Pattern Recognition

Notes on Spoken Language Processing

Chia-Ping Chen

Department of Computer Science and Engineering

National Sun Yat-Sen University

Kaohsiung, Taiwan ROC

Classification and Recognition

- In a classification problem, we design a system that can decide the class of a test sample.
  - need to determine the set of classes
  - need a function to map data to its class
- The tasks involved in building such a system include feature design, data collection, training, and testing.
- Pattern recognition refers to the generally harder problems where the set of classes is not clearly defined, even infinite.

Class-Related Probability Functions

- Denote the set of classes by $\{\omega_k, k = 1, \ldots, K\}$, and the observed data (features) by x.
- three basic probabilities
  - a priori (or prior) probability $P(\omega_k)$
  - conditional probability $p(x|\omega_k)$
  - a posteriori probability (or posterior) $P(\omega_k|x)$
- By Bayes' rule

$$P(\omega_k|x) = \frac{p(x, \omega_k)}{p(x)} = \frac{p(x|\omega_k)P(\omega_k)}{p(x)}$$

Bayesian Decision Rule

- The probability that x is of class ω_k is $p(\omega_k|x)$.
- An intuitive decision is the Bayes decision rule

$$\begin{aligned} k^* &= \arg\max_k \; P(\omega_k|x) \\ &= \arg\max_k \; p(x|\omega_k)P(\omega_k). \end{aligned}$$

- It is also called the maximum a posteriori (MAP) decision.
- Thus the Bayes decision rule gives the minimum error probability for deciding the class of x.

Minimum Error Classification

- Let the classification function be $\omega(X)$.
- The error rate averaged over all observations is

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

- $Pr(E)$ is minimized if $p(\omega(x)|x)$ is maximized for every x.
- It is obvious that the Bayes decision rule achieves the minimum error rate.

Discriminant Functions

- A classifier can have K discriminant functions, one for each class, such that an instance is classified as ω_j if

$$d_j(x) > d_i(x), \;\; \forall i \neq j.$$

- The posterior probabilities are optimal discriminant functions: they achieve the minimum error rate.
- Yet, functions equivalent to or even approximate to $P(\omega_k|x)$ can be used, often to simplify the classifier.

Likelihood Ratio

- In a two-class classification problem, the Bayes' decision rule is equivalent to comparing

$$p(x|\omega_1)P(\omega_1) \text{ and } p(x|\omega_2)P(\omega_2).$$

- Define the likelihood ratio

$$l(x) = \frac{p(x|\omega_1)}{p(x|\omega_2)},$$

  and use a threshold value, $\frac{P(\omega_2)}{P(\omega_1)}$, for decision.

- Equivalently, the log-likelihood difference can be used.

Gaussian Classifier

- The class-conditional pdf is assumed to be Gaussian.

$$p(\mathbf{x}|\omega_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_k^{-1} (\mathbf{x}-\mu_k)}$$

- A Gaussian classifier has equivalent quadratic discriminant functions: taking the logarithm and ignoring constant of the above,

$$d_k(\mathbf{x}) = \log p(\mathbf{x}|\omega_k) P(\omega_k) = -\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k) + const$$

Linear Discriminant

- If all Gaussians have the same covariance matrix, then the quadratic term $\mathbf{x}^T \Sigma_k^{-1} \mathbf{x}$ is constant and the equivalent discriminant functions become linear

$$d_k(\mathbf{x}) = \mathbf{a}_k^T \mathbf{x} + c_k$$

- For a two-class problem, the decision boundary is a hyperplane.

Estimating Error Rate

- The error rate of a classifier can be estimated from sample data.
- The error rate on the training set can be seen as a lower bound, i.e., an optimistic estimate.
- It is common to use a hold-out set to have a better estimate.
- The hold-out set ought to be disjoint from the training set.

Error Bounds

- Often the true error probability p of a classifier C is unknown.
- Suppose one observes k errors out of n tests. Can he bound p by some (p_1, p_2)?
- We decide p_1 such that at p_1 the probability that C would make at least k errors in n trials is $0.05/2$.
- Similarly, at p_2 the probability that C would make at most k errors in n trials is $0.05/2$.
- (p_1, p_2) is called the 0.95 confidence interval.

Comparing Classifiers

- Suppose classifier A gives the correct answer and classifier B gives the wrong answer for a sample x. Does that mean A is better than B?
- Suppose classifier A gives a lower error rate than classifier B on a test set T. Does that mean A is better than B?
- Statisticians have developed certain tests to answer such questions.

Null Hypothesis

- In comparing two classifiers, the null hypothesis H_0 is that they have the same error rates, i.e.

$$H_0 : p_A = p_B.$$

- Based on the observation of error patterns, one decide whether H_0 can be rejected at the level of significance of 0.05.

McNemar's Test

- Consider N, where
  - N_{00}: no. of tests that A, B are both correct.
  - N_{01}: no. of tests that A is correct, B is wrong.
  - N_{10}: no. of tests that A is wrong, B is correct.
  - N_{11}: no. of tests that A, B are both wrong.
- If H_0 is correct, then out of the $N_{01} + N_{10}$ tests that only one classifier makes error, the number N_{10} that A makes the error follows a binomial distribution $B(n, 1/2)$.
- So testing H_0 is equivalent to testing a distribution.

Discriminative Training

- With MLE or MAP training criteria, only data with class label ω is used to train the parameters in the probability function of that class, $p(\mathbf{x}|\omega)$.
- In discriminative training, we use data of other classes as well to train parameters in a class $p(\mathbf{x}|\omega)$.
- By discriminative training, we want not only to increase the probability of the correct class (data), but also to decrease the probability of the wrong class (data), in updating the parameters.

