Probabilistic Modeling

Shan-Hung Wu
shwu@cs.nthu.edu.tw

Department of Computer Science,
National Tsing Hua University, Taiwan

NetDB-ML, Spring 2015

Outline

Summary of Supervised Learning Models

- Three main categories (either parametric or non-parametric):

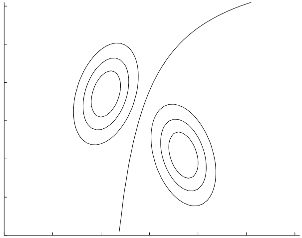
1. Those learning the ***discriminant functions*** f's (no probability interpretation)
    - E.g., perceptron, kNN, etc.
2. Those based on probability and learn $p(r|\boldsymbol{x})$ directly
    - E.g., linear regression, logistic regression, etc.
    - $p(r|\boldsymbol{x};\theta)$ with θ (constant) estimated from $\mathcal{X}$
    - Methods in 1 and 2 are called ***discriminative methods***
3. Those learn $p(r|\boldsymbol{x})$ indirectly from $p(\boldsymbol{x}|r)p(r)$
    - To be discussed later
    - These are called ***generative methods***, as $p(\boldsymbol{x}|r)p(r)$ explains how $\boldsymbol{x}$ (and $\mathcal{X}$) is generated

Probabilistic Modeling

- By assuming the target follows some probability distribution
- Pros and cons?

Probabilistic Modeling

- By assuming the target follows some probability distribution
- Pros and cons?
- Perform well only when the assumption holds
- Essentially solves a problem (i.e., distribution estimation) harder than discrimination
    - E.g., in generative models, if we let $p(\boldsymbol{x}|r) \sim \mathcal{N}(\mu, \Sigma)$, then we can plot the contour of each class in addition to the decision boundary
    - Less efficient; but more descriptive



More About Probabilistic Modeling (1)

- The roles of θ in the prediction function $p(r'|\boldsymbol{x}')$:
  - Constant, from ML estimation of θ:
    - $\theta_{ML} = \arg\max_\theta p(\mathcal{X}|\theta)$
    - $p(r|\boldsymbol{x}') := p(r|\boldsymbol{x}';\theta_{ML})$
  - Constant, from MAP estimation of θ:
    - $\theta_{MAP} = \arg\max_\theta p(\theta|\mathcal{X}) = \arg\max_\theta p(\mathcal{X}|\theta)p(\theta)$
    - $p(r|\boldsymbol{x}) := p(r|\boldsymbol{x};\theta_{MAP})$
  - ***Random variable***, for full Bayesian treatment:
    - $p(y|\boldsymbol{x},\mathcal{X}) = \int p(y,\theta|\boldsymbol{x}',\mathcal{X})d\theta$

More About Probabilistic Modeling (2)

- Can we analyze the generation performance more easily with the aid of distribution assumption?

More About Probabilistic Modeling (3)

- Generative models

The Roles of θ

- The roles of θ in the discrimination function $p(r'|\boldsymbol{x}')$:
  - Constant, from ML estimation of θ:
    - $\theta_{ML} = \arg\max_{\theta} p(\mathcal{X}|\theta)$
    - $p(r'|\boldsymbol{x}') := p(r|\boldsymbol{x};\theta_{ML})$
  - Constant, from MAP estimation of θ:
    - $\theta_{MAP} = \arg\max_{\theta} p(\theta|\mathcal{X}) = \arg\max_{\theta} p(\mathcal{X}|\theta)p(\theta)$
    - $p(r'|\boldsymbol{x}') := p(r'|\boldsymbol{x}';\theta_{MAP})$
  - ***Random variable***, for full Bayesian treatment of r':
    - $p(r'|\boldsymbol{x}',\mathcal{X}) = \int p(r',\theta|\boldsymbol{x}',\mathcal{X})d\theta$

ML Estimator for θ

- The estimators we discussed so far (e.g., ρ_i, $\boldsymbol{m}_i$, and $\boldsymbol{S}_i$ in classification and $\boldsymbol{w}$ in regression) are called the ***Maximum Likelihood (ML) estimators*** since they are derived from $\theta_{ML} = \arg_\theta \max p(\mathcal{X}|\theta)$
  - E.g., in linear regression where $\theta = \boldsymbol{w}$, given a new instance $\boldsymbol{x}'$, the prediction can be made by $y' = \arg_y \max p(y|\boldsymbol{x}'; \boldsymbol{w}_{ML}) = \arg_y \max \mathcal{N}(y|\boldsymbol{w}_{ML}^\top \boldsymbol{x}', \beta^{-1}) = \boldsymbol{w}_{ML}^\top \boldsymbol{x}'$

