Maximum Likelihood Parameter Estimations

Computer Science- Pattern Recognition

Prof. Dr. Dhahir A. Abdullah

Parameter Estimation

- In previous chapters:
  - We could design **an optimal classifier** if we **knew** the **prior probabilities** P(wi) and the **class-conditional probabilities** P(x|wi)

- Unfortunately, in pattern recognition applications **we rarely have this kind of complete knowledge** about the probabilistic structure of the problem.

Parameter Estimation

- We have a number of design samples or training data.
- The problem is to **find some way to use this information to design or train the classifier.**
- One approach:
  - **Use the samples to estimate** the unknown probabilities and probability densities,
  - And then use the resulting estimates as if they **were the true values.**

Parameter Estimation

- We have a number of design samples or training data.
- The problem is to **find some way to use this information to design or train the classifier.**
- One approach:
  - **Use the samples to estimate** the unknown probabilities and probability densities,
  - And then use the resulting estimates as if they **were the true values.**

Parameter Estimation

- **What you are going to estimate in your Homework ?**

Parameter Estimation

- If we know the **number parameters** in advance and our general **knowledge about the problem** permits us to **parameterize the conditional desities** then severity of the problem can be reduced significantly.

Parameter Estimation

- For example:
  - We can reasonaby assume that the **p(x|wi) is a normal density** with mean **μ** and covariance matrix $\sum$,
  - We do not know the exact values of these quantities,
  - However, this knowledge simplifies the problem **from one of estimating an unknown function p(x|wi) to one of estimating the parameters the mean μ_i and covariance matrix $\sum_i$**
  - **Estimating p(x|wi) → estimating μ_i and $\sum_i$**

Parameter Estimation

- Data availability in a Bayesian framework
  - We could design an optimal classifier if we knew:
    - $P(\omega_i)$ (priors)
    - $P(x \mid \omega_i)$ (class-conditional densities)

    Unfortunately, we rarely have this complete information!

- Design a classifier from a training sample
  - No problem with prior estimation
  - Samples are often too small for class-conditional estimation (large dimension of feature space!)

Parameter Estimation

- Given a bunch of data from each class how to estimate the parameters of class conditional densities, $P(x \mid \omega_i)$?
- Ex: $P(x \mid \omega_i) = N(\mu_j, \Sigma_j)$ is Normal.

  Parameters $\theta_j = (\mu j, \Sigma j)$

Two major approaches

- **Maximum-Likelihood Method**
- **Bayesian Method**

  - Use $P(\omega_i \mid x)$ for our classification rule!
  - Results are nearly identical, but the approaches are different

Maximum-Likelihood vs. Bayesian:

- Maximum Likelihood
- Parameters are fixed but unknown!
- Best parameters are obtained by maximizing the probability of obtaining the samples observed

- Bayes
- Parameters are random variables having some known distribution
- Best parameters are obtained by estimating them given the data

Major assumptions

- A priori information P(ω_i) for each category is available
- Samples are i.i.d. and P(x | ω_i) is Normal

$$P(x \mid \omega_i) \sim N(\mu_i, \Sigma_i)$$

- Note: Characterized by 2 parameters

Maximum-Likelihood Estimation

- Has good convergence properties as the sample size increases
- Simpler than any other alternative techniques

Maximum-Likelihood Estimation

We assume that samples are collected randomly from a given form of distribution, whose parameters are unknown.Unknown parameters are denoted by the vector $\boldsymbol{\theta}$.

We partition our training data $\mathbf{X}$ into c class specific subsets $D_1 \ldots D_c$, assuming that the data in D_j have been drawn randomly according to the distribution $p(\mathbf{x}|\omega_j)$

If , for example, we know each distribution is normal, $p(\mathbf{x}|\omega_j) \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ but we do not know the mean and covariance, then our problem is to estimate

$$\boldsymbol{\theta}_j = \begin{pmatrix} \boldsymbol{\mu}_j \\ \boldsymbol{\Sigma}_j \end{pmatrix}$$

To make the dependence of $p(\mathbf{x}|\omega_j)$ on $\boldsymbol{\theta}_j$ more specific, we can write $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)$. Furthermore, to make our life easier – and why not – we assume that all $\boldsymbol{\theta}_j$ are independent, i.e., knowing one does not tell us anything about the others. This will allow us to work for each $\boldsymbol{\theta}_j$ separately, without worrying about interdependence of the parameters. We may therefore drop the class subscripts j

Maximum-Likelihood Estimation

- General principle
  - Assume we have c classes and
  - The samples in class j have been drawn according to the probability law $p(x \mid \omega_j)$

    $\mathbf{p(x \mid \omega_j) \sim N(\mu_j, \Sigma_j)}$

    $\mathbf{p(x \mid \omega_j) \equiv P(x \mid \omega_j, \theta_j)}$ where:

Maximum-Likelihood Estimation

- Our problem is to **use the information provided by the training samples** to **obtain good estimates for the unknown parameter** vectors associated with each category.

