CpE 646 Pattern Recognition and Classification

Prof. Hong Man

Department of Electrical and
Computer Engineering
Stevens Institute of Technology

STEVENS
Institute of Technology

Visual Information Environment Laboratory

Bayesian Decision Theory

Chapter 3 (Section 3.1 – 3.5) Outline:

- Introduction
- Maximum-likelihood Estimation
- Bayesian Estimation
- Bayesian Parameter Estimation: Gaussian Case
- Bayesian Parameter Estimation: General Theory

Introduction

- Data availability in a Bayesian framework
  - We could design an optimal classifier if we knew:
    - $P(\omega_i)$ (priors)
    - $p(x|\omega_i)$ (class-conditional densities)
  - Unfortunately, we rarely have this complete information. What we usually have are
    - some knowledge of the probabilities densities
    - some training data sample

Introduction

- Supervised learning and unsupervised learning
  - In both cases, the observed sample x reflects the prior probabilities $P(\omega_i)$ and the class- conditional densities $p(x|\omega_i)$ independently.
  - Supervised learning assumes that the training sample x is individually labeled with true state of nature, whereas the unsupervised learning only has unlabeled sample.

Introduction

- Design a classifier from a training sample
  - Estimation of prior probabilities is usually not difficult
  - Estimation of class-conditional densities is difficult
    - Samples are often too small and feature dimension maybe too high
    - If we have some knowledge about the density, e.g. certain model with set of parameters, this estimation can be much simplified.
    - For example Normal density $p(x|\omega_i) \sim N(\mu_i, \Sigma_i)$ can be characterized by 2 parameters
    - The density estimation becomes parameter estimation – parametric methods

Introduction

- Parameter estimation techniques
  - Maximum-likelihood (ML) estimation and Bayesian estimation
  - Results are nearly identical, but the approaches are conceptually different
    - In ML methods, parameters are quantities with fixed unknown values. The best estimated value will maximize the probability of obtaining the samples observed.
    - In Bayesian methods, parameters are random variables with known prior distribution, and observations will convert this to a posterior density.

Introduction

- In Bayesian estimation, additional observation sample will sharpen the a posteriori density function, causing it to peak near the true values of the parameters. This phenomenon is called Bayesian learning.

Maximum-Likelihood Estimation

- Maximum-Likelihood Estimation
  - Has good convergence properties as the sample size increases
  - Simpler than any other alternative techniques
- General principle
  - Assume we have ***c*** classes and
  - We have ***c*** data sets, D_1, D_2, …, D_c
  - D_j contains samples independently drawn from $p(x|\omega_j)$, such samples are called independent and identically distributed random variables, i.i.d.

Maximum-Likelihood Estimation

- We assume $p(x|\omega_j)$ has known parametric form and can be uniquely determined by parameters θ_j. For example $p(x|\omega_j) \sim N(\mu_j,\Sigma_j)$
- Let $p(x|\omega_j) \equiv p(x|\omega_j,\theta_j)$. In Normal density case

$$\boldsymbol{\theta_j = (\mu_j, \Sigma_j) = (\mu_j^1, \mu_j^2, \ldots, \sigma_j^{11}, \sigma_j^{22}, \mathrm{cov}(x_j^m, x_j^n), \ldots)}$$

- The objective is to use training samples to estimate such parameter vector for each class, θ_1, θ_2, …, θ_c
- We assume parameters for different classes are independent, so each θ_j can be estimated independently.

Maximum-Likelihood Estimation

- Suppose that D contains n samples, $\{x_1, x_2, \ldots, x_n\}$, because the samples are drawn independently

$$p(D \mid \theta) = \prod_{k=1}^{n} p(x_k \mid \theta)$$

- $p(D|\theta)$ is called the likelihood of θ w.r.t. the set samples
- The ML estimate of θ is, by definition the value $\hat{\theta}$ that maximizes $p(D|\theta)$
  - It is the value of θ that best agrees with the actually observed training sample
- It is usually easier to work with the logarithm of the likelihood – log-likelihood.