MAP Estimator for θ

- If we have the prior knowledge about θ (i.e., $P(\theta)$), we can obtain the ***Maximum A Posteriori (MAP) estimators*** based on $\theta_{MAP} = \arg_{\theta}\max P(\theta|\mathcal{X}) = \arg_{\theta}\max p(\mathcal{X}|\theta)P(\theta)$
  - If we assume that $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$ in linear regression, we have $\log p(\boldsymbol{w}|\mathcal{X}) = \log p(\mathcal{X}|\boldsymbol{w}) + \log p(\boldsymbol{w}) \propto -\frac{\beta}{2}\sum_{t=1}^{N}\left(r^{(t)} - \boldsymbol{w}^\top \boldsymbol{x}^{(t)}\right)^2 - \frac{\alpha}{2}\boldsymbol{w}^\top\boldsymbol{w}$ [Proof]
    - We effectively find $\boldsymbol{w}_{MAP}$ that minimizes $\sum_{t=1}^{N}\left(r^{(t)} - \boldsymbol{w}^\top \boldsymbol{x}^{(t)}\right)^2 + \lambda\boldsymbol{w}^\top\boldsymbol{w}$, where $\lambda = \alpha/\beta$
    - In addition to minimizing the SSE, we ***regularize*** the norm of $\boldsymbol{w}$ to prevent a highly complex model, thereby reducing the generalization error
    - $y' = \arg_y \max p(y|\boldsymbol{x}'; \boldsymbol{w}_{MAP}) = \arg_y \max \mathcal{N}(y|\boldsymbol{w}_{MAP}^\top\boldsymbol{x}', \beta^{-1}) = \boldsymbol{w}_{MAP}^\top\boldsymbol{x}'$

Bayesian Estimator for r'

- The above methods thread θ as a deterministic value when making predictions
- Another technique, called the ***Bayesian estimation*** of r', treats θ as a random variable, and considers all possible values of θ when estimating r':
  - $y' = \arg_y \max p(y|\boldsymbol{x}', \mathcal{X}) = \int p(y, \theta|\boldsymbol{x}', \mathcal{X}) d\theta$
  - E.g., in linear regression, $y' = \arg_y \max p(y|\boldsymbol{x}', \mathcal{X}) = \arg_y \max \int p(y, \boldsymbol{w}|\boldsymbol{x}', \mathcal{X}) d\boldsymbol{w}$
  - No separated estimation phase for θ
- We will discuss how to solve y' in the lecture of graphical models

Regression Revisited (1)

- Given $\mathcal{X} = \{\boldsymbol{x}^{(t)}, r^{(t)}\}_{t=1}^{N}$, where $r^{(t)} \in \mathbb{R}$. Assume
  - $(\boldsymbol{x}^{(t)}, r^{(t)})$ are i.i.d samples drawn from some joint distribution of $\boldsymbol{x}$ and r (otherwise can never learn r from $\boldsymbol{x}$)
  - In particular, $r^{(t)} = f(\boldsymbol{x}^{(t)}; \theta) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \beta^{-1})$ for some ***hyperparameter*** (i.e., constant fixed during the objective solving) β
  - The marginal distribution $p(r|\boldsymbol{x})$ follows: $p(r|\boldsymbol{x}) = p_{\mathcal{N}_{h(\boldsymbol{x};\theta),\beta^{-1}}}(r)$
- We want to estimate f using $\mathcal{X}$
  - Hypothesis: $h(\boldsymbol{x}; w_0, w_1, \cdots, w_d) = w_0 + w_1 x_1 + \cdots + w_d x_d$, a line
  - Once getting $w_0, w_1, \cdots, w_d$, we can predict the unknown r' of a new instance $\boldsymbol{x}'$ by
    $y' = \arg_y \max p(y|\boldsymbol{x}') = \arg_y \max p_{\mathcal{N}_{h(\boldsymbol{x}';\theta),\beta^{-1}}}(y) = h(\boldsymbol{x}'; \theta)$
  - Note that we don't need to know β to make prediction

Regression Revisited (2)

- How to obtain the estimate h of f? How to obtain θ?
- We can pick θ maximizing $p(\theta|\mathcal{X})$, the ***posterior*** probability
- Or, by Baye's theorem, θ maximizing the ***likelihood*** $p(\mathcal{X}|\theta)$ (if we assume $p(\theta)$ remains the same for all θ)
- Or, θ maximizing the ***log likelihood*** $\log p(\mathcal{X}|\theta) = \log\left(\prod_{t=1}^{N} p(\boldsymbol{x}^{(t)}, r^{(t)}|\theta)\right) = \log\left(\prod_{t=1}^{N} p(r^{(t)}|\boldsymbol{x}^{(t)}, \theta)p(\boldsymbol{x}^{(t)}|\theta)\right) = \log\left(\prod_{t=1}^{N} p(h(\boldsymbol{x}^{(t)};\theta)+\epsilon|\boldsymbol{x}^{(t)}, \theta)p(\boldsymbol{x}^{(t)}|\theta)\right)$
- Ignoring $p(\boldsymbol{x}^{(t)}|\theta) = p(\boldsymbol{x}^{(t)})$ (since it is irrelevant to θ) and constants we have $\log p(\mathcal{X}|\theta) \propto -N\log\left(\sqrt{\frac{2\pi}{\beta}}\right) - \frac{\beta}{2}\sum_{t=1}^{N}\left(r^{(t)} - h(\boldsymbol{x}^{(t)};\theta)\right)^2$
- Dropping the first term and constants we have $\log p(\mathcal{X}|\theta) \propto -\sum_{t=1}^{N}\left(r^{(t)} - h(\boldsymbol{x}^{(t)};\theta)\right)^2$; that is, we seek for θ minimizing the SSE (sum of square errors)