Likelihood Function

- Use the information provided by the training samples to estimate $\theta = (\theta_1, \theta_2, \ldots, \theta_c)$, each θ_i (i = 1, 2, …, c) is associated with each category
- Suppose that D contains n samples, $x_1, x_2, \ldots, x_n$

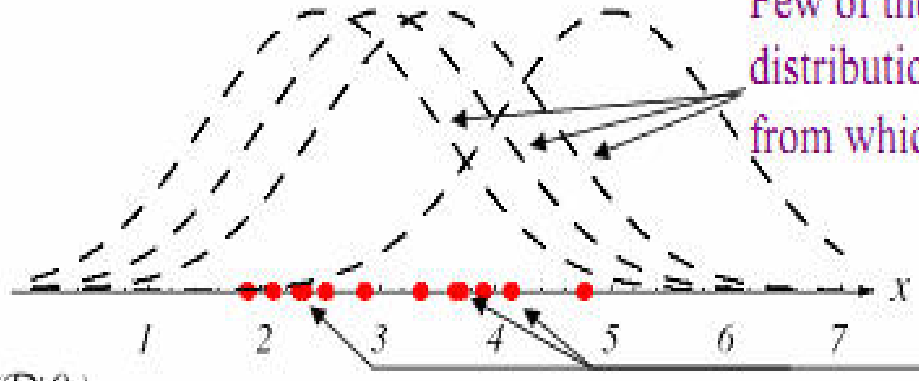
$$P(D \mid \theta) = \prod_{k=1}^{k=n} P(x_k \mid \theta) = F(\theta)$$

$P(D \mid \theta)$ is called the likelihood of θ w.r.t. the set of samples)

Goal: find an estimate $\hat{\theta}$

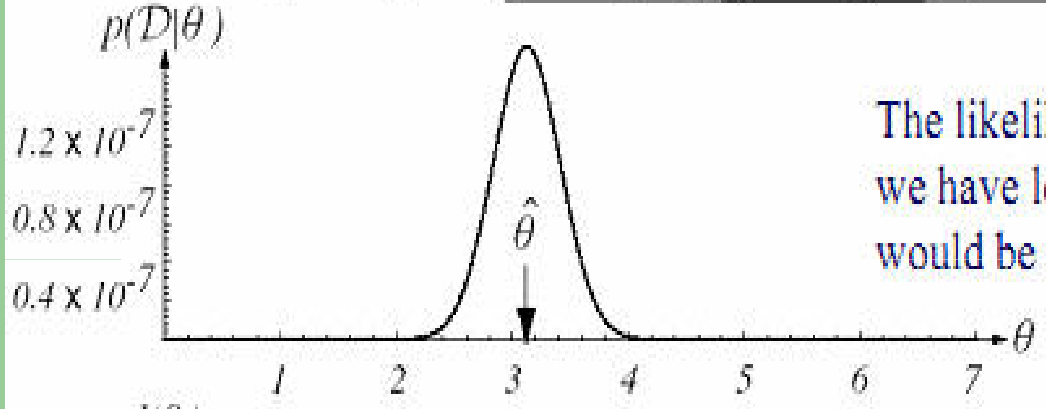
- Find θ which maximizes $P(D \mid \theta)$

"It is the value of θ that best agrees with the actually observed training sample"

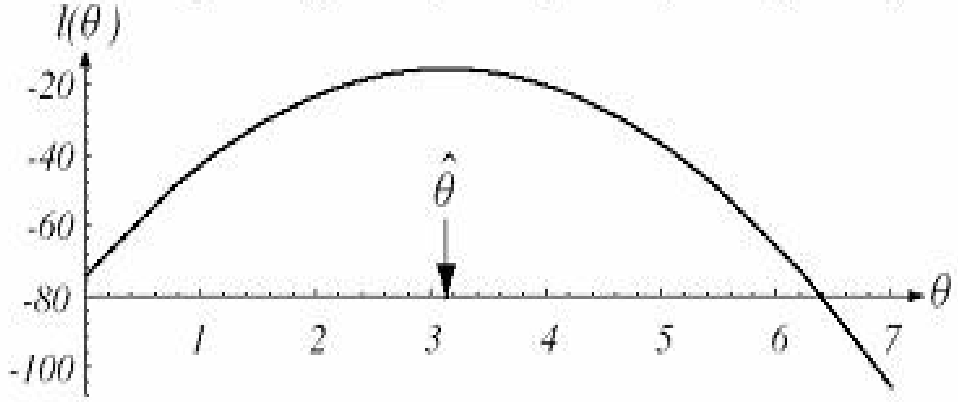


Few of the infinite number of possible Gaussian distributions of some variance and unknown mean θ from which the observed values might be drawn

Observed data points (1D)



The likelihood $p(D \mid \boldsymbol{\theta})$ as a function of the mean.If we have lots of training points, this likelihood would be very narrow.