Maximum-Likelihood Estimation

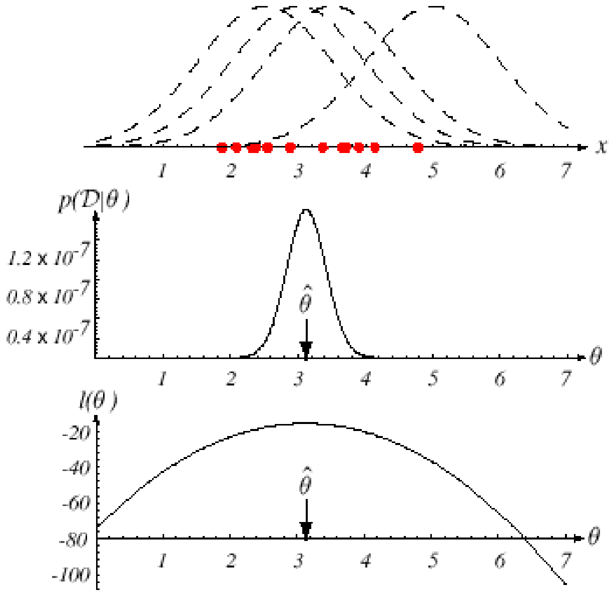


FIGURE 3.1. The top graph shows several training points in one dimension, known or assumed to be drawn from a Gaussian of a particular variance, but unknown mean. Four of the infinite number of candidate source distributions are shown in dashed lines. The middle figure shows the likelihood $p(\mathcal{D}|\theta)$ as a function of the mean. If we had a very large number of training points, this likelihood would be very narrow. The value that maximizes the likelihood is marked $\hat{\theta}$; it also maximizes the logarithm of the likelihood—that is, the log-likelihood $l(\theta)$, shown at the bottom. Note that even though they look similar, the likelihood $p(\mathcal{D}|\theta)$ is shown as a function of θ whereas the conditional density $p(x|\theta)$ is shown as a function of x. Furthermore, as a function of θ, the likelihood $p(\mathcal{D}|\theta)$ is not a probability density function and its area has no significance. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Maximum-Likelihood Estimation

- Optimal estimation
  - Let $\theta = (\theta_1, \theta_2, \ldots, \theta_p)^t$, i.e. θ has p parameters, and let ∇_θ be the gradient operator

$$\nabla_\theta \equiv \left[\frac{\partial}{\partial \theta_1}, \frac{\partial}{\partial \theta_2}, \ldots, \frac{\partial}{\partial \theta_p} \right]^t$$

  - We define $l(\theta)$ as the log-likelihood function

$$l(\theta) = \ln p(D|\theta)$$

  - New problem statement:

    determine θ that maximizes the log-likelihood

$$\hat{\theta} = \arg\max_{\theta} l(\theta)$$

Maximum-Likelihood Estimation

Thus

$$l(\theta) = \ln p(D \mid \theta) = \ln\left(\prod_{k=1}^{n} p(x_k \mid \theta)\right) = \sum_{k=1}^{n} \ln p(x_k \mid \theta)$$

$$\nabla_\theta l = \sum_{k=1}^{n} \nabla_\theta \ln p(x_k \mid \theta)$$

the necessary condition for an optimum is

$$\nabla_\theta l = 0$$

- The solution to this equation $\hat{\theta}$ may represent a true global maximum, a local maximum or minimum, or an inflection point of $l(\theta)$

Maximum *A Posteriori*

- A related estimator is maximum *a posteriori* (MAP) estimator
  - It finds the value of θ that maximizes $\boldsymbol{l(\theta)p(\theta)}$**,** where $\boldsymbol{p(\theta)}$ is the prior probability of different parameter values.
  - The MAP estimator becomes the ML estimator for the uniform prior.
  - The drawback of MAP estimator is that if the parameter space is transformed, the densities will change and the MAP solution will no longer hold. (For example the signal is normalized so the mean values is shifted.)

