Probabilistic classification

CE-717: Machine Learning

Sharif University of Technology

M. Soleymani

Fall 2016

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

$K = 2$

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

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

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

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

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

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

Bayes minimum error

- Bayes minimum error classifier:

$$\min_{\alpha(.)} E_{\boldsymbol{x},y}[I(\alpha(\boldsymbol{x}) \neq y)] \qquad \text{Zero-one loss}$$

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

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

Bayes theorem

- Bayes' theorem

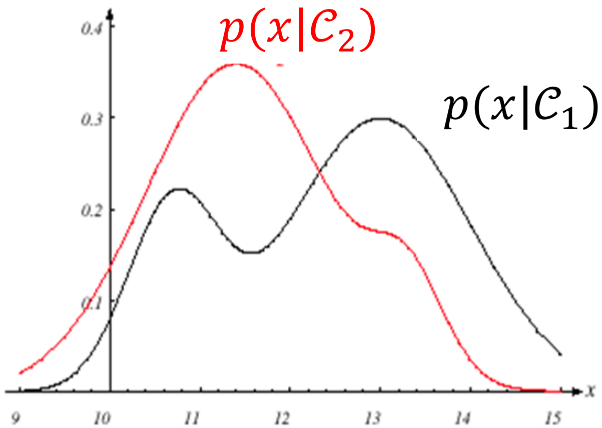
$$\underbrace{p(\mathcal{C}_k|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overbrace{p(\boldsymbol{x}|\mathcal{C}_k)}^{\text{Likelihood}}\overbrace{p(\mathcal{C}_k)}^{\text{Prior}}}{p(\boldsymbol{x})}$$

- 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 $\mathcal{C}_k$

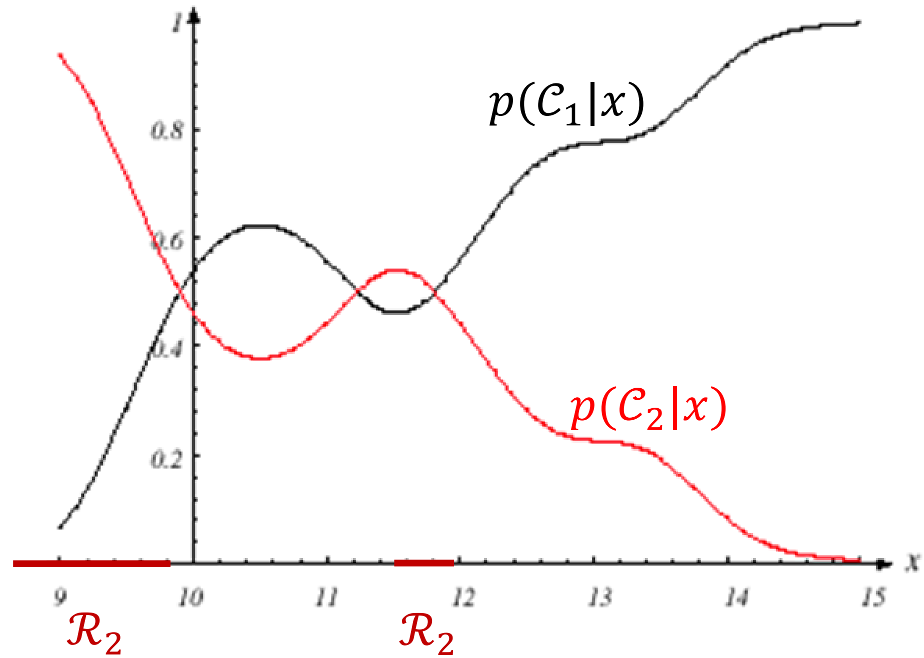
Bayes decision rule: example

- 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

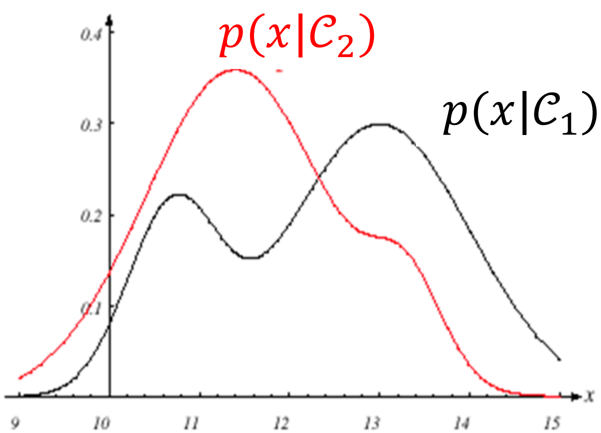
- If $\frac{p(\boldsymbol{x}|\mathcal{C}_1)P(\mathcal{C}_1)}{p(\boldsymbol{x})} > \frac{p(\boldsymbol{x}|\mathcal{C}_2)P(\mathcal{C}_2)}{p(\boldsymbol{x})}$ decide $\mathcal{C}_1$
  otherwise decide $\mathcal{C}_2$