The Bias/Variance Dilemma (1/4)

- The likelihood-based classification and regression share the same idea that the estimators $h(x;\theta_{\mathcal{X}})$ are obtained by $\theta_{\mathcal{X}} = \arg_\theta \max p(\mathcal{X}|\theta)$
  - In classification, $h(x;\theta_{\mathcal{X}})$ estimates the discriminant of a class; in regression, $h(x;\theta_{\mathcal{X}})$ estimates f
- Given a new instance x' where r' is unknown, the expected square error (over the joint distribution of (x,r)) of our prediction can be written as

$$
\begin{aligned}
E[(r-h(x';\theta_{\mathcal{X}}))^2|x'] &= \int (r-h(x';\theta_{\mathcal{X}}))^2 p(r|x')dr \\
&= \int \left[(r-E[r|x']) + (E[r|x'] - h(x';\theta_{\mathcal{X}})\right]^2 p(r|x')dr \\
&= \int (r-E[r|x'])^2 p(r|x')dr + (E[r|x'] - h(x';\theta_{\mathcal{X}}))^2 \int p(r|x')dr - 2\cdot 0 \\
&= E[(r-E[r|x'])^2|x'] + (E[r|x'] - h(x';\theta_{\mathcal{X}}))^2
\end{aligned}
$$

- The first term does not depend on h but the assumption of the joint distribution of (x,r)

The Bias/Variance Dilemma (2/4)

- The second term changes as we vary our hypothesis h and its complexity
- Note that in regression, $E[r|x'] = E[f(x') + \epsilon|x'] = f(x') + E[\epsilon|x'] = f(x')$ so the second term measures how our estimator h is difference from its target f
  - The similar argument applies to the case of classification
- Recall that we can measure how good the estimator h is by using the mean square error $E_{\mathcal{X}}[(h-f)^2]$ ***over all possible $\mathcal{X}$ of the same size***[1]
- Since h and f are functions, we can rewrite the mean square error as follows given an instance x' :

$$
\begin{aligned}
E_{\mathcal{X}}[(h(x';\theta_{\mathcal{X}}) - E[r|x'])^2|x'] &= bias^2 + variance \\
&= \left(E_{\mathcal{X}}[h(x';\theta_{\mathcal{X}})] - E[r|x']\right)^2 + E_{\mathcal{X}}\left[\left(h(x';\theta_{\mathcal{X}}) - E_{\mathcal{X}}[h(x';\theta_{\mathcal{X}})]\right)^2\right]
\end{aligned}
$$

[1] Here we distinguish $E_{\mathcal{X}}$ (over $\mathcal{X}$) from E (over the joint distribution of (x, r))

The Bias/Variance Dilemma (3/4)

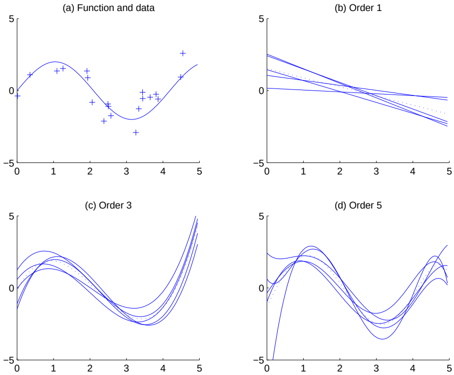


Figure : (a) A function $f(x) = 2\sin(1.5x)$ and a noisy training set ($\epsilon =_{s.t.} N_{0,1}$) consisting of 20 examples. There are totally 5 training sets $\mathcal{X}_i$, $1 \leqslant i \leqslant 5$, generated to calculate $E_{\mathcal{X}}$. (b), (c), and (d) are 5 polynomial fits, namely $h(x;\theta_{\mathcal{X}_i})$ of order 1, 3, and 5 respectively. For each case, the dotted line shows the average of the 5 fits, namely $E_{\mathcal{X}_i}\left[h(x;\theta_{\mathcal{X}_i})\right]$.

The Bias/Variance Dilemma (4/4)

- As we can see, a complex (i.e., high order) hypothesis h has
  - Low bias, as the average of the 5 fits is close to f
  - But high variance, as its shape is affected by noise
  - The variance decreases as N increase, since when N is large the different training sets $\mathcal{X}_i$ look similar
- This is a mathematical way to justify: generalization error $\propto$ empirical error + (model complexity / N)
  - Empirical error corresponds to the bias
  - The second term corresponds to the variance

Model Selection

- The right order of h can be determined using the cross validation technique
- Given the validation results at right (the dotted line), which order should we take?