A Gaussian Case

- A Gaussian case with unknown μ
  - $p(x_i|\mu) \sim N(\mu, \Sigma)$: samples are drawn from a multivariate normal population
  - Let $\theta = \mu$,

$$\ln p(\mathbf{x}_k \mid \boldsymbol{\mu}) = -\frac{1}{2}\ln\left[(2\pi)^d \left|\Sigma\right|\right] - \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^t \Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

$$\textbf{and} \quad \nabla_{\mu} \ln p(\mathbf{x}_k \mid \boldsymbol{\mu}) = \Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

  - The ML estimate for μ must satisfy:

$$\sum_{k=1}^{n} \Sigma^{-1}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = 0$$

A Gaussian Case

- Multiplying by Σ (because $|\Sigma| \neq 0$) and rearranging, we have:

$$\hat{\boldsymbol{\mu}} = \frac{1}{\boldsymbol{n}} \sum_{\boldsymbol{k}=1}^{\boldsymbol{n}} \boldsymbol{x}_{\boldsymbol{k}}$$

  - This is just the arithmetic average of the samples of the training samples – sample mean

- Conclusion:
  - If $p(x_k \mid \omega_j)$ $(j = 1, 2, \ldots, c)$ is supposed to be Gaussian in a d-dimensional feature space; then we can estimate the vector $\theta = (\theta_1, \theta_2, \ldots, \theta_c)^t$ and perform an optimal classification

Another Gaussian Case

- Another Gaussian case with unknown μ and Σ
  - In univariate case $\Sigma \Rightarrow \sigma^2$ and $\theta = (\theta_1, \theta_2) = (\mu, \sigma^2)$

$$\boldsymbol{l} = \ln p(x_k \mid \boldsymbol{\theta}) = -\frac{1}{2}\ln 2\pi\boldsymbol{\theta}_2 - \frac{1}{2\boldsymbol{\theta}_2}(x_k - \boldsymbol{\theta}_1)^2$$

$$\nabla_{\theta} \boldsymbol{l} = \begin{bmatrix} \dfrac{\partial}{\partial \boldsymbol{\theta}_1}(\ln P(x_k \mid \boldsymbol{\theta})) \\ \dfrac{\partial}{\partial \boldsymbol{\theta}_2}(\ln P(x_k \mid \boldsymbol{\theta})) \end{bmatrix} = \begin{bmatrix} \dfrac{1}{\boldsymbol{\theta}_2}(x_k - \boldsymbol{\theta}_1) \\ -\dfrac{1}{2\boldsymbol{\theta}_2} + \dfrac{(x_k - \boldsymbol{\theta}_1)^2}{2\boldsymbol{\theta}_2^2} \end{bmatrix} = 0$$

Another Gaussian Case

- Conditions:

$$\begin{cases} \displaystyle\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2}(x_k - \hat{\theta}_1) = 0 & \textbf{(1)} \\ \displaystyle -\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0 & \textbf{(2)} \end{cases}$$

- Combining (1) and (2):

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k \ , \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu})^2$$

Another Gaussian Case

– In multivariate case

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k$$

$$\hat{\Sigma}^2 = \frac{1}{n}\sum_{k=1}^{n} (x_k - \hat{\mu})(x_k - \hat{\mu})^t$$

- The ML estimate for the mean is the sample mean, and the ML estimate for the covariance matrix is the arithmetic average of the *n* matrices

$$(x_k - \hat{\mu})(x_k - \hat{\mu})^t$$

Bias

- The ML estimate for σ^2 is biased, that is the expected value over all data set of size *n* of the sample variance is not equal to the true variance (imaging *n*=1)

$$E\left[\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right]=E\left[\frac{1}{n}\sum_{i=1}^{n}((x_i-\mu)-(\bar{x}-\mu))^2\right]$$

$$=E\left[\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2-2(\bar{x}-\mu)\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)+(\bar{x}-\mu)^2\right]$$

$$=E\left[\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2-(\bar{x}-\mu)^2\right]=\sigma^2-E\left[(\bar{x}-\mu)^2\right]$$

$$=\sigma^2-\frac{1}{n}\sigma^2=\frac{n-1}{n}\sigma^2<\sigma^2$$

Bias

- An elementary unbiased estimator for *Σ* is:

$$\underbrace{C = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \hat{\mu})(x_k - \hat{\mu})^t}_{\text{Sample covariance matrix}}$$

- If an estimator is unbiased for all distributions, it is called absolutely unbiased.
- If an estimator becomes unbiased only when the number of sample is very large, e.g. (*n*-1)/*n*→1, it is called asymptotically unbiased.
- Biased estimator *v.s.* unbiased estimator
  - We just choose one that gives the better classification performance.

