Pattern Classification

All materials in these slides were taken from
Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000
with the permission of the authors and the publisher
Chapter 3: Maximum-Likelihood & Bayesian Parameter Estimation

- Introduction
- Maximum-Likelihood Estimation
- Bayesian Estimation
- Curse of Dimensionality
- Component analysis & Discriminants
- EM Algorithm
Introduction

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!

Design a classifier based on a set of labeled training samples (supervised learning)
- Assume priors are known
- Need sufficient no. of training samples for estimating class-conditional densities, especially when the dimensionality of the feature space is large
Assumption about the problem: parametric model of $P(x | \omega_i)$ is available.

Normality of $P(x | \omega_i)$

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

Characterized by 2 parameters.

Estimation techniques
- Maximum-Likelihood (ML) and Bayesian estimation
- Results of the two procedures are nearly identical, but the approaches are different.
- Parameters in ML estimation are fixed but unknown! Bayesian parameter estimation procedure, by its nature, utilizes whatever prior information is available about the unknown parameter.
- MLE: Best parameters are obtained by maximizing the probability of obtaining the samples observed.
- Bayesian methods view the parameters as random variables having some known prior distribution; How do we know the priors?
- In either approach, we use $P(\omega_i | x)$ for our classification rule!
**Maximum-Likelihood Estimation**

- Has good convergence properties as the sample size increases; estimated parameter value approaches the true value as $n$ increases.
- Simpler than any other alternative technique.

**General principle**

- Assume we have $c$ classes and

$$P(x | \omega_j) \sim N(\mu_j, \Sigma_j)$$

$$P(x | \omega_j) \equiv P(x | \omega_j, \theta_j),$$

where

$$\theta = (\mu_j, \Sigma_j) = (\mu_{1j}, \mu_{2j}, \ldots, \sigma_{11}^j, \sigma_{22}^j, \text{cov}(x_j^m, x_j^n), \ldots)$$

Use class $\omega_j$ samples to estimate class $\omega_j$ parameters.
Use the information in training samples to estimate \( \theta = (\theta_1, \theta_2, \ldots, \theta_c) \); \( \theta_i \) \((i = 1, 2, \ldots, c)\) is associated with the \( i \)th category.

Suppose sample set \( D \) contains \( n \) iid samples, \( x_1, x_2, \ldots, x_n \).

\[
P(D | \theta) = \prod_{k=1}^{k=n} P(x_k | \theta) = F(\theta)
\]

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

ML estimate of \( \theta \) is, by definition, the value \( \hat{\theta} \) that maximizes \( P(D | \theta) \).

“It is the value of \( \theta \) that best agrees with the actually observed training samples.”
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 $\theta$ whereas the conditional density $p(x|\theta)$ is shown as a function of $x$. Furthermore, as a function of $\theta$, 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.
Optimal estimation

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

$$
\nabla_\theta = \begin{bmatrix}
\frac{\partial}{\partial \theta_1}, & \frac{\partial}{\partial \theta_2}, & \cdots, & \frac{\partial}{\partial \theta_p}
\end{bmatrix}^t
$$

- We define $l(\theta)$ as the log-likelihood function
  $$
l(\theta) = \ln P(D | \theta)$$

- New problem statement:
  determine $\theta$ 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 $\mu$

- $P(x \mid \mu) \sim N(\mu, \Sigma)$
  
  (Samples are drawn from a multivariate normal population)

\[
\ln P(x_k \mid \mu) = -\frac{1}{2} \ln[(2\pi)^d \mid \Sigma\mid] - \frac{1}{2} (x_k - \mu)^t \Sigma^{-1} (x_k - \mu)
\]

and 
\[
\nabla_{\theta \mu} \ln P(x_k \mid \mu) = \Sigma^{-1} (x_k - \mu)
\]

$\theta = \mu$, therefore the ML estimate for $\mu$ must satisfy:

\[
\sum_{k=1}^{k=n} \Sigma^{-1} (x_k - \hat{\mu}) = 0
\]
• Multiplying by $\Sigma$ and rearranging, we obtain:

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

which is the arithmetic average or the mean of the samples of the training samples!

Conclusion:

Given $P(x_k | \omega_j)$, $j = 1, 2, \ldots, c$ to be Gaussian in a $d$-dimensional feature space, estimate the vector $\theta = (\theta_1, \theta_2, \ldots, \theta_c)^t$ and perform a classification using the Bayes decision rule of chapter 2!
ML Estimation:

Univariate Gaussian Case: *unknown* $\mu$ & $\sigma$

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

\[
I = \ln P(x_k \mid \theta) = -\frac{1}{2}\ln 2\pi \theta_2 - \frac{1}{2\theta_2}(x_k - \theta_1)^2
\]

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

\[
\begin{cases}
\frac{1}{\theta_2} (x_k - \theta_1) = 0 \\
-\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2} = 0
\end{cases}
\]
Summation:

\[
\begin{align*}
\sum_{k=1}^{k=n} \frac{1}{\hat{\theta}_2} (x_k - \theta_1) &= 0 \quad (1) \\
- \sum_{k=1}^{k=n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{k=n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} &= 0 \quad (2)
\end{align*}
\]

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

\[
\begin{align*}
\mu &= \sum_{k=1}^{k=n} \frac{x_k}{n} \quad ; \quad \sigma^2 = \frac{\sum_{k=1}^{k=n} (x_k - \mu)^2}{n}
\end{align*}
\]
Bias

- ML estimate for $\sigma^2$ is biased

$$E\left( \frac{1}{n} \Sigma (x_i - \bar{x})^2 \right) = \frac{n - 1}{n} \sigma^2 \neq \sigma^2$$