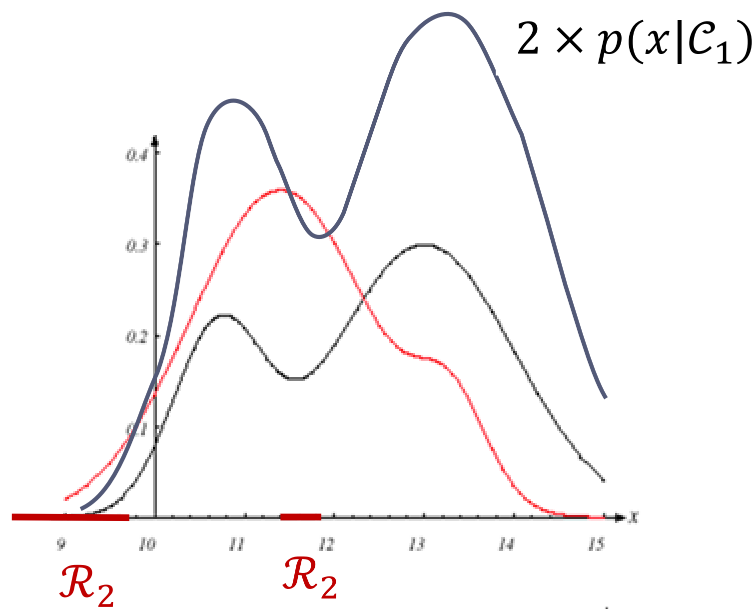
Equivalent

- 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

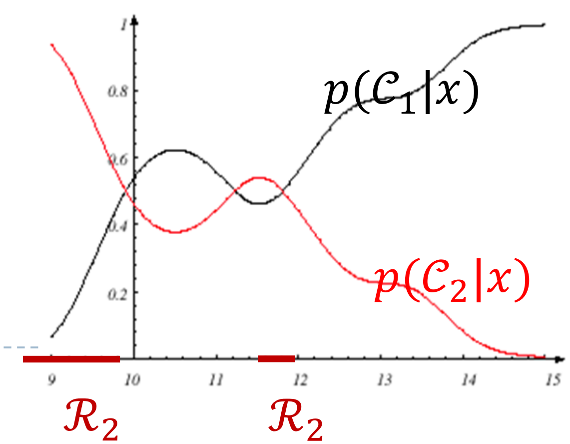
- Bayes decision: Choose the class with highest $p(\mathcal{C}_k|\boldsymbol{x})$





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

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



Bayes Classier

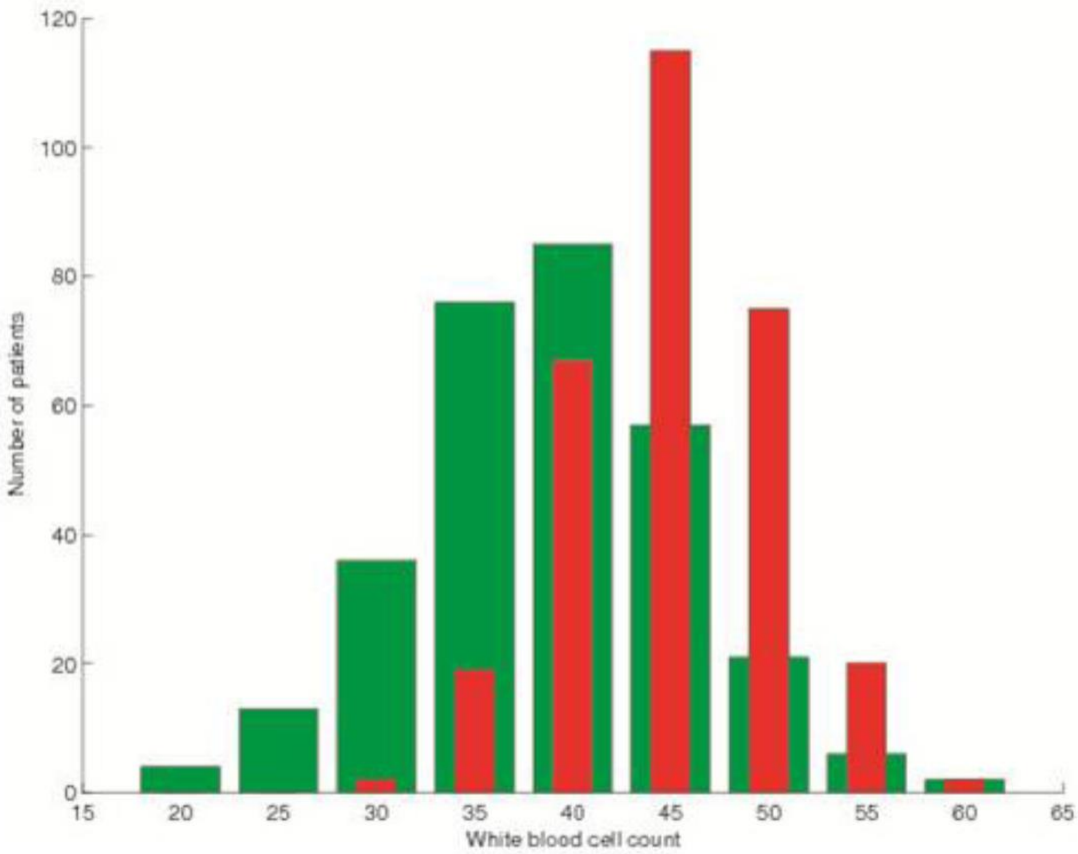
- Simple Bayes classifier: estimate posterior probability of each class