Bayesian Estimation

- Bayesian Estimation (Bayesian learning)
  - In *ML* estimation method θ was supposed to be fixed
  - In Bayesian estimation, θ is a random variable with a certain distribution
  - The training data will allow us to convert the distribution of θ into a posterior probability density $P(\omega_i|x)$
  - Goal: compute $P(\omega_i|x,D)$

Bayesian Estimation

- The original Bayes formula is

$$P(\omega_i | x) = p(x|\omega_i) \cdot P(\omega_i) / p(x)$$

- Given the sample *D*, Bayes formula can be written

$$P(\omega_i \mid x, D) = \frac{p(x \mid \omega_i, D) P(\omega_i \mid D)}{\sum_{j=1}^{c} p(x \mid \omega_j, D) P(\omega_j \mid D)}$$

where

$$p(x) \approx p(x \mid D) = \sum_{j} p(x, \omega_j \mid D) = \sum_{j=1}^{c} p(x \mid \omega_j, D) P(\omega_j \mid D)$$

Bayesian Estimation

- We can assume that
  - ***P* (*ω*$_i$) = *P* (*ω*$_i$|*D*),** i.e. the true prior probabilities are known from the training data,
  - The samples in *D* are labeled, so we have {D_1, D_2, …, D_c} where D_i belongs to ***ω*$_i$**,
  - D_i will have no influence to ***p*(*x*|*ω*$_j$,*D*)** for *i*≠*j*
  - then

$$P(\omega_i \mid x, D) = \frac{p(x \mid \omega_i, D_i) P(\omega_i)}{\sum_{j=1}^{c} p(x \mid \omega_j, D_j) P(\omega_j)}$$

Bayesian Estimation

- General theory:
  - Ultimately we want to estimate $p(x)$, and the best we can achieve is $p(x|D)$
  - We assume $p(x)$ takes known parametric form, i.e. $p(x|\theta)$ function is known, but parameters are unknown
  - We assume some knowledge of $p(\theta)$
  - Observation samples $D=\{x_1, x_2, \ldots, x_n\}$ will convert prior density $p(\theta)$ to posterior density $p(\theta|D)$ which we hope will peak at true value of θ.

Bayesian Estimation

- Given training sample samples $D=\{x_1, x_2, \ldots, x_n\}$, because sample are drawn independently, we have

$$p(D|\theta)=\prod_{k=1}^{n} p(x_k|\theta)$$

- With $p(\theta)$ and $p(D|\theta)$, the posterior density $p(\theta|D)$ is

$$p(\theta|D)=\frac{p(D|\theta)p(\theta)}{\int p(D|\theta)p(\theta)d\theta}=\frac{\prod_{k=1}^{n} p(x_k|\theta)p(\theta)}{\int p(D|\theta)p(\theta)d\theta}$$

note that $\int p(D|\theta)p(\theta)d\theta$ **is a normalization factor independent of** θ

Bayesian Estimation

- Finally to compute *p*(*x*|*D*), where *D* are training samples and *x* are testing samples

$$\begin{aligned} p(x \mid D) &= \int p(x, \theta \mid D) d\theta \\ &= \int p(x \mid \theta, D) p(\theta \mid D) d\theta \\ &= \int p(x \mid \theta) p(\theta \mid D) d\theta \end{aligned}$$

note that $p(x|\theta,D) = p(x|\theta)$ because *x* and *D* are drawn independently.

- This is formal Bayesian solution to this problem.

