Statistical Pattern Recognition

Notes on Speech and Audio Processing

Chia-Ping Chen

Department of Computer Science and Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan ROC

Introduction

- Pattern classification is the problem of determining the category of a data. For example, in an orange packaging factory, it is needed to distinguish between high-quality and low-quality oranges and put them in different bins. In addition, the size may be a factor, too.
- The challenge, is to do it automatically by machines.
- The word “pattern” is quite graphical, but the same word can apply to more intangible ideas such as acoustic and linguistic patterns, among others.

Examples of Pattern Classification

- Swedish basketball players vs. ASR researchers. If we want to distinguish between basketball players and ASR researchers, we can get some hint from height h and weight w. We may collect (h, w)s from basketball players and ASR researchers to decide the best strategy for classification.
- Everyone can distinguish different vowels easily. But how do you teach a machine to do that? What numbers are indicative of the vowel identity? Such numbers are called **features**. The formants are good features for the classification of vowels.

Feature Extraction

- The first step in pattern recognition is to design a representation of the data that is informative of the class membership, called a feature set.
- There are a few factors in designing features.
  - within-class variance: the variance of feature values of data of the same class.
  - between-class variance: the variance of feature values of data of different class.
  - numbers of features (a.k.a. feature dimension)
  - geometry of class boundaries
  - reliability

Deterministic Classifiers

- Suppose we have decided the features for classification. Every data is now a point in the feature space.
- Classification is now a “partition” of the feature space, where the points in a region is classified to the same class. Examples of classifiers (partitions) are
  - minimum distance classifier: a data point is classified to be of the same class as the nearest neighbor.
  - discriminant function classifier: given a data point, compute the class-dependent discriminant function values. The class with the largest value is assigned to the data point.

Statistical Framework

- Many classification problems have a probabilistic nature: The instances of a given class may well have different looks.
- The probability theory is a natural framework to describe random processes. We next develop a few results for statistical pattern classifications.

Random Variables

- A random variable X is a function that maps the sample space of a random experiment to the set of real numbers.
- A random variable can be discrete or continuous.
  - For a discrete random variable, the probability mass function $p(x)$ is defined by

$$p(x) = Pr(X = x),$$

  - For a continuous random variable, the density function $f(x)$ is defined by

$$Pr(x < X \leqslant x + dx) = f(x)dx.$$

Joint Probability

- Two random variables may be related and we may want to describe them together. We define the joint distribution function for X and Y by

$$F(x, y) = Pr(X \leq x, Y \leq y).$$

- If X and Y are discrete, we can define the joint probability mass function by

$$p(x, y) = Pr(X = x, Y = y).$$

  In this case, the distribution function is step-wise.

Joint Density Function

- If both X and Y are continuous, we can define the joint density function $f(x, y)$ by the distribution function

$$F(x, y) = \int_{-\infty}^{x} \int_{-\infty}^{y} f(u, v) du dv.$$

That is,

$$f(x, y) = \frac{\partial^2 F(x, y)}{\partial x \partial y}.$$

- Note that $F(x, y)$ is non-decreasing in "upper-right" directions and $f(x, y)$ is non-negative everywhere.

Conditional Probability: Discrete

- For two events A and B, the conditional probability that A occurs given B occurs is defined by

$$Pr(A|B) = \frac{Pr(A \cap B)}{Pr(B)}.$$

- For two discrete RVs, X and Y, the conditional probability of $\{Y = y\}$ given $\{X = x\}$ is

$$Pr(Y = y|X = x) = \frac{Pr(X = x, Y = y)}{Pr(X = x)} = \frac{p(x, y)}{p(x)}.$$

Conditional Probability: Continuous

- For two continuous RVs, one may suspect the legitimacy of the ratio.
- In fact, this is not an issue. Both the numerator and denominator approach 0 but the limit is well-defined.

$$\frac{Pr(y < Y \leq y + dy,\ x < X \leq x + dx)}{Pr(x < X \leq x + dx)}$$

$$= \frac{f(x,y)dxdy}{f(x)dx} = \frac{f(x,y)dy}{f(x)} = f(y|x)dy.$$

- Thus we have $f(y|x) = \frac{f(x,y)}{f(x)}$.

Conditional Probability: Mixed case

- The last case we will discuss is the mixed case: Y is discrete and X is continuous. In this case,

$$\frac{Pr(Y = y,\ x < X \leq x + dx)}{Pr(x < X \leq x + dx)} = \frac{f(x, y)dx}{\sum_{y'} f(x, y')dx}$$

$$\Rightarrow p(y|x) = \frac{f(x, y)}{\sum_{y'} f(x, y')} = \frac{p(y)f(x|y)}{\sum_{y'} p(y')f(x|y')}.$$

- In pattern recognition we often have continuous feature and discrete membership. What we have shown here is that the conditional probability of class given features is well-defined.