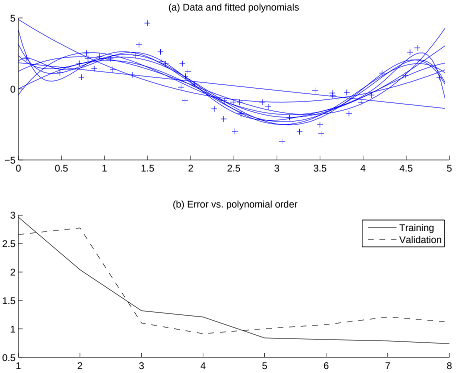


Figure : Cross validation results of 8 hypotheses with orders 1 to 8. Both the training and cross validation sets contain 50 instances.

Model Selection

- The right order of h can be determined using the cross validation technique
- Given the validation results at right (the dotted line), which order should we take?3
  - Why not 4? Occam's razor tells us that we should choose the simplest hypothesis provided that its error is comparable
  - Note the validation results may not be as V-shaped as we might expect when N is large

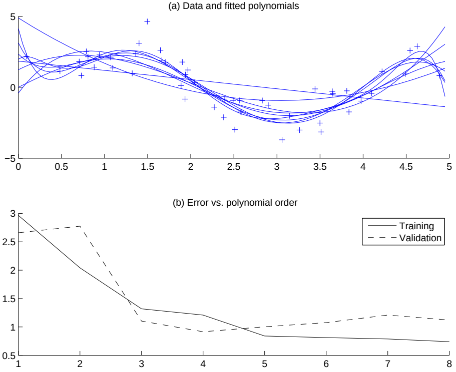


Figure : Cross validation results of 8 hypotheses with orders 1 to 8. Both the training and cross validation sets contain 50 instances.

Outline

Univariate Classification

- Given a training set $\mathcal{X} = \{x^{(t)}, \boldsymbol{r}^{(t)}\}_{t=1}^{N}$, where $r_i^{(t)} = 1$ if $x^{(t)} \in C_i$ and 0 otherwise, we find the discriminant $f_i(x) = P(C_i|x)$ for each class C_i, and then classify a new instance x' as $C_{y'}$ if $y' = \arg_i \max P(C_i|x)$
- Based on the generative assumption and Bayes' rule, we pick C_i such that $f_i(x') = \log\left(p(x'|C_i)P(C_i)\right) = \log p(x'|C_i) + \log P(C_i)$ is maximized
- To be able to make prediction given all possible x'
  - We estimate the prior $P(C_i)$ by $\widehat{P}[C_i] = \frac{\sum_{t=1}^{N} r_i^{(t)}}{N}$
  - By assuming that instances of the same class are normally distributed, we estimate the likelihood $p(x|C_i)$ by $\widehat{p}(x|C_i) = \frac{1}{\sqrt{2\pi s_i^2}} exp\left(\frac{-(x-m_i)^2}{2s_i^2}\right)$, where $m_i = \frac{\sum_{t=1}^{N} x^{(t)} r_i^{(t)}}{\sum_{t=1}^{N} r_i^{(t)}}$ and $s_i^2 = \frac{\sum_{t=1}^{N} (x^{(t)} - m_i)^2 r_i^{(t)}}{\sum_{t=1}^{N} r_i^{(t)} - 1}$

Outline

Maximum Likelihood Estimation

- Why $\widehat{P}(C_i) = \frac{\sum_{t=1}^{N} r_i^{(t)}}{N}$ and $\widehat{p}(x|C_i) = \frac{1}{\sqrt{2\pi s_i^2}} exp\left(\frac{-(x-m_i)^2}{2s_i^2}\right)$ are good choices?

Maximum Likelihood Estimation

- Why $\widehat{P}(C_i) = \frac{\sum_{t=1}^{N} r_i^{(t)}}{N}$ and $\widehat{p}(x|C_i) = \frac{1}{\sqrt{2\pi s_i^2}} exp\left(\frac{-(x-m_i)^2}{2s_i^2}\right)$ are good choices?
  - It turns out that each of these estimators maximizes the likelihood $p(\mathcal{X}|\theta)$, where θ is the parameters of the distribution used to model the target probability ($P(C_i)$ and $p(x|C_i)$ respectively)
- When we talk about the likelihood-based classification, ***the "likelihood" actually refers to the one ($p(\mathcal{X}|\theta)$) of θ given $\mathcal{X}$ rather than that ($p(\boldsymbol{x}'|C_i)$) of C_i given x'***

ML Estimation for $P(C_i)$ (1/2)

- To estimate $P(C_i)$, we first assume that $P(C_i)$ has the Bernoulli distribution parametrized by $\theta = \rho_i$ and can be written as $P(C_i) = P(C_i; \theta)$
  - Let X_i be a random variable where $X_i = 1$ if the event "the outcome of a toss is C_i and $X_i = 0$ if "the outcome is not C_i"
  - Let ρ_i be the probability that $X_i = 1$, we have $P(X_i = c; \theta) = \rho_i^c (1 - \rho_i)^{1-c}$, $c \in \{0, 1\}$
- Now the problem estimating $P(C_i|\theta) = P(X_i = 1; \theta) = \rho_i$ can be reduced to estimating $\theta = \rho_i$