Bayesian Estimation

- Comments:
  - If $p(\theta|D)$ will peak around some value of $\hat{\boldsymbol{\theta}}$, we will have $\boldsymbol{p(x \mid D) \simeq p(x \mid \hat{\theta})}$, i.e. the result is equivalent to substituting the estimate $\hat{\boldsymbol{\theta}}$ for the true parameter vector θ.
  - If there is no clear best estimate $\hat{\boldsymbol{\theta}}$, the $p(x|D)$ expression represents an average over all possible values of θ. This expression is then usually evaluated through Monte-Carlo simulations.

Gaussian Case

- Gaussian case: to calculate the a posteriori density $p(\theta|D)$ and the desired density $p(x|D)$ for the case where $p(x|\mu) \sim N(\mu,\Sigma)$
- The univariate case where μ is the only unknown parameter: $p(x|\mu) \sim N(\mu,\sigma^2)$
  - Compute $p(\mu|D)$
    - We assume prior density for μ as $p(\mu) \sim N(\mu_0,\sigma_0^2)$ where μ_0, σ_0^2 are known (they represent our knowledge on μ)

Gaussian Case

- Let $D=\{x_1, x_2, \ldots, x_n\}$, we have

$$p(\mu \mid D) = \frac{p(D \mid \mu)\, p(\mu)}{\int p(D \mid \mu)\, p(\mu) d\mu}$$

$$= \alpha \prod_{k=1}^{n} p(x_k \mid \mu)\, p(\mu) \qquad \textbf{(1)}$$

where α represents the normalization factor that is dependent of D but is independent of μ

Gaussian Case

- Also we have

$$p(x_k \mid \mu) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_k - \mu}{\sigma}\right)^2 \right]$$

$$p(\mu) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{1}{2}\left(\frac{\mu - \mu_0}{\sigma_0}\right)^2 \right]$$

Gaussian Case

- Substituting $p(x|\mu)$ and $p(\mu)$ into (1), we find the *a posteriori* density $p(\mu|D)$ remains a normal density

$$p(\mu\,|\,D)=\frac{1}{\sqrt{2\pi}\sigma_n}\exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right]$$

where

$$\begin{cases}\mu_n=\left(\dfrac{n\sigma_0^2}{n\sigma_0^2+\sigma^2}\right)\hat{\mu}_n+\dfrac{\sigma^2}{n\sigma_0^2+\sigma^2}\mu_0\\[2ex] \sigma_n^2=\dfrac{\sigma_0^2\sigma^2}{n\sigma_0^2+\sigma^2}\end{cases}$$

where $\hat{\mu}_n=\dfrac{1}{n}\sum_{k=1}^{n}x_k$

Gaussian Case

- Because $p(\mu|D)$ remains a normal density as number n of training samples increases, $p(\mu|D)$ is said to be a reproducing density and the prior $p(\mu)$ is called a conjugate prior.
- μ_n roughly represents the best guess of μ after observing D with n samples, and σ_n^2 represents our uncertainty about this guess.
- As n→∞, $\sigma_n^2 \rightarrow \sigma^2/n$, i.e. $p(\mu|D)$ will peak around μ_n as a Dirac delta function. This behavior is called Bayesian learning.