Class-Related Probability Functions

Assume that the data is X. We want to make a decision of its membership ω. The following probabilities are crucial to this decision.

- a priori (or prior) probability $p(\omega)$: the probability that a sample (not yet observed) is from class ω.
- conditional probability $f(x|\omega)$: the probability density that a sample of class ω has value x.
- a posteriori probability (or posterior) $p(\omega|x)$: the probability a given sample $X = x$ is of class ω.

Minimum Error Classification

- A moment of reflection should convince you that the probability of error to classify a given data X to be of class ω is $1 - p(\omega|X)$.
- Let the classification function be $\omega(X)$. Then

$$\begin{aligned} Pr(E) &= \int f(x, E)dx = \int p(E|x)f(x)dx \\ &= \int (1 - p(\omega(x)|x))f(x)dx = 1 - \int p(\omega(x)|x)f(x)dx. \end{aligned}$$

- The overall probability of error is minimized if $p(\omega(x)|x)$ is maximized for every x. This leads to the maximum a posteriori (MAP) decision rule,

$$\omega^*(x) = \arg\max_{\omega} p(\omega|x).$$

Likelihood-Based Classification

- By the Bayes' rule,

$$p(\omega|x) = \frac{f(x,\omega)}{f(x)} = \frac{f(x|\omega)p(\omega)}{f(x)}.$$

- Since $f(x)$ is constant for fixed x, the MAP decision rule is equivalent to

$$\omega^* = \arg\max_{\omega} \left[\log f(x|\omega) + \log p(\omega)\right].$$

- $f(x|\omega)$ is called the likelihood function of class ω. It can be learned with labeled samples, independently for each class.

Gaussian Models

- A Gaussian model assumes that the likelihood function for class ω is a Gaussian,

$$f(x|\omega) = \frac{1}{(\sqrt{2\pi})^d |\Sigma_\omega|^{1/2}} e^{-\frac{1}{2}(x-\mu_\omega)^T \Sigma_\omega^{-1}(x-\mu_\omega)}.$$

- Ignoring constant terms, the MAP decision rule selects ω^* maximizing

$$g_\omega(x) = -\frac{1}{2}(x-\mu_\omega)^T \Sigma_\omega^{-1}(x-\mu_\omega) - \frac{1}{2}\log|\Sigma_\omega| + \log p(\omega).$$

Parameter Learning from Samples

- Suppose a coin is flipped 1000 times and the head turns up in 600 times. Is there a reason to think the coin is biased (for head)? How about $10/6$?
- Do you believe that the average height of women in Taiwan is higher than 170 cm?
- Questions related to true distributions are asked but we only have data (and our brains).
- Having data is a blessing. All we need to do is to learn the distributions from data.

Maximum Likelihood Estimate

- In MLE we assume a parameterized probability distribution and then find the values of parameters that maximizes the probability of the data.
- Let Θ be the parameter set and D be the data, then

$$\Theta^* = \arg\max_{\Theta} f(D|\Theta).$$

- In certain cases there are closed-form solutions.
  - The relative frequency is the ML estimate of probability of a discrete random variable.
  - For a Gaussian model, the average of samples is the ML estimate for the mean.

EM Algorithm

- In most cases, the ML criterion does not lead to a closed-form solution. We need a numerical recipe to update parameters with higher data likelihoods. The expectation-maximization (EM) algorithm is such a procedure.
- EM is iterative. Each iteration consists of an E-step and an M-step.
  - E-step: compute the expectation value of (log) data likelihood.
  - M-step: find the parameters that maximize the expected data likelihood.

The Q Function

We define the auxiliary function

$$Q(\Theta, \Theta_0) = E[\log p(S, x|\Theta)]$$
$$= \sum_{s=1}^{M} p(s|x, \Theta_0) \log p(s, x|\Theta),$$

where Θ_0 is the current model parameter set, x is data, and S is a discrete hidden variable. Without loss of generality, we assume that the value set of S is $\{1, 2, \ldots, M\}$.

