第 30 卷 第 3 期
2004 年 5 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 30, No. 3
May, 2004

A Novel PNN Classification for Speaker Identification[1)]

WANG Cheng-Ru WANG Jin-Jia LIAN Qiu-Sheng

(*Department of Communication and Electronic Engineering, Yanshan University, Qinghuangdao* 066004)

(E-mail: 01016888@sina.com)

Abstract A novel σ PNN model is proposed for class conditional density estimation based on the mixtures of PNN of shared pattern layers and PNN of separated pattern layers. Each class not only has a set of pattern layers belonging to itself, but also has several pattern layers shared for all class, where "shared" means that each kernel may contribute to the estimation of the conditional density of all classes. The training of the novel model utilizes the maximum likelihood criterion and an effective EM algorithms to adjust model parameters is developed. These results of the closed-set text-independent speaker identification experiments indicate the proposed model and algorithms improve identification accuracy.

Key words Probabilistic neural networks, maximum likelihood, expectation-maximization, speaker identification

1 Introduction

Speaker recognition is to recognize the speaker by his speech signal and his feature parameters extracted in advance. The task of closed-set text-independent speaker identification is to identify from a set of known speakers the speaker who is the most closely related to the sample of the tested speech. The speaker identification system usually includes two parts: the feature parameters extraction and the classifier[1]. Because of the research of psychological and physiological acoustics in the aspect of speaker recognition mechanism, it is still difficult to find out the speaker's speech features which have long-term validity and can be used in different channels and different noise environment. After extracting non-exclusive feature parameters, the design of the classifier becomes the focal point for the purpose of improving the correction rate of the classification. The recognition rate of dynamic time warping (DTW) is high, but it requires extensive storage and computation. As a benchmark classifier, vector quantization (VQ) is often considered having lost the time sequence information of the speech. Though the hidden Markov model (HMM) is applied widely, the hypothesis of its output independence is not reasonable. The Gaussian mixture model (GMM) is applied successfully in speaker recognition, but it requires a large number of speech samples and noises make its performance deteriorate sharply. The artificial neural networks (ANNs) classifier provides a novel approach to speaker recognition. In this paper, in order to avoid the problem of overtime-training for ANNs, the probabilistic neural network (PNN) is used as the core classifier for the speaker recognition system.

It is well known that a PNN is used as a classifier due to its good generalization ability and fast learning capability, case of on-line updating, and sound statistical foundation in Bayesian estimation theory. PNN has become an effective tool for solving many difficult classification problems in practice. The PNN classifier was used for optical character recognition in [2], satellite cloud classification in [3], and speaker identification in [4]. The original PNN is a direct neural-network implementation based on Pazer nonparametric

1) Supported by National Key Laboratory of Vision and Hearing Signal Processing of Peking University
Received March 20, 2003; in revised form October 3, 2003
收稿日期 2003-03-20; 收修改稿日期 2003-10-03

probability density function estimation and Bayesian classification rule. It consists of input layer, pattern layer and summation layer. The main drawback of the original PNN is its large network structure, since every training pattern has to be stored. One natural idea to simplify the PNN is to reduce the number of pattern layer neurons. In [6], Streit *et al.* improved the PNN by using finite Gaussian mixture models and maximum likelihood (ML) training scheme. However, the ML-based training does not necessarily lead to optimization, and there are also a larger number of model parameters. It is more difficult that the number of the Gaussian components has to be decided by experimentation. Other schemes such as learning vector quantization and vector quantization reduction, have also been proposed for clustering the training samples[7, 8].

First, we propose a novel scheme to reduce the number of pattern layer neurons. In the pattern layer, the traditional separated model is changed into the shared model. That is the class conditional probability density of summary layer is expressed by the kernels of the shared pattern layer. Each kernel contributes to the estimation of the conditional density of all classes. This novel model is called shared patter layer based PNN (SPNN).

Secondly, we further extend the SPNN model and develop a mixture, called σ PNN model, based on the PNN and SPNN models. The special parameter σ adds constraints to the model parameters in order to adjust shared and separated kernel parameters among classes. The training of the novel scheme still use the maximum likelihood estimation; expectation-maximization (EM) approach can be used to adjust the model parameters.