Gaussian Case

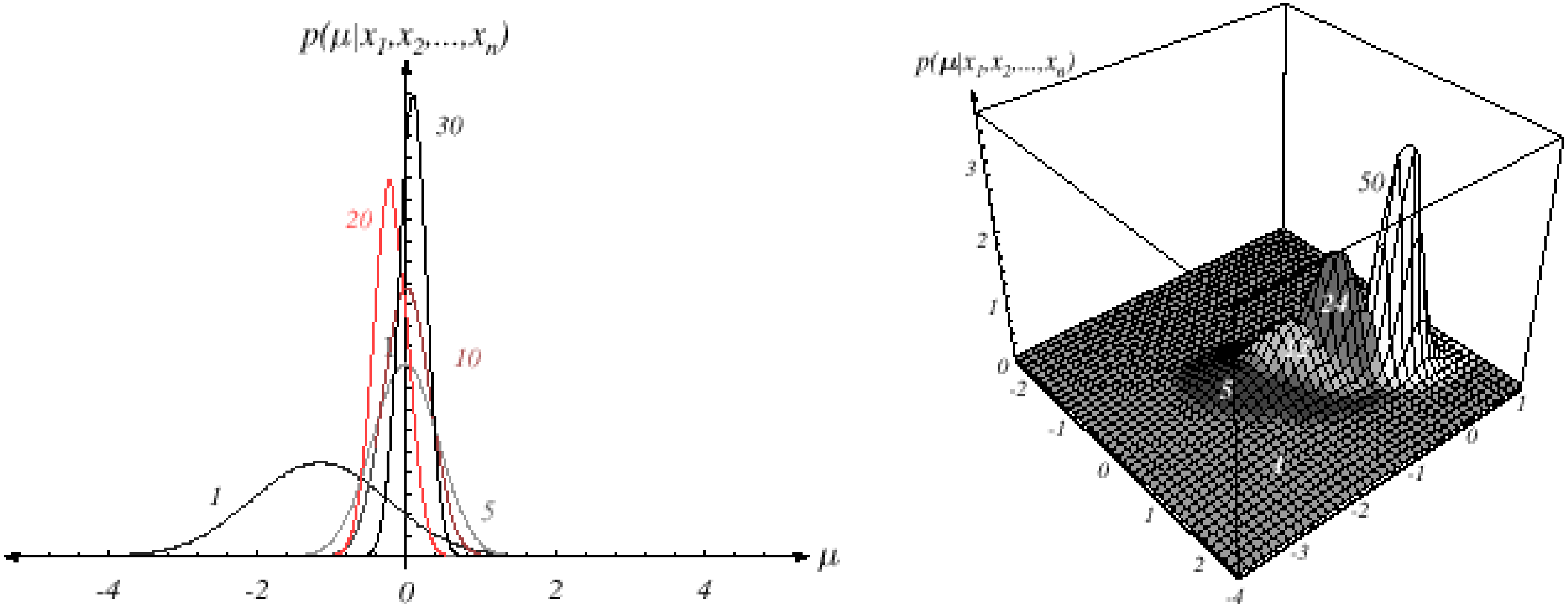


FIGURE 3.2. Bayesian learning of the mean of normal distributions in one and two dimensions. The posterior distribution estimates are labeled by the number of training samples used in the estimation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Gaussian Case

– Compute $p(x|D)$ we have

$$p(x\,|\,D)=\int p(x\,|\,\mu)\,p(\mu\,|\,D)\,d\mu$$

$$=\int \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]\frac{1}{\sqrt{2\pi}\sigma_n}\exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right]d\mu$$

$$\sim N(\mu_n,\sigma^2+\sigma_n^2)$$

- Comment: to compute $p(x|D)$ whose parametric form is known to be $p(x|\mu)\sim N(\mu,\sigma^2)$, we can just replace μ with μ_n and σ^2 as $(\sigma^2+\sigma_n^2)$

Recursive Bayes Learning

- Let $D^n=\{x_1, x_2, \ldots, x_n\}$ where n is number of training samples,

becaues $p(D\,|\,\theta)=\prod_{k=1}^{n} p(x_k\,|\,\theta)$

when $n>1$ **for every new sample** x_n **we have**

$$p(D^n\,|\,\theta)=p(x_n\,|\,\theta)\,p(D^{n-1}\,|\,\theta)$$

and $p(D^0\,|\,\theta)=p(\theta)$,

then we have the recursive relation

$$p(\theta\,|\,D^n)=\frac{p(x_n\,|\,\theta)\,p(\theta\,|\,D^{n-1})}{\int p(x_n\,|\,\theta)\,p(\theta\,|\,D^{n-1})d\theta}$$

Recursive Bayes Learning

- Use this recursive relation, we can calculate the sequence of densities $p(\theta)$, $p(\theta|x_1)$, $p(\theta|x_1, x_2)$,…., $p(\theta|D^n)$. This is called recursive Bayes approach.
- When we reach a Dirac delta function around the true parameter, we have Bayesian learning