ML Estimation for $P(C_i)$ (2/2)

- Given the training set $\mathcal{X}$, a good estimate of θ is the one that maximizes $P(\theta|\mathcal{X})$
  - From Bayes' rule, we can instead pick $\widehat{\theta}$ maximizing $P(\mathcal{X}|\theta)$ if we don't have prior reason to favor certain θ
  - Equivalently, we pick $\widehat{\theta}$ maximizing $\log P(\mathcal{X}|\theta)$
  - We have $\log P(\mathcal{X}|\theta) = \log\left(\prod_{t=1}^{N} \rho_i^{r_i^{(t)}} (1-\rho_i)^{1-r_i^{(t)}}\right)$
  - Solving $\frac{d(\log P(\mathcal{X}|\theta))}{d\rho} = 0$ we obtain the ***Maximum Likelihood (ML) estimator*** $\widehat{\rho}_i = \frac{\sum_{t=1}^{N} r_i^{(t)}}{N}$ [Proof]
- $\widehat{P}[C_i] = P(C_i|\widehat{\theta}) = \widehat{\rho}_i$
- Note we can also consider all classes together and assume that $P(C_i)$ follows the Multinomial distribution parametrized by $\theta = (\rho_1, \cdots, \rho_K)$ with constrains $\sum_{i=1}^{K} \rho_i = 1$
  - The ML estimator for each ρ_i will be the same as the above [Homework]

ML Estimation for $p(x|C_i)$ (1/2)

- We assume that $p(x|C_i)$ is normal and can be written as $p(x|C_i) = p(x|C_i;\theta)$ with some $\theta = (\mu_i, \sigma_i)$
  - $p(x|C_i;\theta) = p_{N_{\mu_i,\sigma^2}}(x) = \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left(\frac{-(x-\mu_i)^2}{2\sigma_i^2}\right)$
- Now the problem estimating $p(x|C_i;\theta)$ cane be reduced to estimating $\theta = (\mu_i, \sigma_i)$
- Given the training set $\mathcal{X}$, a good estimate of θ is the one that maximizes $\log p(\mathcal{X}|\theta)$
  - We have $\log p(\mathcal{X}|\theta) = \log\left(\prod_{t=1}^{N}\left(\frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left(\frac{-(x^{(t)}-\mu_i)^2}{2\sigma_i^2}\right)\right)^{r_i^{(t)}}\right)$
  - Taking the partial derivatives of $\log p(\mathcal{X}|\theta)$ in terms of μ_i and σ_i and setting them equal to 0 we obtain the estimators $m_i = \frac{\sum_{t=1}^{N} x^{(t)} r_i^{(t)}}{\sum_{t=1}^{N} r_i^{(t)}}$ and $s_i^2 = \frac{\sum_{t=1}^{N}(x^{(t)}-m_i)^2 r_i^{(t)}}{\sum_{t=1}^{N} r_i^{(t)}}$ respectively [Proof]

ML Estimation for $p(x|C_i)$ (2/2)

- Recall that $s_i^2 = \frac{\sum_{t=1}^{N}(x^{(t)}-m_i)^2 r_i^{(t)}}{\sum_{t=1}^{N} r_i^{(t)}}$ is a bias estimator, we can replace the denominator with $\sum_{t=1}^{N} r_i^{(t)} - 1$
  - This step is optional
  - The difference, actually, is negligible when N is large
- $\widehat{p}(x|C_i) = p(x|C_i, \widehat{\theta}) = \frac{1}{\sqrt{2\pi s_i^2}} exp\left(\frac{-(x-m_i)^2}{2s_i^2}\right)$

Outline

Multivariate Data

- Let's go back to a higher dimensional feature space
  - We are given a training set $\mathcal{X} = \{\boldsymbol{x}^{(t)}, \boldsymbol{r}^{(t)}\}_{t=1}^{N}$ where $\boldsymbol{x}^{(t)} \in \mathbb{R}^d$ and $(\boldsymbol{x}^{(t)}, \boldsymbol{r}^{(t)})$ are i.i.d. samples drawn from some unknown (multivariate) distribution
  - Typically, the features of $\boldsymbol{x}^{(t)}$ are correlated (otherwise we can discuss each attribute individually using the univariate methods)
- It might be a good idea to review the multivariate distributions now

Multivariate Classification

- The idea remains the same: given a new instance $\boldsymbol{x}' \in \mathbb{R}^d$, we make prediction by picking the class C_i if its discriminant $f_i(\boldsymbol{x}') = P(C_i|\boldsymbol{x}')$ is maximized
  - Generative assumption: pick C_i if $f_i(\boldsymbol{x}') = \log p(\boldsymbol{x}'|C_i) + \log P(C_i)$ is maximized
- It's common to assume that $p(\boldsymbol{x}|C_i)$ follows the multivariate normal distribution, i.e.,
  $p(\boldsymbol{x}|C_i) = p_{\mathbf{N}_{\mu_i,\Sigma_i}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2} det(\boldsymbol{\Sigma}_i)^{1/2}} exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right]$
- Why?
  - Major reason: analytical simplicity
  - Studies also show that the model is robust to datasets departing from normality