- 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

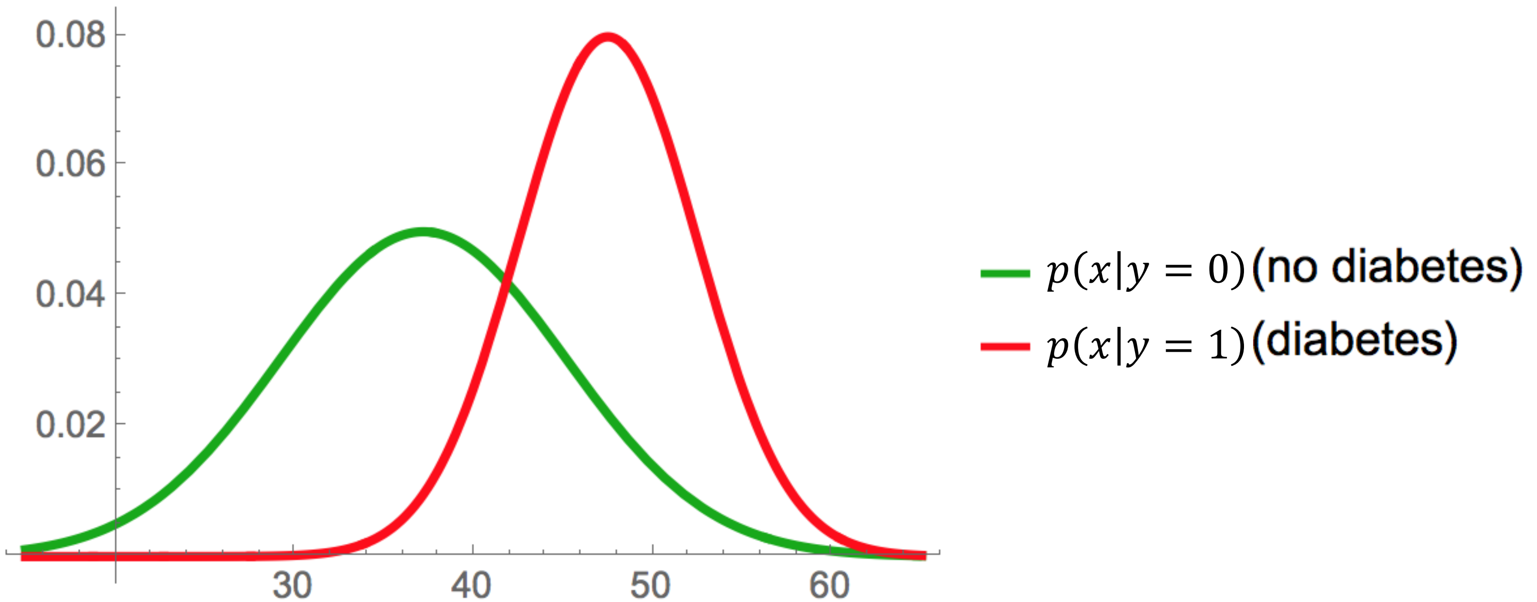
Diabetes example

- 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



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

Estimate probability densities from data

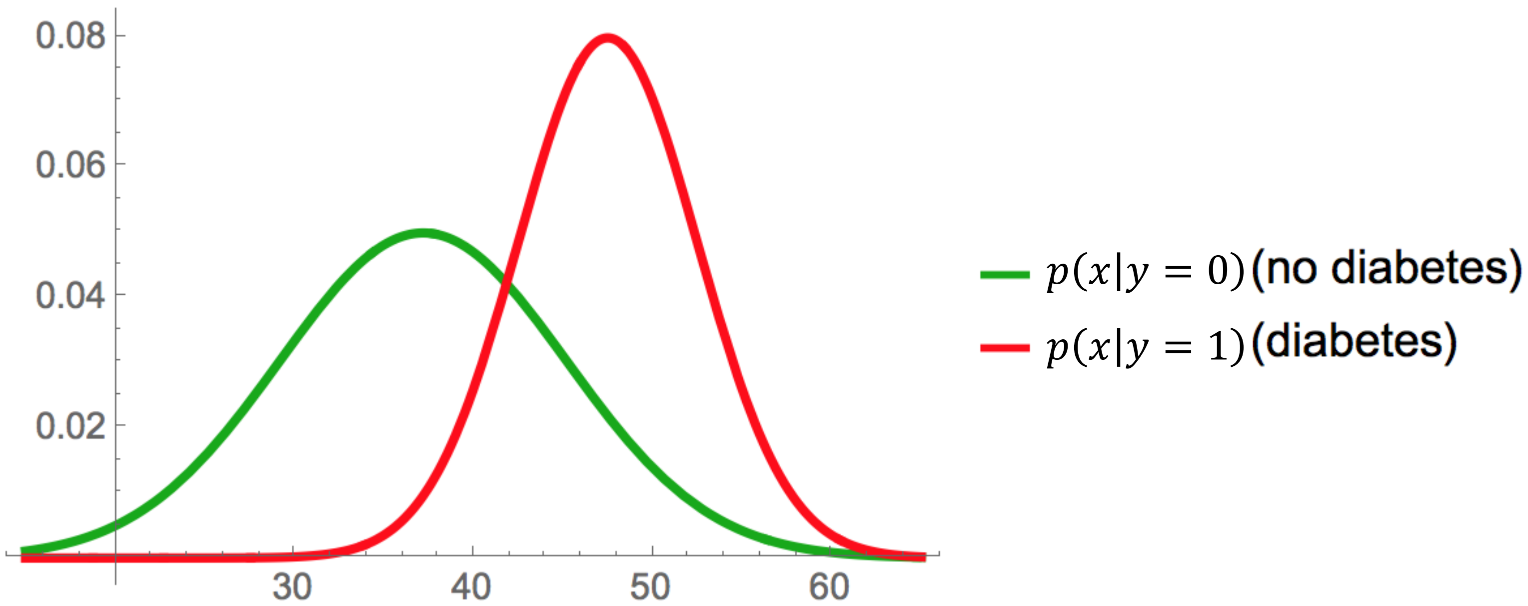
- If we assume Gaussian distributions for $p(x|\mathcal{C}_1)$ and $p(x|\mathcal{C}_2)$

- Recall that for samples $\{x^{(1)}, \ldots, x^{(N)}\}$, 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