Example 1 Recursive Bayes Learning

- Problem statement:
  - We assume our 1-D samples coming from a uniform distribution with unknown parameter θ

$$\boldsymbol{p(x \mid \theta) \sim U(0,\theta) = \begin{cases} 1/\theta & 0 \le x \le \theta \\ 0 & \textbf{otherwise} \end{cases}}$$

  - We also assume the prior knowledge about θ

$$\boldsymbol{p(\theta) \sim U(0,10)}$$

  - Now we observed four samples D={4, 7, 2, 8}
  - We like to estimate θ

Example 1 Recursive Bayes Learning

- Recursive Bayes learning:
  - Before the first sample arrives, we have

$$p(\theta|D^0) = p(\theta)=U(0,10)$$

  - When the first sample x_1=4 arrives, we have

$$\boldsymbol{p(\theta \mid D^1) \propto p(x_1 \mid \theta)\, p(\theta \mid D^0) =} \begin{cases} 1/\boldsymbol{\theta} & 4 \le \boldsymbol{\theta} \le 10 \\ 0 & \textbf{otherwise} \end{cases}$$

  - When the second sample x_2=7 arrives, we have

$$\boldsymbol{p(\theta \mid D^2) \propto p(x_2 \mid \theta)\, p(\theta \mid D^1) =} \begin{cases} 1/\boldsymbol{\theta}^2 & 7 \le \boldsymbol{\theta} \le 10 \\ 0 & \textbf{otherwise} \end{cases}$$

Example 1 Recursive Bayes Learning

– When the third sample x_3=2 arrives, we have

$$p(\theta \mid D^3) \propto p(x_3 \mid \theta)\, p(\theta \mid D^2) = \begin{cases} 1/\theta^3 & 7 \le \theta \le 10 \\ 0 & \textbf{otherwise} \end{cases}$$

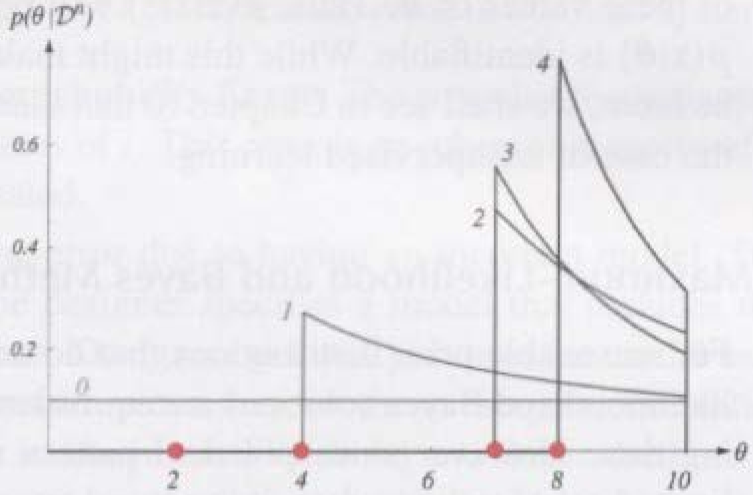
– When the forth sample x_4=8 arrives, we have

$$p(\theta \mid D^4) \propto p(x_4 \mid \theta)\, p(\theta \mid D^3) = \begin{cases} 1/\theta^4 & 8 \le \theta \le 10 \\ 0 & \textbf{otherwise} \end{cases}$$

