbol{x}; \boldsymbol{w})$ predicts posterior probabilities $P(y = 1|\boldsymbol{x})$ $K = 2$

$$h(\boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{w}^T \boldsymbol{x})$$

$$\boldsymbol{x} = [1, x_1, \ldots, x_d]$$
$$\boldsymbol{w} = [w_0, w_1, \ldots, w_d]$$

- Sigmoid (logistic) function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Logistic regression

- $h(\boldsymbol{x}; \boldsymbol{w})$: probability that $y = 1$ given $\boldsymbol{x}$ (parameterized by $\boldsymbol{w}$)
  - $0 \le h(\boldsymbol{x}; \boldsymbol{w}) \le 1$

$$K = 2$$
$$y \in \{0,1\}$$

$$P(y = 1|\boldsymbol{x}; \boldsymbol{w}) = h(\boldsymbol{x}; \boldsymbol{w})$$
$$P(y = 0|\boldsymbol{x}; \boldsymbol{w}) = 1 - h(\boldsymbol{x}; \boldsymbol{w})$$

- Decision surface

$$h(\boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{w}^T \boldsymbol{x}) = \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{x})}} = 0.5$$

Logistic regression: ML estimation

- Maximum (conditional) log likelihood:

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}} \ \log \prod_{i=1}^{n} p\left(y^{(i)} \middle| \boldsymbol{w}, \boldsymbol{x}^{(i)}\right)$$

$$p\left(y^{(i)} \middle| \boldsymbol{w}, \boldsymbol{x}^{(i)}\right) = h\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)^{y^{(i)}} \left(1 - h\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right)^{(1-y^{(i)})}$$

$$\log\ p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) = \sum_{i=1}^{n} \left[y^{(i)} \log\left(h\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right) + (1 - y^{(i)}) \log\left(1 - h\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right)\right]$$

Logistic regression: cost function

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, J(\boldsymbol{w})$$

$$J(\boldsymbol{w}) = -\sum_{i=1}^{N} \log p\left(y^{(i)} \middle| \boldsymbol{w}, \boldsymbol{x}^{(i)}\right)$$
$$= \sum_{i=1}^{N} -y^{(i)} \log\left(h\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right) - (1 - y^{(i)}) \log\left(1 - h\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right)$$

- No closed form solution for

$$\nabla_{\boldsymbol{w}} J(\boldsymbol{w}) = 0$$

- However $J(\boldsymbol{w})$ is convex.
  - Global optimum can be found by gradient ascent

Logistic regression: Gradient descent

- The gradient descent update

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \nabla_{\boldsymbol{w}} J(\boldsymbol{w}^t)$$

$$\nabla_{\boldsymbol{w}} J(\boldsymbol{w}) = \sum_{i=1}^{N} \left(h\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right) - y^{(i)}\right)\boldsymbol{x}^{(i)}$$

- Is it similar to gradient of SSE for linear regression?

$$\nabla_{\boldsymbol{w}} J(\boldsymbol{w}) = \sum_{i=1}^{N} \left(\boldsymbol{w}^T \boldsymbol{x}^{(i)} - y^{(i)}\right)\boldsymbol{x}^{(i)}$$

Posterior probabilities

- Two-class LR

$$p(\mathcal{C}_1|\boldsymbol{x}) = \sigma(a(\boldsymbol{x})) = \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{x})}}$$

- Multi-class: $p(\mathcal{C}_k|\boldsymbol{x})$ can be written as a soft-max function

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{e^{-(\boldsymbol{w}_k^T \boldsymbol{x})}}{\sum_{j=1}^{K} e^{-(\boldsymbol{w}_j^T \boldsymbol{x})}}$$

- To make a prediction:

$$\alpha(\boldsymbol{x}) = \underset{k=1,\ldots,K}{\operatorname{argmax}}\, h_k(\boldsymbol{x})$$

Logistic regression: multi-class

- The gradient descent update

$$\widehat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmin}} J(\boldsymbol{W}) \qquad \boldsymbol{W} = [\boldsymbol{w}_1 \quad \cdots \quad \boldsymbol{w}_K]$$

$$\begin{aligned} J(\boldsymbol{W}) &= -\log \prod_{i=1}^{n} p\left(\boldsymbol{y}^{(i)} \middle| \boldsymbol{x}^{(i)}, \boldsymbol{W}\right) \\ &= -\log \prod_{i=1}^{n} \prod_{k=1}^{K} h_k\left(\boldsymbol{x}^{(i)}; \boldsymbol{W}\right)^{y_k^{(i)}} \\ &= -\sum_{i=1}^{n} \sum_{k=1}^{K} y_k^{(i)} \log\left(h_k\left(\boldsymbol{x}^{(i)}; \boldsymbol{W}\right)\right) \end{aligned}$$