Data Likelihood and Q Function

The log data likelihood and Q function is related by

$$
\begin{aligned}
& Q(\Theta, \Theta_0) - Q(\Theta_0, \Theta_0) \\
&= \sum_{s=1}^{M} [p(s|x, \Theta_0) \log f(s, x|\Theta) - p(s|x, \Theta_0) \log f(s, x|\Theta_0)] \\
&= \sum_{s=1}^{M} p(s|x, \Theta_0) \left[\log f(x|\Theta) + \log p(s|x, \Theta)\right] \\
&\quad - \sum_{s=1}^{M} p(s|x, \Theta_0) \left[\log f(x|\Theta_0) + \log p(s|x, \Theta_0)\right] \\
&= \log f(x|\Theta) - \log f(x|\Theta_0) - \sum_{s=1}^{M} p(s|x, \Theta_0) \log \frac{p(s|x, \Theta_0)}{p(s|x, \Theta)} \\
&= \log f(x|\Theta) - \log f(x|\Theta_0) - D(p_0||p).
\end{aligned}
$$

Non-Decreasing Likelihood

- Suppose Θ^* maximizes $Q(\Theta, \Theta_0)$. We have

$$\begin{aligned}
&\log f(x|\Theta^*) - \log f(x|\Theta_0) \\
&= Q(\Theta^*, \Theta_0) - Q(\Theta_0, \Theta_0) + D(p_0||p) \\
&\geq Q(\Theta^*, \Theta_0) - Q(\Theta_0, \Theta_0) \geq 0,
\end{aligned}$$

  where we use the fact that $D(p||q)$ (called the KL-distance between distributions p and q) is always non-negative.

- Therefore, the data likelihood is non-decreasing with each iteration. It converges to a local maximum.

Batched Data

For a batch of data samples, $\{x_1, x_2, \ldots, x_N\}$, the Q function is

$$Q(\Theta, \Theta_0) = \sum_{i=1}^{N} E \log f(S_i, x_i|\Theta)$$

$$= \sum_{i=1}^{N} \sum_{s_i=1}^{M} p(s_i|x_i, \Theta_0) \log f(s_i, x_i|\Theta).$$

Non-Decreasing of Data Likelihood

To show that increasing the value of the Q function increases the data likelihood, note that

$$\sum_{i=1}^{N} \log f(x_i|\Theta) - \sum_{i=1}^{N} \log f(x_i|\Theta_0)$$

$$= Q(\Theta, \Theta_0) - Q(\Theta_0, \Theta_0) + \sum_{i=1}^{N} D(p_0^i || p^i),$$

where p_0^i is the posterior probability of S given x_i with parameter Θ_0.

Gaussian Mixture Model

- In Gaussian mixture models (GMM), the likelihood function of ω is a weighted sum of Gaussians,

$$f_\omega(x|\Theta) = \sum_{k=1}^{K} p(k|\Theta) f(x|k,\Theta) = \sum_{k=1}^{K} c_k N(x;\mu_k,\sigma_k^2),$$

  where $N(x;\mu,\sigma^2)$ is a Gaussian with mean μ and variance σ^2.
- The parameters in GMM are the mixture weights $c_k's$, the means $\mu_k's$, and the covariances and $\sigma_k's$.

The Q Function for GMM

- We can use EM to estimate these parameters by samples $D = \{x_1, \ldots, x_N\}$ of class ω.
- The Q function is

$$Q = \sum_{i=1}^{N} \sum_{k=1}^{K} p(k|x_i, \Theta_0) \left[\log(p(k|\Theta) + \log f(x_i|k, \Theta)\right]$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} p(k|x_i, \Theta_0) \left[\log c_k + \log N(x_i; \mu_k, \sigma_k^2)\right].$$

- The first term inside the bracket depends only on c_k while the second term depends only on μ_k and σ_k. Therefore they can be maximized separately.

Update Equations

It can be shown that with

$$p_k^i = p(k|x_i, \Theta_0) = \frac{f(x_i, k|\Theta_0)}{f(x_i|\Theta_0)},$$

the update equations are

$$\begin{cases} c_k = \frac{1}{N}\sum_{i=1}^{N} p_k^i, \\ \mu_k = \frac{\sum_{i=1}^{N} p_k^i \; x_i}{\sum_{i=1}^{N} p_k^i}, \\ \sigma_k^2 = \frac{\sum_{i=1}^{N} p_k^i (x_i - \mu_k)^2}{\sum_{i=1}^{N} p_k^i}. \end{cases}$$

Learning GMM Parameters

Refer to Figure 9.1. The samples are generated from a Gaussian mixture with two components.

- For demonstration, bad initial parameters are used.
- After only one iteration of EM, the parameters become close to the “true” values.
- It is remarkable that in this approach, we do not need to know the labels of data. (But we do assume the correct number of components.)