Maximum Likelihood Estimation

- The ML estimators of $P(C_i)$ is $\widehat{P}[C_i] = \sum_{t=1}^{N} r_i^{(t)} / N$
  - We have seen this in the univariate cases before
- The ML estimators of $p(\boldsymbol{x}|C_i)$ is $\frac{1}{(2\pi)^{d/2} det(\boldsymbol{S}_i)^{1/2}} exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^\top \boldsymbol{S}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right]$, where $\boldsymbol{m}_i = \frac{\sum_{t=1}^{N} \boldsymbol{x}^{(t)} r_i^{(t)}}{\sum_{t=1}^{N} r_i^{(t)}}$ and $\boldsymbol{S}_i = \frac{\sum_{t=1}^{N} (\boldsymbol{x}^{(t)}-\boldsymbol{m}_i)(\boldsymbol{x}^{(t)}-\boldsymbol{m}_i)^\top r_i^{(t)}}{\sum_{t=1}^{N} r_i^{(t)}}$
- Why?

Maximum Likelihood Estimation

- The ML estimators of $P(C_i)$ is $\widehat{P}[C_i] = \sum_{t=1}^{N} r_i^{(t)}/N$
  - We have seen this in the univariate cases before
- The ML estimators of $p(\boldsymbol{x}|C_i)$ is $\frac{1}{(2\pi)^{d/2} det(\boldsymbol{S}_i)^{1/2}} exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^\top \boldsymbol{S}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right]$, where
  $\boldsymbol{m}_i = \frac{\sum_{t=1}^{N} \boldsymbol{x}^{(t)} r_i^{(t)}}{\sum_{t=1}^{N} r_i^{(t)}}$ and
  $\boldsymbol{S}_i = \frac{\sum_{t=1}^{N} (\boldsymbol{x}^{(t)}-\boldsymbol{m}_i)(\boldsymbol{x}^{(t)}-\boldsymbol{m}_i)^\top r_i^{(t)}}{\sum_{t=1}^{N} r_i^{(t)}}$
- Why? It's a good idea to review the matrix calculus now

ML Estimator of μ_i

- Let $\theta = (\mu_i, \Sigma_i)$, we have the likelihood $\log p(\mathcal{X}|\theta) = \log\left(\prod_{t=1}^{N}\left(\frac{1}{(2\pi)^{d/2}det(\Sigma_i)^{1/2}}e^{-\frac{1}{2}(\boldsymbol{x}^{(t)}-\mu_i)^\top\Sigma_i^{-1}(\boldsymbol{x}^{(t)}-\mu_i)}\right)^{r_i^{(t)}}\right) = -\frac{N_i d}{2}\log(2\pi) - \frac{N_i}{2}\log(det(\Sigma_i)) - \frac{1}{2}\sum_{t=1}^{N} r_i^{(t)}(\boldsymbol{x}^{(t)}-\mu_i)^\top\Sigma_i^{-1}(\boldsymbol{x}^{(t)}-\mu_i)$, where $N_i = \sum_{t=1}^{N} r_i^{(t)}$
- Recall that for any $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{A} \in \mathbb{R}^{n\times n}$,
  - $\frac{\partial}{\partial x}(\boldsymbol{a}^\top\boldsymbol{x}) = \frac{\partial}{\partial x}(\boldsymbol{x}^\top\boldsymbol{a}) = \boldsymbol{a}^\top$
  - $\frac{\partial}{\partial x}(\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}) = \boldsymbol{x}^\top(\boldsymbol{A}+\boldsymbol{A}^\top)$
- Taking the partial derivative of $\log p(\mathcal{X}|\theta)$ with respect to μ_i and setting it to zero, we get $\sum_{t=1}^{N} r_i^{(t)}(\boldsymbol{x}^{(t)}-\mu_i)^\top\Sigma_i^{-1} = \boldsymbol{0}^\top$ [Proof]
- So $\boldsymbol{m}_i = \frac{\sum_{t=1}^{N}\boldsymbol{x}^{(t)}r_i^{(t)}}{\sum_{t=1}^{N} r_i^{(t)}}$

ML Estimator of Σ_i (1/2)

- $\log p(\mathcal{X}|\theta) = -\frac{N_i d}{2}\log(2\pi) - \frac{N_i}{2}\log(det(\boldsymbol{\Sigma}_i)) - \frac{1}{2}\sum_{t=1}^{N} r_i^{(t)}(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)$
- Note $\log(det(\boldsymbol{\Sigma}_i^{-1})) = -\log(det(\boldsymbol{\Sigma}_i))$
- Also, $(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i) = tr\left(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)^\top\right)$ [Proof]
- We can rewrite the likelihood as $\log p(\mathcal{X}|\theta) = -\frac{N_i d}{2}\log(2\pi) + \frac{N_i}{2}\log(det(\boldsymbol{\Sigma}_i^{-1})) - \frac{1}{2}\sum_{t=1}^{N} r_i^{(t)} tr\left(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)^\top\right)$