$\boldsymbol{y}$ is a vector of length K (1-of-K coding)
e.g., $\boldsymbol{y} = [0,0,1,0]^T$ when the target class is C_3

Logistic regression: multi-class

- The gradient descent update

$$\boldsymbol{w}_j^{t+1} = \boldsymbol{w}_j^t - \eta \nabla_{\boldsymbol{W}} J(\boldsymbol{W}^t)$$

$$\nabla_{\boldsymbol{w}_j} J(\boldsymbol{W}) = \sum_{i=1}^{n} \left(h_j\left(\boldsymbol{x}^{(i)}; \boldsymbol{W}\right) - y_j^{(i)} \right) \boldsymbol{x}^{(i)}$$

LR vs. GDA

- d-dimensional feature space
  - Logistic regression: $d+1$ parameters
$$\boldsymbol{w} = (w_0, w_1, \ldots, w_d)$$
  - GDA with shared covariance matrix
    - $2d$ parameters for means
    - $d(d+1)/2$ parameters for shared covariance matrix
    - one parameter for class prior $p(C_1)$

- LR is more robust, less sensitive to incorrect modeling assumptions

Summary of alternatives

- Generative
  - Most demanding, because it finds the joint distribution $p(\boldsymbol{x}, \mathcal{C}_k)$
  - Usually needs a large training set to find $p(\boldsymbol{x}|\mathcal{C}_k)$
  - Can find $p(\boldsymbol{x}) \Rightarrow$ Outlier detection

- Discriminative
  - Specifies what is really needed (i.e., $p(\mathcal{C}_k|\boldsymbol{x})$)
  - More computationally efficient

Generalization of Bayes decision rule
Minimizing Bayes risk (expected loss)

$$E_{\boldsymbol{x},y}[L(\alpha(\boldsymbol{x}),y)]$$

$$= \int \sum_{j=1}^{K} L(\alpha(\boldsymbol{x}),\mathcal{C}_j)p(\boldsymbol{x},\mathcal{C}_j)d\boldsymbol{x}$$

$$= \int p(\boldsymbol{x}) \underbrace{\sum_{j=1}^{K} L(\alpha(\boldsymbol{x}),\mathcal{C}_j)p(\mathcal{C}_j|\boldsymbol{x})}_{\text{for each } \boldsymbol{x} \text{ minimize it that is called conditional risk}} d\boldsymbol{x}$$

- Bayes minimum loss (risk) decision rule: $\hat{\alpha}(\boldsymbol{x})$

$$\hat{\alpha}(\boldsymbol{x}) = \underset{i=1,\dots,K}{\operatorname{argmin}} \sum_{j=1}^{K} L_{ij}\, p(\mathcal{C}_j|\boldsymbol{x})$$

The loss of assigning a sample to $\mathcal{C}_i$ where the correct class is $\mathcal{C}_j$

Minimizing expected loss: special case (loss = misclassification rate)

- Problem definition for this special case:
  - If action $\alpha(\boldsymbol{x}) = i$ is taken and the true category is $\mathcal{C}_j$, then the decision is correct if $i = j$ and otherwise it is incorrect.
    - Zero-one loss function:

$$L_{ij} = 1 - \delta_{ij} = \begin{cases} 0 & i = j \\ 1 & o.w. \end{cases}$$

$$\hat{\alpha}(\boldsymbol{x}) = \underset{i=1,\dots,K}{\operatorname{argmin}} \sum_{j=1}^{K} L_{ij} p(\mathcal{C}_j|\boldsymbol{x})$$

$$= \underset{i=1,\dots,K}{\operatorname{argmin}}\, 0 \times p(\mathcal{C}_i|\boldsymbol{x}) + \sum_{j \neq i} p(\mathcal{C}_j|\boldsymbol{x})$$

$$= \underset{i=1,\dots,K}{\operatorname{argmin}}\, 1 - p(\mathcal{C}_i|\boldsymbol{x}) = \underset{i=1,\dots,K}{\operatorname{argmax}}\, p(\mathcal{C}_i|\boldsymbol{x})$$

Resources

- [1] C. Bishop, “Pattern Recognition and Machine Learning”, Chapter 4.2-4.3.
- [2]: Andrew Ng, Machine learning, Stanford
- [3]: Pattern classification, Duda, Hart & Stork, 2002
- [4]: Mahdieh Soleymani, Machine learning, Sharif university of technology