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

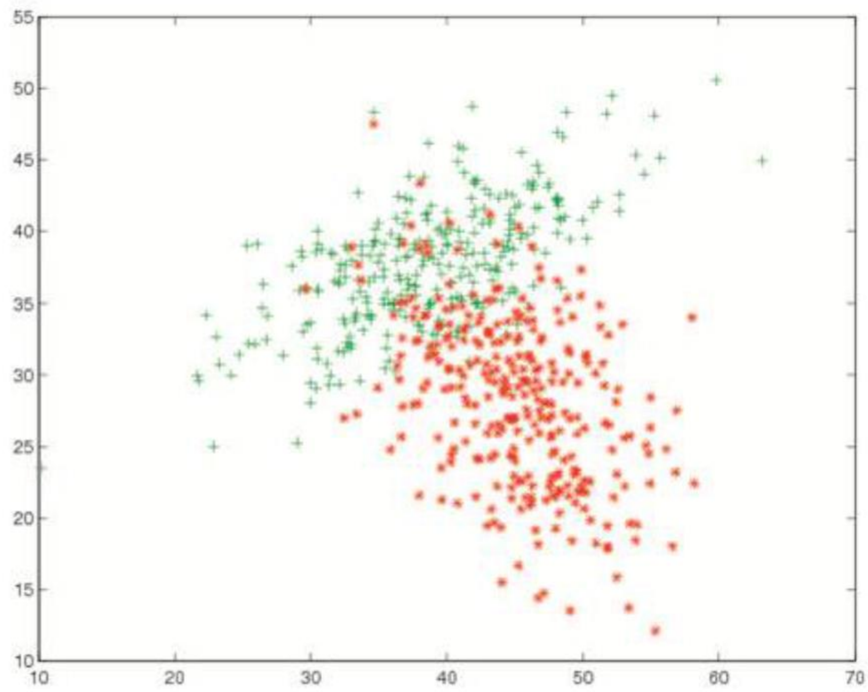
$$\mu_1 = \frac{\sum_{n:\, y^{(n)}=1} x^{(n)}}{\sum_{n:\, y^{(n)}=1} 1} = \frac{\sum_{n:\, y^{(n)}=1} x^{(n)}}{N_1}$$

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

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 $p(x|\mathcal{C}_k)$:

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

$$k = 1,2$$

- Prior distribution $p(x|\mathcal{C}_k)$:
  - $p(y=1)=\pi, \qquad p(y=0)=1-\pi$

MLE for multivariate Gaussian

- For samples $\{x^{(1)}, \ldots, x^{(N)}\}$, 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

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

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

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

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

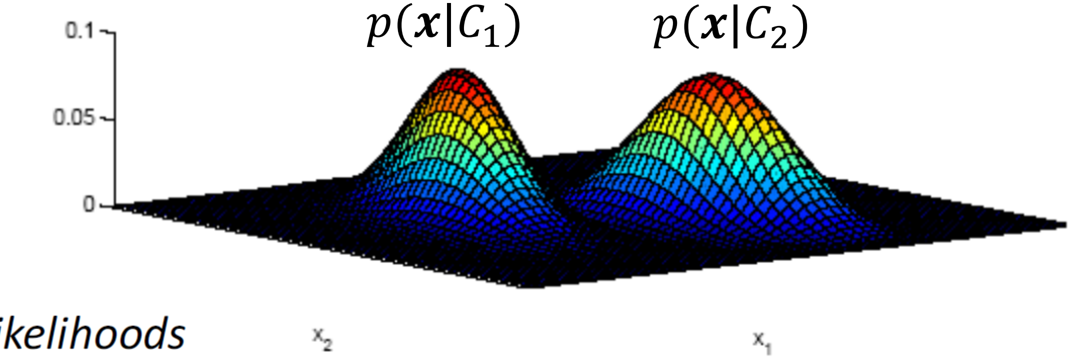
$$\ln p(\boldsymbol{x}|\mathcal{C}_1) + \ln p(\mathcal{C}_1) - \ln p(\boldsymbol{x})$$
$$= \ln p(\boldsymbol{x}|\mathcal{C}_2) + \ln p(\mathcal{C}_2) - \ln p(\boldsymbol{x})$$

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

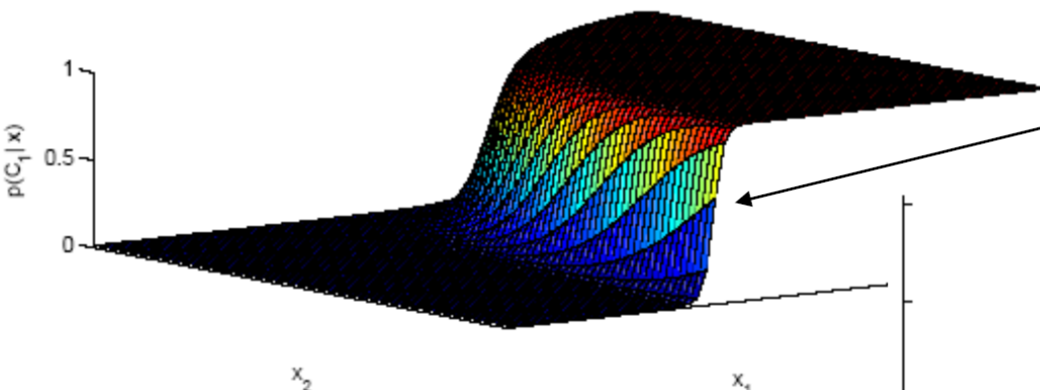
$$\ln p(\boldsymbol{x}|\mathcal{C}_k)$$
$$= -\frac{d}{2}\ln 2\pi - \frac{1}{2}\ln\left|\boldsymbol{\Sigma}_k^{-1}\right| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)$$

