DISCRIMINATIVE LEARNING AND INFORMATIVE LEARNING IN PATTERN RECOGNITION

Xuechuan Wang, Kuldip K. Paliwal

School of Microelectronic Engineering
Griffith University, Brisbane, QLD 4111, Australia

Phone: +61 7 3875 3754

E-mail: wang@me.gu.edu.au, K.Paliwal@me.gu.edu.au

Web: spl.me.gu.edu.au

ABSTRACT

In pattern recognition, the goal of classification can be achieved from two different types of learning strategy – discriminative learning and informative learning. Discriminative learning focuses on extracting the discriminative information between classes. Informative learning emphasizes on the learning of the class information such as class densities. In this paper we review major discriminative learning methods, namely, principal component analysis (PCA), linear discriminant analysis (LDA), minimum classification error (MCE) training algorithm and Support Vector Machine (SVM) and one informative learning method – Gaussian mixture models (GMM). We also discuss the combination of the two types of learning and give the corresponding experiments results.

1. INTRODUCTION

The goal of pattern classification is to minimize the misclassification or the expected cost of misclassification [1]. The criteria for minimizing the misclassification vary in different learning strategies. These learning strategies can be generally segmented into two groups by their focuses on the class information. One is discriminative learning and the other is informative learning.

Discriminative learning approaches focus on the information difference between classes. Basically, these methods require simultaneous consideration of all classes. The most popular criteria for discriminative learning are principal component criterion and discriminant criterion, which correspond to PCA [2] and LDA [3], respectively. Both PCA and LDA, however, model classes in two separate steps: first projecting the original features into a new subspace by discriminative learning criterion, then finding the class models by a separate misclassification criterion. This may cause serious problems to pattern recognition systems. To

mend this drawback, several integrated algorithms are proposed, such as MCE training algorithm [4] and SVM [5].

Informative learning approaches focus on the class information such as densities. Classification is done by assigning the features to the most likely class. The samples of classes are often assumed to be identically and independently distributed (iid). Thus the class densities are considered separately from each other. Popular examples include Gaussian mixture models (GMM) [6] and Hidden Markov Models (HMM) [7].

The two approaches emphasize on the different aspects of classes information. Generally speaking, discriminative learning provides a framework including feature extraction and classification process, which makes the combination of the two types of approaches possible. This paper reviews major discriminative learning approaches, which include LDA, PCA, MCE training algorithm and SVM, and GMM as the informative learning method. The combination of discriminative learning approaches and informative approaches is also studied.

2. DISCRIMINATIVE LEARNING

2.1. Linear Discriminant Analysis

Suppose we have K classes, $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_K$. Let the i th observation vector from the $\mathcal{X}_j$ be x_{ji} , where $j = 1, \dots, J$ and $i = 1, \dots, N_j$. J is the number of classes and N_j is the number of observations from class j . The within-class covariance matrix S_W and between-class covariance matrix are defined by:

$$S_W = \sum_{j=1}^K \sum_{i=1}^{N_j} (x_{ji} - \mu_j)(x_{ji} - \mu_j)^T \quad (1)$$

and

$$S_B = \sum_{j=1}^K N_j (\mu_j - \mu)(\mu_j - \mu)^T \quad (2)$$

where μ_j is the mean of class j and μ is the overall mean. LDA chooses a linear transformation matrix t that maximizes the objective function

$$J(T) = \frac{|T^T S_B T|}{|T^T S_W T|} \quad (3)$$

It can be shown that the solution of Eq. (3) is that the i th column of an optimal W is the eigenvector corresponding to the i th largest eigenvalue of matrix $S_W^{-1} S_B$.

2.2. Principal Component Analysis

PCA is based on the assumption that most information about the classes is contained in the directions along which the variations are the largest [2]. The axes along these directions are called principal axes. For a given p -dimensional data set $\mathcal{X}$, the m principal axes $T_1, T_2, \dots, T_m$, where $1 \leq m \leq p$, are orthonormal axes onto which the retained variance is maximum in the projected space. Generally, $T_1, T_2, \dots, T_m$ can be given by the m leading eigenvectors of the sample covariance matrix $S = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^T (x_i - \mu)$, where $x_i \in \mathcal{X}$, μ is the sample mean and N is the number of samples, so that:

$$S T_i = \lambda_i T_i \quad i \in 1, \dots, m \quad (4)$$

where λ_i is the i th largest eigenvalue of S . The m principal components of a given observation vector $x \in \mathcal{X}$ are given by:

$$y = [y_1, \dots, y_m] = [T_1^T x, \dots, T_m^T x] = T^T x \quad (5)$$

The m principal components of x are then uncorrelated in the projected space. In multi-class problems, the variations of data are determined on a global basis, that is, the principal axes are derived from a global covariance matrix:

$$\hat{S} = \frac{1}{N} \sum_{j=1}^K \sum_{i=1}^{N_j} (x_{ji} - \hat{\mu})(x_{ji} - \hat{\mu})^T \quad (6)$$

where $\hat{\mu}$ is the global mean of all the samples, K is the number of classes, N_j is the number of samples in class j , $N = \sum_{j=1}^K N_j$ and x_{ji} represents the i th observation from class j . The principal axes $T_1, T_2, \dots, T_m$ are therefore the m leading eigenvectors of $\hat{S}$:

$$\hat{S} T_i = \hat{\lambda}_i T_i \quad i \in 1, \dots, m \quad (7)$$

where $\hat{\lambda}_i$ is the i th largest eigenvalue of $\hat{S}$.

2.3. MCE Training Algorithm

Consider an input vector x and a transformation T , let $y = T x$ be the feature vector in the feature space $\mathcal{F}$. The classifier makes its decision in $\mathcal{F}$ by the following decision rule:

$$x \in \text{Class } k \text{ if } g_k(y, \Lambda) = \max_{\text{for all } i \in K} g_i(y, \Lambda) \quad (8)$$

where $g_i(y, \Lambda)$ is discriminant function of y to class i , Λ is the parameter set and K is the number of classes. A differentiable misclassification measure can be formulated from the decision rule as:

$$d_k(y, \Lambda) = \frac{[\frac{1}{N-1} \sum_{\text{for all } i \neq k} (g_i(y, \Lambda))^\eta]^{1/\eta}}{g_k(y, \Lambda)} \quad (9)$$

where η is a positive number and $g_k(x, \Lambda)$ is the discriminant of observation x to its known class k . When η approaches ∞ , misclassification measure reduces to:

$$d_k(y, \Lambda) = \frac{g_j(y, \Lambda)}{g_k(y, \Lambda)} \quad (10)$$

where class j has the largest discriminant value among all the classes other than class k . The misclassification measure in Eq. (10) is not suitable for direct minimization yet. A loss function which employs the sigmoid function, is defined to smooth the misclassification measure. Expressing explicitly in x and T , loss function is given as:

$$l_k(Tx, \Lambda) = f(d_k(Tx, \Lambda)) = \frac{1}{1 + e^{-\xi d_k(Tx, \Lambda)}} \quad (11)$$

where $\xi > 0$. For a training set $\mathcal{X}$, the empirical loss is:

$$L(\Lambda) = E\{l_k(Tx, \Lambda)\} = \sum_{k=1}^K \sum_{i=1}^{N_k} l_k(Tx^{(i)}, \Lambda) \quad (12)$$

where N_k is the number of samples in class k . The class parameter set Λ and transformation matrix T is optimized by minimizing the loss function through the steepest gradient descent algorithm. This is an iterative algorithm and the iteration rules are:

$$\begin{aligned} \Lambda_{t+1} &= \Lambda_t - \epsilon \nabla L(\Lambda)|_{\Lambda=\Lambda_t} \\ T_{sq}(t+1) &= T_{sq}(t) - \epsilon \frac{\partial L}{\partial T_{sq}} \Big|_{T_{sq}=T_{sq}(t)} \\ \nabla L(\Lambda) &= \begin{bmatrix} \partial L / \partial \lambda_1 \\ \vdots \\ \partial L / \partial \lambda_m \end{bmatrix} \end{aligned} \quad (13)$$

where t denotes t th iteration, $\lambda_1, \dots, \lambda_m \in \Lambda$ are class parameters, $\epsilon > 0$ is the adaption constant and s and q are the row and column indicators of T .