The **θ** that maximizes this likelihood – the value we are looking for – also maximizes the logarithm of the likelihood $l(\boldsymbol{\theta})$. Since log is a monotonically increasing function

Maximize log likelihood function: $l(\theta) = \ln P(D \mid \theta)$

- Let $\theta = (\theta_1, \theta_2, \ldots, \theta_p)^t$ and let ∇_θ be the gradient operator

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

- New problem statement: Determine θ that maximizes the log-likelihood

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

Set of necessary conditions for an optimum is:

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

$$\nabla_\theta l = 0$$

- Example of a specific case: unknown μ

  - $P(x_i \mid \mu) \sim N(\mu, \Sigma)$

    (Samples are drawn from a multivariate normal population)

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

$$\text{and} \quad \nabla_{\theta\mu} \ln P(\mathbf{x}_k \mid \mu) = \Sigma^{-1}(\mathbf{x}_k - \mu)$$

$\theta = \mu$ therefore:

- The ML estimate for μ must satisfy:

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

- **Multiplying by Σ and rearranging, we obtain:**

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

Just the arithmetic average of the samples of the training samples!

Conclusion:

If $P(x_k \mid \omega_j)$ (j = 1, 2, …, 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!

- ML Estimation:
  - Gaussian Case: ***unknown μ and σ***

    $\theta = (\theta_1, \theta_2) = (\mu, \sigma^2)$

$$l = \ln P(x_k \mid \theta) = -\frac{1}{2}\ln 2\pi\theta_2 - \frac{1}{2\theta_2}(x_k - \theta_1)^2$$

$$\nabla_\theta l = \begin{pmatrix} \frac{\sigma}{\sigma\theta_1}(\ln P(x_k \mid \theta)) \\ \frac{\sigma}{\sigma\theta_2}(\ln P(x_k \mid \theta)) \end{pmatrix} = 0$$

$$\begin{cases} \frac{1}{\theta_2}(x_k - \theta_1) = 0 \\ -\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2} = 0 \end{cases}$$

Summation:

$$\begin{cases} \sum_{k=1}^{k=n} \frac{1}{\hat{\theta}_2}(x_k - \theta_1) = 0 & (1) \\ -\sum_{k=1}^{k=n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{k=n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0 & (2) \end{cases}$$

Combining (1) and (2), one obtains:

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

Example 1:

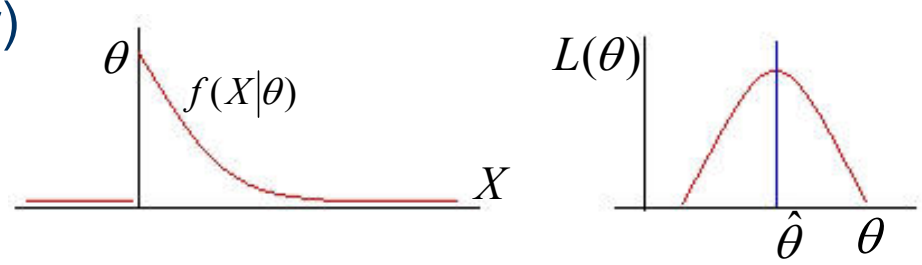
Consider an exponential distribution

$$f(X;\theta) = \begin{cases} \theta e^{-\theta x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

(single feature, single parameter)

With a random sample

$$\{X_1, X_2, \ldots, X_n\}$$



Estimate θ ?

Example 1:

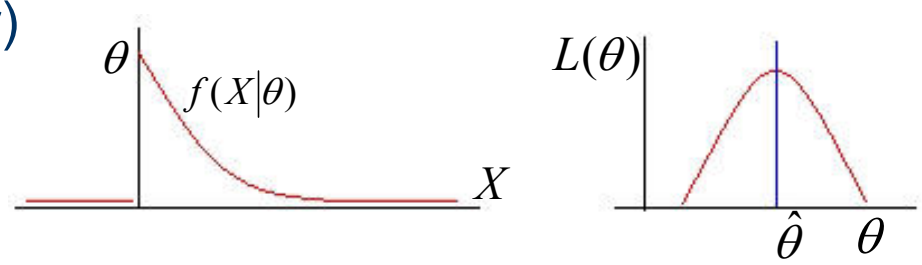
Consider an exponential distribution

$$f(X;\theta) = \begin{cases} \theta e^{-\theta x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

(single feature, single parameter)