Decision boundary

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



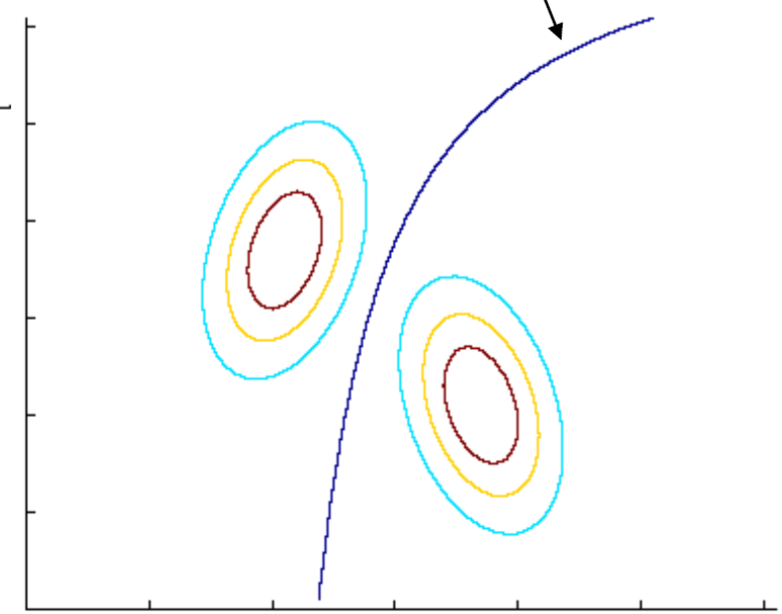
likelihoods



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

discriminant:

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



Shared covariance matrix

- When classes share a single covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$

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

$$k = 1,2$$

- $p(C_1) = \pi, \qquad p(C_2) = 1 - \pi$

Likelihood

$$\prod_{n=1}^{N} p(\boldsymbol{x}^{(n)}, y^{(n)} | \pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$

$$= \prod_{n=1}^{N} p(\boldsymbol{x}^{(n)} | y^{(n)}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) p(y^{(n)} | \pi)$$

Shared covariance matrix

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

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

$$\boldsymbol{\mu}_1 = \frac{\sum_{n=1}^{N} y^{(n)} \boldsymbol{x}^{(n)}}{N_1}$$

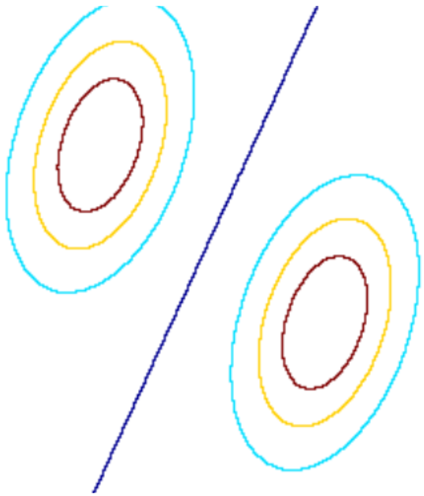
$$\boldsymbol{\mu}_2 = \frac{\sum_{n=1}^{N}(1 - y^{(n)})\boldsymbol{x}^{(n)}}{N_2}$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\left(\sum_{n \in C_1}\left(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_1\right)\left(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_1\right)^T + \sum_{n \in C_2}\left(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_2\right)\left(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_2\right)^T\right)$$

Decision boundary when shared covariance matrix

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

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



Bayes decision rule
Multi-class misclassification rate

- Multi-class problem: Probability of error of Bayesian decision rule
  - Simpler to compute the probability of correct decision

$$P(error) = 1 - P(correct)$$

$$P(Correct) = \sum_{i=1}^{K} \int_{\mathcal{R}_i} p(\boldsymbol{x}, \mathcal{C}_i)\, d\boldsymbol{x}$$
$$= \sum_{i=1}^{K} \int_{\mathcal{R}_i} p(\mathcal{C}_i|\boldsymbol{x}) p(\boldsymbol{x})\, d\boldsymbol{x}$$

$\mathcal{R}_i$: the subset of feature space assigned to the class $\mathcal{C}_i$ using the classifier

Bayes minimum error

- Bayes minimum error classifier:

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

$$\alpha(\boldsymbol{x}) = \underset{y}{\operatorname{argmax}}\, p(y|\boldsymbol{x})$$

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: $\hat{\alpha}(\boldsymbol{x})$

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

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:

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

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

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

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

Probabilistic discriminant functions

- **Discriminant functions**: A popular way of representing a classifier
  - A discriminant function $f_i(\boldsymbol{x})$ for each class $\mathcal{C}_i$ ($i = 1, \dots, K$):
    - $\boldsymbol{x}$ is assigned to class $\mathcal{C}_i$ if:

$$f_i(\boldsymbol{x}) > f_j(\boldsymbol{x}) \ \ \forall j \neq i$$

- Representing Bayesian classifier using discriminant functions:
  - Classifier minimizing error rate: $f_i(\boldsymbol{x}) = P(\mathcal{C}_i|\boldsymbol{x})$
  - Classifier minimizing risk: $f_i(\boldsymbol{x}) = -\sum_{j=1}^{K} L_{ij} p(\mathcal{C}_j|\boldsymbol{x})$

Naïve Bayes classifier

- Generative methods
  - High number of parameters

- Assumption: Conditional independence

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

Naïve Bayes classifier

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

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

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

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

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

Naïve Bayes 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.