2.4. Support Vector Machine

Considering a two-class case, suppose the two classes are ω_1 and ω_2 and we have a training set $\mathcal{X} = \{x_1, \dots, x_N\} \subset \mathcal{R}^p$. The training data are labelled as:

$$y_i = \begin{cases} +1 & x_i \in \omega_1 \\ -1 & x_i \in \omega_2 \end{cases} \quad (14)$$

SVM first maps the training data into a high dimensional feature space $\mathcal{F}$ through a non-linear mapping $\Phi : \mathcal{R}^p \rightarrow \mathcal{F}$, where $\mathcal{R}^p$ is the sample space. Then a linear function in $\mathcal{F}$ is computed as:

$$f(x) = (w \cdot \Phi(x)) + b \quad (15)$$

where $(\cdot)$ denotes the dot product. Ideally, all the data in these two classes satisfy the following constraints [5]:

$$\begin{aligned} (w \cdot \Phi(x_i)) + b &\geq +1 \quad \text{for } y_i = +1 \\ (w \cdot \Phi(x_i)) + b &\leq -1 \quad \text{for } y_i = -1 \end{aligned} \quad (16)$$

These two inequalities can be combined into one:

$$y_i(w \cdot \Phi(x_i)) + b - 1 \geq 0 \quad \forall i \quad (17)$$

Considering the points $\Phi(x_i)$ in $\mathcal{F}$ for which the equality in (16) holds, these points lie on two hyper-planes $H_1 : (w \cdot \Phi(x_i)) + b = +1$ and $H_2 : (w \cdot \Phi(x_i)) + b = -1$. These two hyper-planes are parallel and no training points fall between them. The margin between them is $\frac{2}{\|w\|}$. Therefore we can find a pair of hyper-planes with maximum margin by minimizing $\|w\|^2$ subject to (17)[5]. This problem can be written as a convex optimization problem:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|w\|^2 \\ &\text{subject to} \quad y_i(w \cdot \Phi(x_i)) + b - 1 \geq 0 \quad \forall i \end{aligned} \quad (18)$$

where the first function is *primal* objective function and the second function is the corresponding constraints. Eq. (18) can be solved by constructing a Lagrange function. Introducing positive Lagrange multipliers $\alpha_i, i = 1, \dots, N$, one for each constraint in Eq.(18), the Lagrange function is given by:

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot \Phi(x_i) + b) + \sum_{i=1}^N \alpha_i \quad (19)$$

By defining kernel function $k(x_i, x) = (\Phi(x_i) \cdot \Phi(x))$ in $\mathcal{F}$ and solving Eq. (19), we obtain the dual optimization problem:

$$\begin{aligned} &\text{maximize} \quad -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k_{ij} + \sum_{i=1}^N \alpha_i \\ &\text{subject to} \quad \sum_{i=1}^N \alpha_i y_i = 0 \\ &\quad \alpha_i \geq 0 \quad \forall i \end{aligned} \quad (20)$$

This is a quadratic programming problem. [8] has a complete description on solving quadratic programming problems.

3. INFORMATIVE LEARNING

GMM is a popular informative approach. Consider a continuous random vector $x \in \mathcal{R}^d$ and a training set $\mathcal{X}$ of iid.

We choose the Gaussian mixtures as an estimation of the probability density of x :

$$p(x|\Theta) = \sum_{i=1}^M c_i p(x|\Theta_i) \quad (21)$$

where M is the number of mixtures, $c_i, i = 1, \dots, M$ are mixture coefficients under constraints $c_i \geq 0$ and $\sum_{i=1}^M c_i = 1$. $p(x|\Theta_i)$ are multivariate normal densities:

$$p(x|\Theta_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right] \quad (22)$$

where Σ_i and μ_i are i th densities parameters. We can formulate the log-likelihood over training set $\mathcal{X}$ as:

$$l(\Theta) = \left[\prod_{j=1}^N p(x^j|\Theta) \right] = \sum_{j=1}^N \log \sum_{i=1}^M c_i p(x^j|\Theta_i) \quad (23)$$

Maximum likelihood estimation of the parameter set $\hat{\Theta} = \{\hat{c}_i, \hat{\mu}_i, \hat{\Sigma}_i\}$ can be efficiently computed with the EM algorithm [6]. EM algorithm involves two iterative steps:

In the E-step, the *a posteriori* probability of k th density responsible for the generation of j th sample is estimated as:

$$p(k|x^j, \mu_k, \Sigma_k) = \frac{c_k p(x^j|k, \mu_k, \Sigma_k)}{\sum_{i=1}^M c_i p(x^j|i, \mu_i, \Sigma_i)} \quad (24)$$

In the M-step, the parameters of k th density are estimated as:

$$\hat{c}_k = \sum_{j=1}^N p(k|x^j, \mu_k, \Sigma_k) \quad (25)$$

$$\hat{\mu}_k = \frac{\sum_{j=1}^N p(k|x^j, \mu_k, \Sigma_k) x^j}{\sum_{j=1}^N p(k|x^j, \mu_k, \Sigma_k)} \quad (26)$$

$$\hat{\Sigma}_k = \frac{\sum_{j=1}^N p(k|x^j, \mu_k, \Sigma_k) (x^j - \hat{\mu}_k)(x^j - \hat{\mu}_k)^T}{\sum_{j=1}^N p(k|x^j, \mu_k, \Sigma_k)} \quad (27)$$

4. EXPERIMENTS

An evaluation of these approaches was made on Deterding vowel database [9], which has 11 vowel classes as shown in the Table 1. Each of these 11 vowels are uttered 6 times by 15 different speakers. This gives a total of 990 vowel tokens. A central frame of speech signal is excised from each of these 990 vowel tokens. A 10th order linear prediction analysis is carried out for each frame resulting in 10 log-area parameters. These 10 parameters define the original 10 dimensional feature space. 528 frames from the eight speakers are used to train the models and 462 frames from the seven speakers are used to test the models.

vowel	word	vowel	word	vowel	word
i	heed	O	hod	I	hid
E	head	U	hood	A	had
a:	hard	3:	heard	Y	hud
C:	hoard	u:	who'd		

Table 1: Vowels and words used in Deterding database.

Eight approaches are employed in the experiment. They are LDA+Distance classifier, PCA+distance classifier, MCE training algorithm, SVM(Polynomial kernel), SVM(RBF kernel), GMM, LDA+GMM classifier and MCE+GMM classifier. The last two approaches combine the discriminative learning approaches, namely LDA and MCE training algorithm with informative learning approach, GMM, together. Table 2 shows the recognition rate of each approach.

Algorithms	Training	Testing
LDA+Distance classifier	93.75%	51.95%
PCA+distance classifier	92.42%	49.13%
MCE training algorithm	98.11%	53.68%
SVM(Polynomial kernel)	88.45%	57.36%
SVM(RBF kernel)	90.72%	58.44%
GMM	91.86%	54.11%
LDA+GMM classifier	94.70%	54.76%
MCE+GMM classifier	97.16%	60.17%

Table 2: Recognition rate on Deterding database.

5. CONCLUSIONS

The experiments results show that integrated discriminative learning approaches, such as MCE training and SVM perform better than non-integrated approaches, such as LDA and PCA. GMM, as an informative learning approach, performs better on testing data than LDA, PCA and MCE training algorithm when these discriminative learning approaches employ simple distance classifier. But GMM does not perform as well as SVM. As expected, when combining the discriminative approach and informative approach together, the classifier's performance is improved significantly.

6. REFERENCES

- [1] Y.D. Rubinstein and T. Hastie, Discriminative vs informative learning, *Proceedings Third International Conference on Knowledge Discovery and Data Mining*, pp. 49–53, 1997
- [2] I.T. Jolliffe, *Principal component analysis*, Springer-Verlag, New York, 1986
- [3] R.O.Duda and P.E.Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons Press, New York, 1973.
- [4] B.H. Juang and S. Katagiri, Discriminative Learning for Minimum Error Classification, *IEEE Transactions on Signal Processing*, Vol. 40, No. 12, December, 1992.
- [5] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, 2(2):955-974, 1998.
- [6] A.P. Dempster, N.M. Laird, and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society, Series B*, 39(1):1–38, 1977
- [7] L. Rabiner and B. Juang, *Fundamentals of speech recognition*, Prentice Hall, New York, 1993
- [8] R.J. Vanderbei, LOQO: An interior point code for quadratic programming, *Optimization Methods and Software*, vol. 11, pp. 451-484, 1999.
- [9] D.H. Deterding, Speaker Normalisation for Automatic Speech Recognition, Ph.D. Thesis, University of Cambridge, 1989