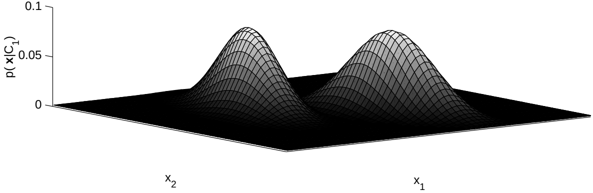
ML Estimator of Σ_i (2/2)

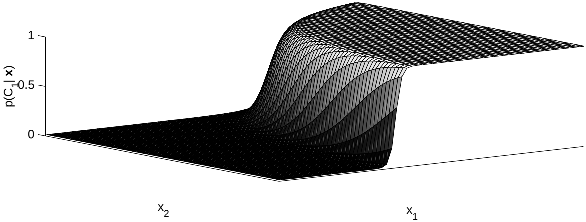
- Given any function $f(x)$, let $g(x) = f(\frac{1}{x})$ for any $x > 0$, then x^* is a stationary point of g iff $\frac{1}{x^*}$ is a stationary point of f
  - The matrix version $g(\boldsymbol{A}) = f(\boldsymbol{A}^{-1})$ applies when $\boldsymbol{A}$ is positive definite
- We can seek for the partial derivative of $\log p(\mathcal{X}|\theta)$ with respect to $\boldsymbol{\Sigma}_i^{-1}$
- Recall that $\frac{\partial}{\partial \boldsymbol{A}} \ln(det(\boldsymbol{A})) = (\boldsymbol{A}^{-1})^\top$, and $\frac{\partial}{\partial \boldsymbol{A}} tr(\boldsymbol{AB}) = \boldsymbol{B}^\top$
- Taking the partial derivative of $\log p(\mathcal{X}|\theta)$ with respect to $\boldsymbol{\Sigma}_i^{-1}$ and setting it to zero, we get $\frac{N_i}{2}\boldsymbol{\Sigma}_i - \frac{1}{2}\sum_{t=1}^N r_i^{(t)}(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)^\top = \boldsymbol{O}$ [Proof]
- Therefore, $\boldsymbol{S}_i = \frac{\sum_{t=1}^N (\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)(\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)^\top r_i^{(t)}}{\sum_{t=1}^N r_i^{(t)}}$

Quadratic Discrimination

- Ignoring the constant terms we have the discriminant $f_i(\boldsymbol{x}) = -\frac{1}{2}\log(det(\boldsymbol{S}_i)) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^\top \boldsymbol{S}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i) + \log \widehat{P}[C_i]$, which can be rewritten as
  $f_i(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{W}_i \boldsymbol{x} + \boldsymbol{w}_i^\top \boldsymbol{x} + w_i$, where
  $\boldsymbol{W}_i = -\frac{1}{2}\boldsymbol{S}_i^{-1}$,
  $\boldsymbol{w}_i = \boldsymbol{S}_i^{-1}\boldsymbol{m}_i$, and
  $w_i = -\frac{1}{2}\boldsymbol{m}_i^\top \boldsymbol{S}_i^{-1}\boldsymbol{m}_i - \frac{1}{2}\log(det(\boldsymbol{S}_i)) + \log \widehat{P}[C_i]$ [Proof]
  - The classification is done via ***quadratic discrimination***
  - The decision boundary between any two classes is quadratic too [Proof]

Multivariate Classification (3/3)





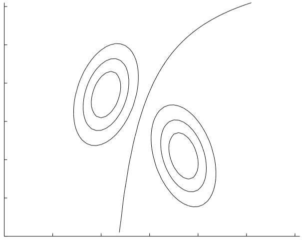
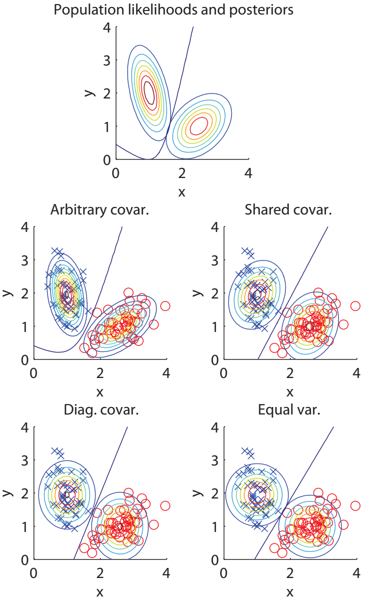


Figure : (a) The graphs of $p(\boldsymbol{x}|C_i)$ for two classes with different covariance matrices. (b) The graph of posterior $P(C_1|\boldsymbol{x})$. (c) Level sets of $p(\boldsymbol{x}|C_i)$ and the decision boundary.