Naïve Bayes: discrete example

- $p(H = Yes) = 0.3$

- $p(D = Yes|H = Yes) = \frac{1}{3}$
- $p(S = Yes|H = Yes) = \frac{2}{3}$

- $p(D = Yes|H = No) = \frac{2}{7}$
- $p(S = Yes|H = No) = \frac{2}{7}$

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

- Decision on $\boldsymbol{x} = [Yes, Yes]$ (a person that has diabetes and also smokes):
  - $p(H = Yes|\boldsymbol{x}) \propto p(H = Yes)p(D = yes|H = Yes)p(S = yes|H = Yes) = 0.066$
  - $p(H = No|\boldsymbol{x}) \propto p(H = No)p(D = yes|H = No)p(S = yes|H = No) = 0.057$
  - Thus decide $H = yes$

Probabilistic classifiers

- How can we find the probabilities required in the Bayes decision rule?

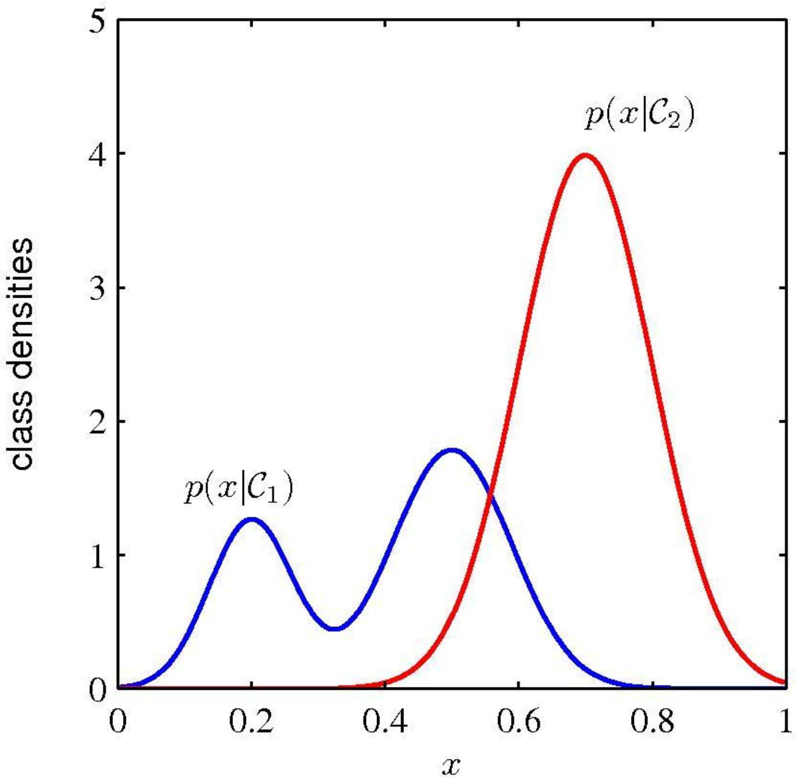
- Probabilistic classification approaches can be divided in two main categories:
  - Generative
    - Estimate pdf $p(\boldsymbol{x}, \mathcal{C}_k)$ for each class $\mathcal{C}_k$ and then use it to find $p(\mathcal{C}_k|\boldsymbol{x})$
      - or alternatively estimate both pdf $p(\boldsymbol{x}|\mathcal{C}_k)$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

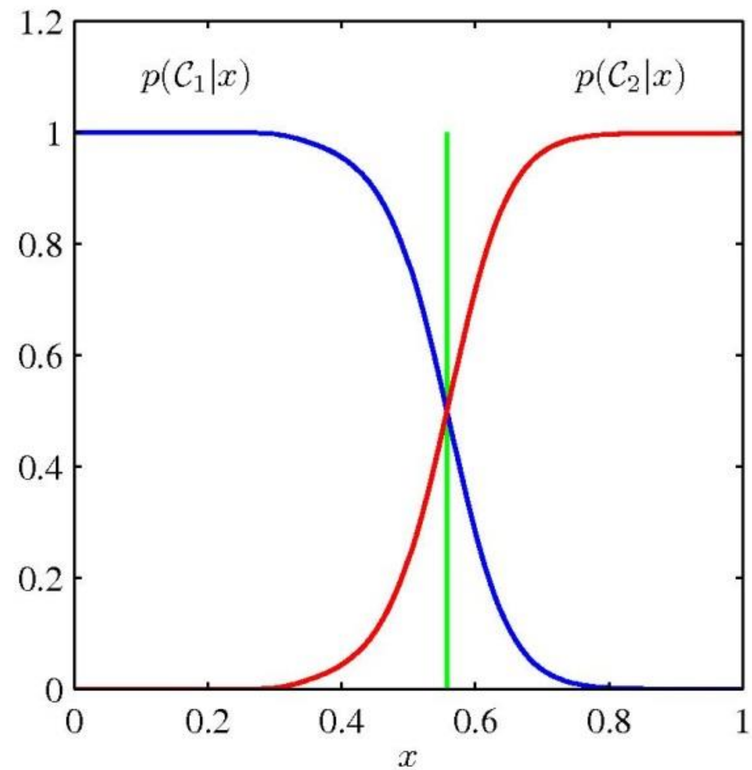
Generative approach

- <u>Inference stage</u>
  - Determine class conditional densities $p(\boldsymbol{x}|\mathcal{C}_k)$ and priors $p(\mathcal{C}_k)$
  - Use the Bayes theorem to find $p(\mathcal{C}_k|\boldsymbol{x})$

- <u>Decision stage</u>: After learning the model (inference stage), make optimal class assignment for new input
  - if $p(\mathcal{C}_i|\boldsymbol{x}) > p(\mathcal{C}_j|\boldsymbol{x}) \quad \forall j \neq i$ then decide $\mathcal{C}_i$