Conditional Likelihood

- The conditional likelihood is defined by

$$P(\omega|\mathbf{x}) = \frac{p(\mathbf{x}|\omega)P(\omega)}{p(\mathbf{x})}$$

- The maximum conditional likelihood estimator is

$$\Phi_{\text{CMLE}} = \arg\max_{\Phi} p_{\Phi}(\omega|\mathbf{x})$$

- Since $p(\mathbf{x}) = \sum_{\omega'} p(\mathbf{x}|\omega')P(\omega')$,

$$P(\omega|\mathbf{x}) = \frac{1}{1 + \frac{\sum_{\omega' \neq \omega} p(\mathbf{x}|\omega')P(\omega')}{p(\mathbf{x}|\omega)P(\omega)}}.$$

Maximum Mutual Information

- The mutual information between sample $\mathbf{x}$ and its label ω is defined by

$$I(\mathbf{x};\omega) = \log \frac{p(\mathbf{x},\omega)}{p(\mathbf{x})P(\omega)}$$

- The MMI estimator is

$$\Phi_{\text{MMIE}} = \arg\max_{\Phi} I_{\Phi}(\omega;\mathbf{x})$$

- MMIE is equivalent to CMLE if equal prior is assumed.

Minimum Classification Error

- Define error functions which are related to the discriminant functions of a sample $(\mathbf{x}, \omega_k)$

$$e_k(\mathbf{x}) = -d_k(\mathbf{x}, \Phi) + \left[\frac{1}{K-1} \sum_{j \neq k} d_j(\mathbf{x}, \Phi)^{\eta} \right]^{1/\eta}$$

- Define loss functions

$$l_k(\mathbf{x}; \Phi) = sigmoid(e_k(\mathbf{x}))$$

- The total loss on the training data set T is

$$L(\Phi) = \sum_{\mathbf{x} \in T} l(\mathbf{x}; \Phi) = \sum_{\mathbf{x} \in T} \sum_{k=1}^{K} l_k(\mathbf{x}; \Phi) \delta(\omega, \omega_k)$$

Unsupervised Learning

- In some cases, the class label ω for sample $\mathbf{x}$ is unknown.
- Such data is sometimes called incomplete data.
- Learning from unlabeled data is called unsupervised learning.

Vector Quantization

- The first unsupervised learning method we describe is the vector quantization.
- We want to represent data with prototype vectors, a.k.a. codewords.
- The set of codewords is called the codebook.
- A data point is represented by its closest codeword.
- The main problem here is the distortion (distance) measure and the codebook generation.

K-Means Algorithm

- an iterative algorithm
- initialized by a heuristic set of codewords
- iteratively relabels the data points to their nearest codewords and recomputes the codeword as the centroid of the data points related to a codeword, until a convergence is achieved
- commonly used for codebook generation, as well as clustering

LBG Algorithm

- LBG stands for Linde, Buzo and Gray
- first computes a 1-vector codebook.
- uses a splitting algorithm to obtain a new 2-vector codebook, and apply K-means
- iteratively split codewords and apply K-means
- continue until the number of codewords is achieved

EM Algorithm

- Use the auxiliary function $Q(\Phi, \bar{\Phi})$, which is the expected log data-likelihood,

$$Q(\Phi, \bar{\Phi}) = \sum_{i=1}^{N} \sum_{x_i \in \Omega_x} P(x_i | y_i, \Phi) \log p(x_i, y_i | \bar{\Phi})$$

- Choose an initial estimate Φ.
- Decide $\hat{\Phi}$ that maximizes $Q(\Phi, \bar{\Phi})$ with respect to $\bar{\Phi}$.
- Set $\Phi = \hat{\Phi}$.
- Repeat until convergence.

CART

- CART = classification and regression tree
- a tree (often binary)
- a question is associated with a non-leaf node
- a data point is classified according to the answers to the questions
- closely related to the decision trees, but the questions are decided by data
- can handle high-dimensional data

Choice of Question Set

- standard set of questions
  - simple/singleton questions: each question is about just one variable, say x_d
  - If x_d is discrete, then the question assumes the form: Is $x_d \in C$?, where C is any subset of the set of possible values.
  - If x_d is continuous, then the question is $x_d \leq c$. Candidate c can be decided from data.

Tree and Node Entropy

- By splitting data in a node, we want the data in the resultant nodes to be as pure as possible.
- This can be quantified by entropy.
- The entropy of a node n is

$$H(n) = -\sum_k p(\omega_k|n) \log p(\omega_k|n)$$

- The entropy of a tree T is

$$\bar{H}(T) = \sum_{n \text{ is a leaf node}} \bar{H}(n) = \sum_n H(n)p(n).$$

Splitting Node

- The entropy is non-increasing when a node is splitted.
- We want to select a question such that the reduction in entropy is maximized

$$\Delta \bar{H}_n(q) = \bar{H}(n) - \sum_i \bar{H}(n_i; q),$$

$$q^* = \arg\max_q \Delta \bar{H}_n(q),$$

- The tree entropy is minimized as a result.

Stopping Criteria

- A node is “terminal” if
  - no more splitting is possible; all nodes are “pure”.
  - the optimal entropy reduction is below a threshold.
  - the number of data samples in a child node would be below some threshold.
- The tree growing algorithm stops if all leaf nodes are terminal.

A Few Comments

- For training continuous pdf in child nodes, the likelihood gain is used, as there is no straightforward measure for entropy.
- The least squared error can also be used in node splitting: find the optimal reduction in weighted squared error.
- Complex questions can be composed by clustering terminal nodes obtained by simple questions.
- Use independent test samples or cross validation for pruning.