Outline

Simplifications (1/2)

- Quadratic discrimination:
  - Attributes in different classes have different covariance matrices $\boldsymbol{S}_i$ ($\{\boldsymbol{x} : p(\boldsymbol{x}|C_i) = c\}$ are ellipsoids)
- ***Linear discrimination***:
  - Attributes in different classes share the same correlation $\boldsymbol{S}_i = \boldsymbol{S}$ (ellipsoids with the same shape/orientation)
  - Attributes in each classes are independent $\boldsymbol{S}_i = \boldsymbol{S} = \boldsymbol{D}$ (axis-aligned ellipsoids with the same shape/orientation)
  - Attributes in each classes has the same variance $\boldsymbol{S}_i = \boldsymbol{S} = s^2\boldsymbol{I}$ (equal-sized spheres)



Simplifications (2/2)

- Linear discrimination models seem to be oversimplified, but why are they popular in real applications?

Simplifications (2/2)

- Linear discrimination models seem to be oversimplified, but why are they popular in real applications?
- Quadratic discrimination has lower bias, but higher variance
- Experience tells us that ***when we have a small dataset***, it may be better to assume a shared and simplified covariance matrix
  - $\boldsymbol{S}_i = \boldsymbol{S}$ can be estimated using all examples in a dataset together
  - $\boldsymbol{S} = \boldsymbol{D}$ if we do not have enough data to estimate the covariance between attributes accurately
  - $\boldsymbol{D} = s^2 \boldsymbol{I}$ if attributes are z-normalized
- Linear discrimination is ***not*** necessarily linear
  - We can augment the inputs (e.g., $x_{d+1} = exp(x_1 + x_4)$) to build a higher dimensional feature space, if we believe this is useful
  - Linear discrimination in the augmented feature space corresponds to a nonlinear model in the original input space
- We can perform the cross validation to decide which assumption is the best

Linear Discrimination ($S_i = S$)

- The discriminant for each class is $f_i(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{W}_i \boldsymbol{x} + \boldsymbol{w}_i^\top \boldsymbol{x} + w_i$, where
  $\boldsymbol{W}_i = -\frac{1}{2}\boldsymbol{S}_i^{-1}$,
  $\boldsymbol{w}_i = \boldsymbol{S}_i^{-1}\boldsymbol{m}_i$, and
  $w_i = -\frac{1}{2}\boldsymbol{m}_i^\top \boldsymbol{S}_i^{-1}\boldsymbol{m}_i - \frac{1}{2}\log(det(\boldsymbol{S}_i)) + \log\widehat{P}[C_i]$
- We can replace $\boldsymbol{S}_i$ with $\boldsymbol{S}$, the estimator of $\boldsymbol{\Sigma}$ of all instances in the training set
  - The level sets $\{\boldsymbol{x} : p(\boldsymbol{x}|C_i) = c\}$ are ellipsoids with the same shape/orientation
- Ignoring the constant terms, the discriminant now becomes
  $f_i(\boldsymbol{x}) = \boldsymbol{w}_i^\top \boldsymbol{x} + w_i$, where
  $\boldsymbol{w}_i = \boldsymbol{S}^{-1}\boldsymbol{m}_i$ and
  $w_i = -\frac{1}{2}\boldsymbol{m}_i^\top \boldsymbol{S}^{-1}\boldsymbol{m}_i + \log\widehat{P}[C_i]$ [Proof]

Naive Bayes Classifiers ($S_i = S = D$)

- We can further assume that attributes are independent with each other, i.e., $\boldsymbol{S}_i = \boldsymbol{S} = \begin{bmatrix} s_0^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_d^2 \end{bmatrix}$ are diagonal
  - Likelihood-based classifiers using this strong (naive) independence assumption are called the ***naive Bayes' classifiers***
  - The level sets $\{\boldsymbol{x} : p(\boldsymbol{x}|C_i) = c\}$ are axis-aligned ellipsoids
  - $f_i(\boldsymbol{x}) = -\frac{1}{2}\sum_{j=1}^{d}\left(\frac{m_{i,j}^2 - 2x_j m_{i,j}}{s_j^2}\right) + \log \widehat{P}[C_i]$ [Proof]
- If we further assume that that attributes have the same variance, i.e., $\boldsymbol{S}_i = \boldsymbol{S} = s\boldsymbol{I}$
  - The level sets $\{\boldsymbol{x} : p(\boldsymbol{x}|C_i) = c\}$ degenerate into spheres
  - $f_i(\boldsymbol{x}) = -\frac{1}{2s^2}\left(\|\boldsymbol{m}_i\|^2 - 2\boldsymbol{m}_i^\top \boldsymbol{x}\right) + \log \widehat{P}[C_i]$ (or $f_i(\boldsymbol{x}) = -\frac{1}{2s^2}\|\boldsymbol{x} - \boldsymbol{m}_i\|^2 + \log \widehat{P}[C_i]$) [Proof]
  - If we drop $\log \widehat{P}[C_i]$, we obtain a ***nearest mean classifier***