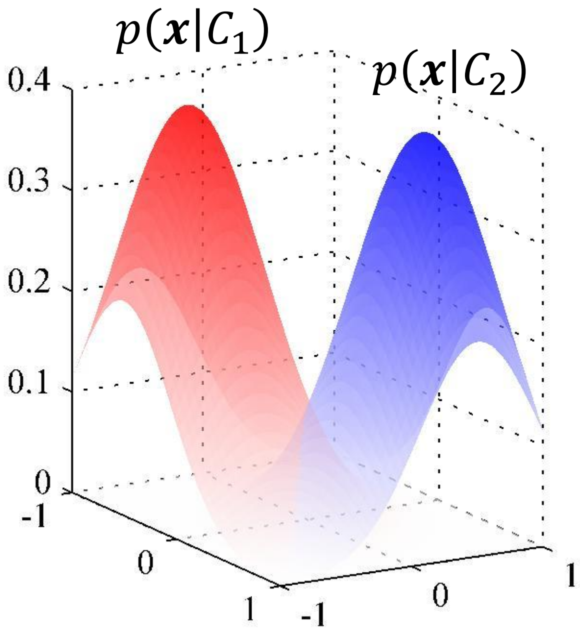
Discriminative vs. generative approach

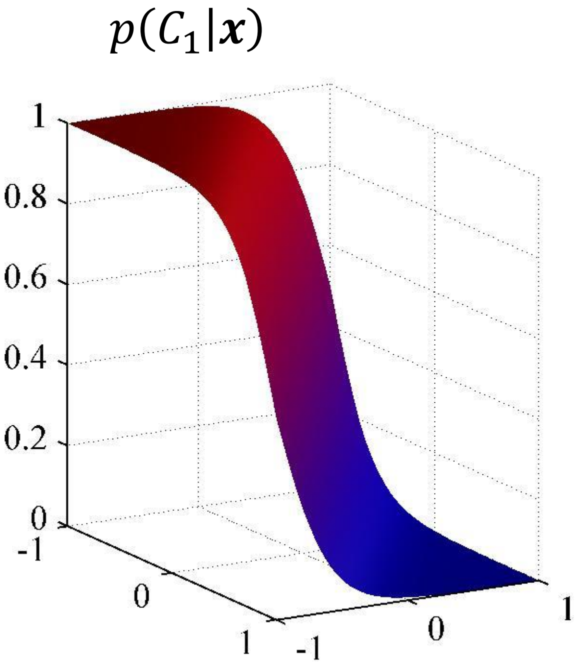




[Bishop]

Class conditional densities vs. posterior





[Bishop]

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

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

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

Discriminative approach

- Inference stage
  - Determine the posterior class probabilities $P(\mathcal{C}_k|\boldsymbol{x})$ directly

- Decision stage: After learning the model (inference stage), make optimal class assignment for new input
  - if $P(\mathcal{C}_i|\boldsymbol{x}) > P(\mathcal{C}_j|\boldsymbol{x}) \;\; \forall j \neq i$ then decide $\mathcal{C}_i$

Posterior probabilities

- Two-class: $p(\mathcal{C}_k|\boldsymbol{x})$ can be written as a logistic sigmoid for a wide choice of $p(\boldsymbol{x}|\mathcal{C}_k)$ distributions

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

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

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{\exp(a_k(\boldsymbol{x}))}{\sum_{j=1}^{K}\exp(a_j(\boldsymbol{x}))}$$

Discriminative approach: logistic regression

$K = 2$

- More general than discriminant functions:
  - $f(\boldsymbol{x}; \boldsymbol{w})$ predicts posterior probabilities $P(y = 1|\boldsymbol{x})$

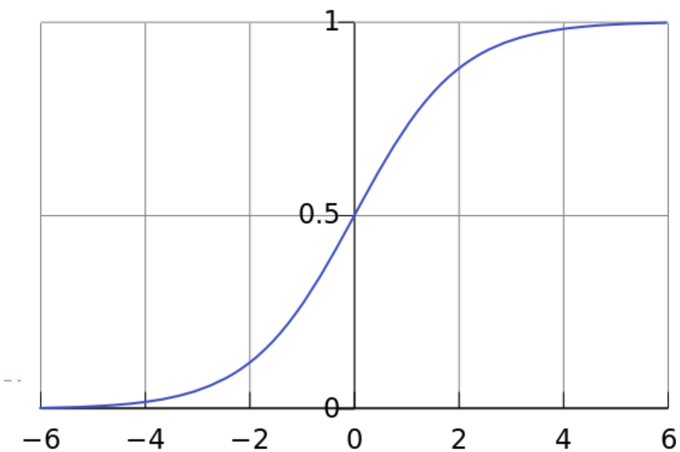
$$f(\boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{w}^T \boldsymbol{x})$$

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

$\sigma(.)$ is an activation function

- Sigmoid (logistic) function
  - Activation function

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



Logistic regression

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

$$P(y = 1|\boldsymbol{x},\boldsymbol{w}) = f(\boldsymbol{x};\boldsymbol{w})$$

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

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

$f(\boldsymbol{x};\boldsymbol{w}) = \sigma(\boldsymbol{w}^T\boldsymbol{x})$
$0 \le f(\boldsymbol{x};\boldsymbol{w}) \le 1$
estimated probability of $y = 1$ on input x

- Example: Cancer (Malignant, Benign)
  - $f(\boldsymbol{x};\boldsymbol{w}) = 0.7$
  - 70% chance of tumor being malignant

Logistic regression: Decision surface

- Decision surface $f(\boldsymbol{x};\boldsymbol{w}) = \text{constant}$
  - $f(\boldsymbol{x};\boldsymbol{w}) = \sigma(\boldsymbol{w}^T\boldsymbol{x}) = \frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x})}} = 0.5$

- Decision surfaces are linear functions of $\boldsymbol{x}$

if $f(\boldsymbol{x};\boldsymbol{w}) \geq 0.5$ then $y = 1$
else $y = 0$

Equivalent to

if $\boldsymbol{w}^T\boldsymbol{x} + w_0 \geq 0$ then $y = 1$
else $y = 0$

Logistic regression: ML estimation

- Maximum (conditional) log likelihood:

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

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

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

Logistic regression: cost function

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

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

$$= \sum_{i=1}^{n} -y^{(i)} \log\left(f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right) - (1 - y^{(i)}) \log\left(1 - f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right)$$

- No closed form solution for

$$\boldsymbol{\nabla}_{\boldsymbol{w}} J(\boldsymbol{w}) = 0$$

- However $J(\boldsymbol{w})$ is convex.