– The general solution is

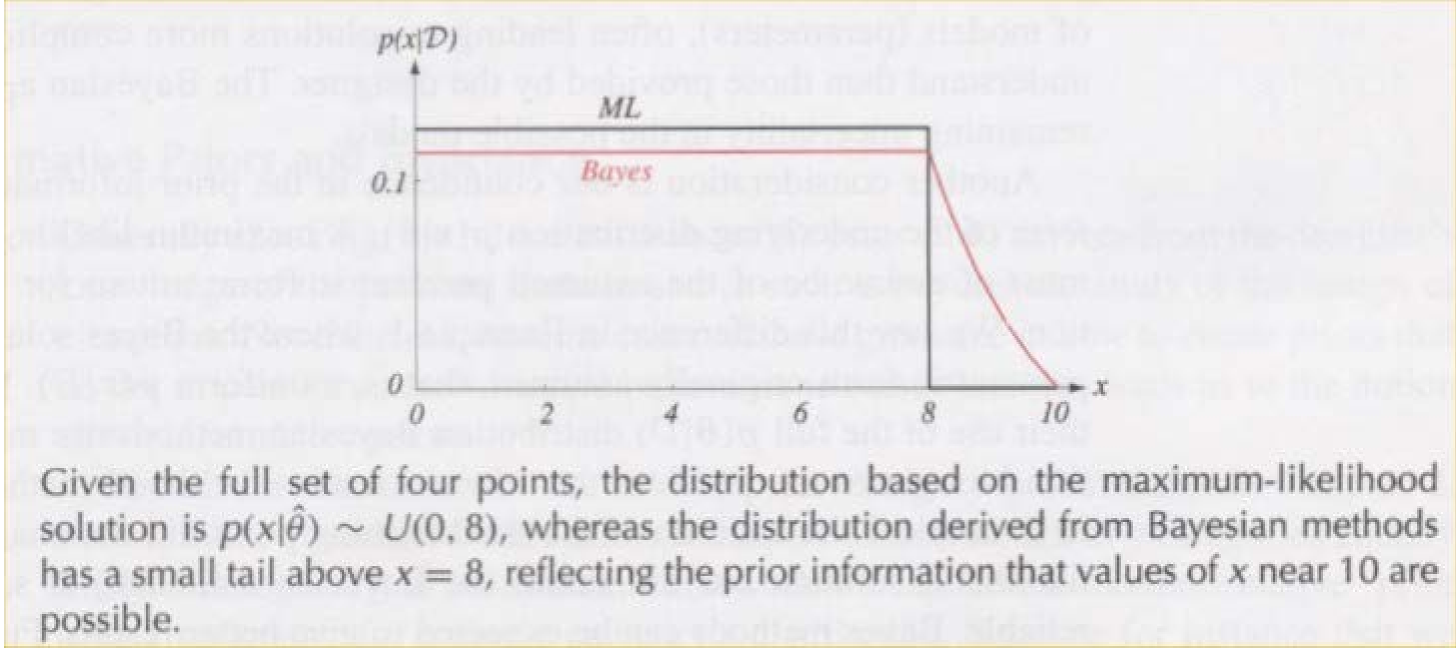
$$p(\theta \mid D^n) \propto 1/\theta^n \quad \textbf{for} \ \max_x[D^n] \le \theta \le 10$$

Example 1 Recursive Bayes Learning



The posterior $p(\theta|\mathcal{D}^n)$ for the model and n points in the data set in this Example. For $n = 0$, the posterior starts our as a flat, uniform density from 0 to 10, denoted $p(\theta) \sim U(0, 10)$. As more points are incorporated, it becomes increasingly peaked at the value of the highest data point.

Example 1 Recursive Bayes Learning



Given the full set of four points, the distribution based on the maximum-likelihood solution is $p(x|\hat{\theta}) \sim U(0, 8)$, whereas the distribution derived from Bayesian methods has a small tail above $x = 8$, reflecting the prior information that values of x near 10 are possible.

ML Method v.s. Bayes Method

- If the prior distribution well includes the true solution, ML solution and Bayes solution are equivalent in the asymptotic limit of infinite training data.
- ML method has less computational complexity and better interpretability
- ML solution is always consistent with the model structure assumption, Bayes solution may be a little different (example 1).
- Bayes method uses more information from data than ML method.

Sources of Classification Errors

- Bayes error: due to overlapping densities $p(x|\omega_i)$ for different i. This error can never be eliminated.
- Model error: due to incorrect model assumption. This error can be reduced with better prior knowledge.
- Estimation error: due to the limited number of training sample in parameter estimation. This error can be reduced by increasing sample size.