With a random sample

$$\{X_1, X_2, \ldots, X_n\}$$



:

$$L(\theta) = f(X_1, X_2, \ldots, X_n|\theta) = \prod_{i=1}^{n} \theta . e^{-\theta . x_i}$$ valid for $x \geq 0$

$$l(\theta) = \ln L(\theta) = \sum_{i=1}^{n} \ln\theta - \theta \sum_{i=1}^{n} x_i = n \ln\theta - \theta \sum_{i=1}^{n} x_i$$

$$\frac{dl}{d\theta} = \frac{d \ln L(\theta)}{d\theta} = \frac{n}{\theta} - \sum_{i=1}^{n} x_i = 0$$

$$\Rightarrow \frac{n}{\hat{\theta}} = \sum_{i=1}^{n} x_i \Rightarrow \hat{\theta} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} x_i}$$ (inverse of average)

Example 2:

Multivariate Gaussian with unknown mean vector M.

Assume Σ is known.

k samples from the same distribution:

$$X_1, X_2, \ldots\ldots\ldots, X_k \qquad \text{(iid)}$$

$$L(X \mid M) = \prod_{i=1}^{k} \frac{1}{(2\pi)^{n/2}\left|\Sigma\right|^{1/2}} e^{-\frac{1}{2}(X_i - M)^T \Sigma^{-1}(X_i - M)}$$

$$\nabla l = \nabla_M \log L = \sum_{i=1}^{k} \nabla_M \log \frac{1}{(2\pi)^{n/2}\left|\Sigma\right|^{1/2}} e^{\frac{1}{2}(X_i - M)^T \Sigma^{-1}(X_i - M)}$$

$$= \sum_{i=1}^{k} \nabla_M \left(\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(X_i - M)^T \Sigma^{-1}(X_i - M)\right)$$

(linear algebra)

$$= \sum_{i=1}^{k} \left(\Sigma^{-1}(X_i - \hat{M})\right)$$

$$\Rightarrow 0 = \Sigma^{-1}(\sum_{i=1}^{k} X_i - k\hat{M})$$

$$\hat{M} = \frac{1}{k}\sum_{i=1}^{k} X_i$$ (sample average or sample mean)

Example 3:

Binary variables with unknown parameters $p_i, 1 \le i \le n$ (n parameters)

How to estimate these $p_i, 1 \le i \le n$

Think about your homework ?

Example 3:

Binary variables with unknown parameters $p_i, 1 \le i \le n$ (n parameters)

$$\log P(X) = \sum_{i=1}^{n} x_i \log p_i + \sum_{i=1}^{n} (1 - x_i) \log(1 - p_i)$$

So,

$$l = \log L = \sum_{j=1}^{k} \log P(X_j)$$ k samples

$$= \sum_{j=1}^{k} \left(\sum_{i=1}^{n} x_{ij} \log p_i + \sum_{i=1}^{n} (1 - x_{ij}) \log(1 - p_i) \right)$$

here x_{ij} is the i^{th} element of j^{th} sample X_j.

So,

$$\nabla_{p_i} \log L = \begin{bmatrix} \frac{\partial}{\partial p_1} \log L \\ \frac{\partial}{\partial p_2} \log L \\ \vdots \\ \frac{\partial}{\partial p_n} \log L \end{bmatrix}$$

$$\frac{\partial}{\partial p_i} \log L = \sum_{j=1}^{k} \left(\frac{x_{ij}}{p_i} - ((1 - x_{ij})(1 - p_i)) \right)$$

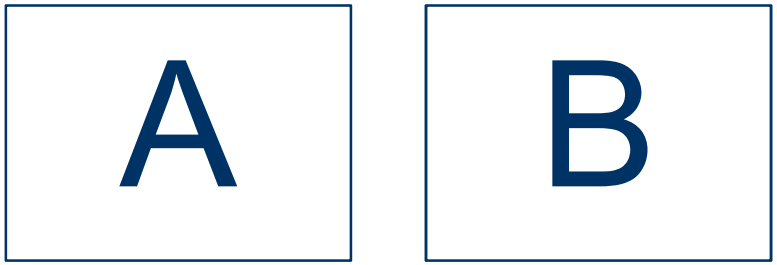
$$\Rightarrow 0 = \frac{1}{\hat{p}_i} \sum_{j=1}^{k} x_{ij} - \frac{1}{1 - \hat{p}_i} \sum_{j=1}^{k} (1 - x_{ij})$$

$$\Rightarrow \hat{p}_i = \frac{1}{k} \sum_{j=1}^{k} x_{ij}$$

❖ $\hat{p}_i$ is the sample average of the feature.

Since X_i is binary, $\sum_{j=1}^{k} x_{ij}$ will be the same as counting the occurances of '1'.

Consider character recognition problem with binary matrices.



For each pixel, count the number of 1's and this is the estimate of p_i .