Logistic regression: Gradient descent

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

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

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

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

Logistic regression: loss function

$$\text{Loss}\big(y, f(\boldsymbol{x};\boldsymbol{w})\big) = -y \times \log\big(f(\boldsymbol{x};\boldsymbol{w})\big) - (1-y) \times \log(1 - f(\boldsymbol{x};\boldsymbol{w}))$$

Since $y = 1$ or $y = 0$ $\Rightarrow$ $\text{Loss}\big(y, f(\boldsymbol{x};\boldsymbol{w})\big) = \begin{cases} -\log(f(\boldsymbol{x};\boldsymbol{w})) & \text{if } y = 1 \\ -\log(1 - f(\boldsymbol{x};\boldsymbol{w})) & \text{if } y = 0 \end{cases}$

How is it related to zero-one loss?

$$\text{Loss}(y, \hat{y}) = \begin{cases} 1 & y \neq \hat{y} \\ 0 & y = \hat{y} \end{cases}$$

$$f(\boldsymbol{x};\boldsymbol{w}) = \frac{1}{1 + exp(-\boldsymbol{w}^T\boldsymbol{x})}$$

Logistic regression: cost function (summary)

- Logistic Regression (LR) has a more proper cost function for classification than SSE and Perceptron

- Why is the cost function of LR also more suitable than?

$$J(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right)^2$$

  where $f(\boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{w}^T\boldsymbol{x})$
  - The conditional distribution $p(y|\boldsymbol{x}, \boldsymbol{w})$ in the classification problem is not Gaussian (it is Bernoulli)
  - The cost function of LR is also convex

Multi-class logistic regression

- For each class k, $f_k(\boldsymbol{x}; \boldsymbol{W})$ predicts the probability of $y = k$
  - i.e., $P(y = k | \boldsymbol{x}, \boldsymbol{W})$

- On a new input $\boldsymbol{x}$, to make a prediction, pick the class that maximizes $f_k(\boldsymbol{x}; \boldsymbol{W})$:

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

if $f_k(\boldsymbol{x}) > f_j(\boldsymbol{x}) \quad \forall j \neq k$ then
decide C_k

Multi-class logistic regression

$K > 2$

$y \in \{1, 2, \dots, K\}$

$$f_k(\boldsymbol{x}; \boldsymbol{W}) = p(y = k|\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_k^T \boldsymbol{x})}{\sum_{j=1}^{K} \exp(\boldsymbol{w}_j^T \boldsymbol{x})}$$

- Normalized exponential (aka softmax)
  - If $\boldsymbol{w}_k^T \boldsymbol{x} \gg \boldsymbol{w}_j^T \boldsymbol{x}$ for all $j \neq k$ then $p(C_k|\boldsymbol{x}) \simeq 1, p(C_j|\boldsymbol{x}) \simeq 0$

$$p(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)p(C_k)}{\sum_{j=1}^{K} p(\boldsymbol{x}|C_j)p(C_j)}$$

Logistic regression: multi-class

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

$$J(\boldsymbol{W}) = -\log \prod_{i=1}^{n} p\left(\boldsymbol{y}^{(i)} \middle| \boldsymbol{x}^{(i)}, \boldsymbol{W}\right)$$

$$= -\log \prod_{i=1}^{n} \prod_{k=1}^{K} f_k\left(\boldsymbol{x}^{(i)}; \boldsymbol{W}\right)^{y_k^{(i)}}$$

$$= -\sum_{i=1}^{n} \sum_{k=1}^{K} y_k^{(i)} \log\left(f_k\left(\boldsymbol{x}^{(i)}; \boldsymbol{W}\right)\right)$$

$\boldsymbol{y}$ 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

$$\boldsymbol{W} = [\boldsymbol{w}_1 \quad \cdots \quad \boldsymbol{w}_K]$$

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}^{(1)} \\ \vdots \\ \boldsymbol{y}^{(n)} \end{bmatrix} = \begin{bmatrix} y_1^{(1)} & \cdots & y_K^{(1)} \\ \vdots & \ddots & \vdots \\ y_1^{(n)} & \cdots & y_K^{(n)} \end{bmatrix}$$

Logistic regression: multi-class

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

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

Logistic Regression (LR): summary

- LR is a linear classifier
- LR optimization problem is obtained by maximum likelihood
  - when assuming Bernoulli distribution for conditional probabilities whose mean is $\frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x})}}$
- No closed-form solution for its optimization problem
  - But convex cost function and global optimum can be found by gradient ascent

Discriminative vs. generative: number of parameters

- d-dimensional feature space

- Logistic regression: $d + 1$ parameters
  - $\boldsymbol{w} = (w_0, w_1, .., w_d)$

- Generative approach:
  - Gaussian class-conditionals with shared covariance matrix
    - $2d$ parameters for means
    - $d(d + 1)/2$ parameters for shared covariance matrix
    - one parameter for class prior $p(C_1)$.

- But 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 or novelty detection

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

Resources

- C. Bishop, “Pattern Recognition and Machine Learning”, Chapter 4.2-4.3.