- An unbiased estimator for $\Sigma$ is:

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

Sample covariance matrix
**Bayesian Estimation** (Bayesian learning approach for pattern classification problems)

- In MLE $\theta$ was supposed to have a fixed value
- In BE $\theta$ is a random variable
- The computation of posterior probabilities $P(\omega_i | x)$ lies at the heart of Bayesian classification
- **Goal:** compute $P(\omega_i | x, D)$

Given the training sample set $D$, Bayes formula can be written

$$P(\omega_i | x, D) = \frac{P(x | \omega_i, D).P(\omega_i | D)}{\sum_{j=1}^{c} P(x | \omega_j, D).P(\omega_j | D)}$$
To demonstrate the preceding equation, use:

\[
P(x, D | \omega_i) = P(x | D, \omega_i) \cdot P(D | \omega_i)
\]
\[
P(x | D) = \sum_{j} P(x, \omega_j | D)
\]
\[
P(\omega_i) = P(\omega_i | D) \quad \text{(Training sample provides this!)}
\]

Thus:

\[
P(\omega_i | x, D) = \frac{P(x | \omega_i, D_i) \cdot P(\omega_i)}{\sum_{j=1}^{c} P(x | \omega_j, D) \cdot P(\omega_j)}
\]
Bayesian Parameter Estimation: Gaussian Case

**Goal:** Estimate $\theta$ using the a-posteriori density $P(\theta | D)$

The univariate Gaussian case: $P(\mu | D)$

$\mu$ is the only unknown parameter

\[
P(x | \mu) \sim N(\mu, \sigma^2) \\
P(\mu) \sim N(\mu_0, \sigma_0^2)
\]

$\mu_0$ and $\sigma_0$ are known!
\( P(\mu \mid D) = \frac{P(D \mid \mu).P(\mu)}{\int P(D \mid \mu).P(\mu)d\mu} \quad (1) \)

\[
\prod_{k=1}^{k=n} P(x_k \mid \mu).P(\mu)
\]

- Reproducing density

\( P(\mu \mid D) \sim N(\mu_n, \sigma_n^2) \quad (2) \)

The updated parameters of the prior:

\[
\mu_n = \left( \frac{n\sigma_0^2}{n_0\sigma_0^2 + \sigma^2} \right) \hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \cdot \mu_0
\]

and

\[
\sigma_n^2 = \frac{\sigma_0^2\sigma^2}{n\sigma_0^2 + \sigma^2}
\]
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.
The univariate case \( P(x \mid D) \)

- \( P(\mu \mid D) \) has been computed
- \( P(x \mid D) \) remains to be computed!

\[
P(x \mid D) = \int P(x \mid \mu)P(\mu \mid D) d\mu \quad \text{is Gaussian}
\]

It provides:

\[
P(x \mid D) \sim \mathcal{N}(\mu_n, \sigma^2 + \sigma_n^2)
\]

Desired class-conditional density \( P(x \mid D_j, \omega_j) \)

\( P(x \mid D_j, \omega_j) \) together with \( P(\omega_j) \) and using Bayes formula, we obtain the Bayesian classification rule:

\[
\max_{\omega_j} \left[ P(\omega_j \mid x, D) \right] \equiv \max_{\omega_j} \left[ P(x \mid \omega_j, D_j)P(\omega_j) \right]
\]
Bayesian Parameter Estimation: General Theory

- P(x | D) computation can be applied to any situation in which the unknown density can be parametrized: the basic assumptions are:
  - The form of P(x | θ) is assumed known, but the value of θ is not known exactly
  - Our knowledge about θ is assumed to be contained in a known prior density P(θ)
  - The rest of our knowledge about θ is contained in a set D of n random variables x_1, x_2, ..., x_n that follows P(x)
The basic problem is:
“Compute the posterior density $P(\theta \mid D)$”
then “Derive $P(x \mid D)$”

Using Bayes formula, we have:

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{\int P(D \mid \theta)P(\theta)d\theta},$$

And by independence assumption:

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

Classification problems with large number of features (hundreds or even thousands) is frequently encountered

Classification accuracy depends upon the the number of features (dimensionality), their discrimination ability and the amount of training data

Two-class multivariate normal with the same covariance matrix

\begin{equation}
    P(\text{error}) = \frac{1}{\sqrt{2\pi} r/2} \int_{-\infty}^{\infty} e^{-\frac{u^2}{2}} du
\end{equation}

where: \( r^2 = (\mu_1 - \mu_2)^t \Sigma^{-1} (\mu_1 - \mu_2) \)

\[ \lim_{r \to \infty} P(\text{error}) = 0 \]
If features are independent then:

\[ \Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2) \]

This shows how each feature contributes to reducing the probability of error. Most useful features are the ones for which the difference between the means is large relative to the standard deviation.

An obvious way to reduce the error further is to introduce new, independent feature.

It has frequently been observed in practice that, when the number of training samples is small, beyond a certain point, the inclusion of additional features leads to worse rather than better performance: (i) we have the wrong model, (ii) small no. of available training samples to estimate parameters.
FIGURE 3.3. Two three-dimensional distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace—here, the two-dimensional $x_1 - x_2$ subspace or a one-dimensional $x_1$ subspace—there can be greater overlap of the projected distributions, and hence greater Bayes error. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
Component Analysis and Discriminants

- Combine features in order to reduce the dimension of the feature space
- Linear combinations are simple to compute and tractable
- Project high dimensional data onto a lower dimensional space
- Two classical approaches for finding “optimal” linear transformation
  - PCA (Principal Component Analysis) “Projection that best represents the data in a least-square sense”
  - MDA (Multiple Discriminant Analysis) “Projection that best separates the data in a least-squares sense”