2 The SPNN model

Suppose a d-dimensional input speech feature vector $\boldsymbol{x}$ belongs to one of the S speakers. The best classifier is given by the fundamental Bayesian decision rule

$$C(\boldsymbol{x}) = \arg\max_{\lambda_k} P(\lambda_k) p(\boldsymbol{x} \mid \lambda_k) \tag{1}$$

where a priori class probability $P(\lambda_k)$, $k=1,2,\cdots,S$ are assumed to be of a uniform distribution and equal to each other; λ_k denotes the parameters of the kth speaker. Suppose that the class conditional probability density function $p(\boldsymbol{x}|\lambda_k)$ can be represented by a Gaussian mixture model, *i. e*,

$$p(\boldsymbol{x} \mid \lambda_k) = \sum_{i=1}^{M_k} \omega_{ik} p_{ik}(\boldsymbol{x}; \mu_{ik}, \Sigma_{ik}) \tag{2}$$

where M_k is the number of Gaussian components of the kth speaker, ω_{ik}'s are the weights of the Gaussian components which satisfy the constraint $\sum_{i=1}^{M_k} \omega_{ik} = 1$, $p_{ik}(\boldsymbol{x}; \mu_{ik}, \Sigma_{ik})$ denotes the multivariate Gaussian density function of the ith component of the kth speaker, and each Gaussian component is parameterized by means of vector μ_{ik} and covariance matrix Σ_{ik}. The GMM can be easily mapped to the PNN structure and the resultant PNN needs much fewer neurons than the traditional PNN. The parameter sets $\lambda_k = \{\omega_{ik}, \mu_{ik}, \Sigma_{ik}\}$ of the PNN model for each speaker need to be estimated from the training data set.

The class conditional probability density function $p(\boldsymbol{x}|\lambda_k)$ is modeled by the following special GMM.

$$p(\boldsymbol{x} \mid \lambda_k) = \sum_{i=1}^{M} \omega_{ik} p_{ik}(\boldsymbol{x}; \mu_i, \Sigma_i) \tag{3}$$

where M denotes the number of Gaussian components shared by all speakers; ω_{ik}'s are the weights of the Gaussian components which satisfy the constraint $\sum_{i=1}^{M} \omega_{ik} = 1$; $p_{ik}(\boldsymbol{x}; \mu_i, \Sigma_i)$ denotes the multivariate Gaussian density function of the ith component of the kth speaker, *i. e.*,

$$p_{ik}(\boldsymbol{x},\mu_i,\Sigma_i)=\frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}\cdot\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\mu_i)^{\mathrm{T}}\Sigma_i^{-1}(\boldsymbol{x}-\mu_i)\right\} \tag{4}$$

where μ_i denotes the d-dimensional mean vector of the ith Gaussian component; Σ_i denotes the $d\times d$ covariance matrix of the ith Gaussian component. The special GMM can be also easily mapped to the shared pattern layer PNN structure and the resultant PNN will need much fewer neurons than the traditional PNN. The parameter sets of the PNN model for each speaker need to be estimated from the training data set. In this paper, the novel model is called SPNN, which is a special PNN because of its shared pattern layer. The first layer just distributes the input to the neurons in the pattern layer. All the input feature vectors $\boldsymbol{x}$ form the feature space set T. The second layer shares the same kernels. The exclusive pattern layer pool consists of M kernels expressing probability density function, and its connective weights denote the priori probability. The output of the third layer corresponds to each class conditional probability density. All the adjusted parameters of the SPNN model can be denoted as $\lambda_k=\{\omega_{ik},\mu_i,\Sigma_i\}$. The training approach of the model parameters utilizes ML estimation. The well-known EM approach can be used to solve the question of maximizing the log-likelihood function. The reader can refer to [9] for the detail treatment.

3 The σ PNN model and its EM training

For the network structure with the same number of kernels of the pattern layer, if the data of different classes are highly overlapped, the SPNN model may utilize the kernels more efficiently than the PNN model. In the same way, if the data of the same class are very distinctly distributed to different regions, the PNN model may describe the data feature space more properly than the SPNN model. The similar character is presented in the radial basis function (RBF) network. The advantage and disadvantage of the two models depend on the data's distribution. So the combination of these two models can be applied to solving the difficult classification problems in practice. We hope that its network structure is more logical, the data's real distribution is more valid, and so is the recognition ability. The model is called the σ PNN model where adjustable parameter σ plays an important role in controlling the proportions between the SPNN and PNN models and satisfies the constraint $\sigma\geqslant 0$.

As usual, the M kernels of the model are partitioned into two classes L_s and L_p which represent the kernel number of the shared and separated pattern layer respectively and satisfy the constraint $L_s+L_p=M$. Meanwhile, according to the speaker number, L_p is partitioned into S groups and satisfies the constraint $\sum_{k=1}^{S}L_{pk}=L_p$. We hope that the classification error rate can be reduced as less as possible, while the L_s number can be as large as possible, that is, the network architecture is minimized. We also hope that L_{pk} will fully contribute to the density estimation of speaker k, and be neglected to the density estimation of other speakers. According to these considerations we introduce the following function

$$p_\sigma(\boldsymbol{x}\mid\lambda_k)=\sum_{i\in L_s}\omega_{ik}p_{ik}(\boldsymbol{x};\mu_i,\Sigma_i)+\sigma\sum_{i\in L_{pk}}\omega_{ik}p_{ik}(\boldsymbol{x};\mu_{ik},\Sigma_{ik}) \tag{5}$$

where ω_{ik}'s denote the weights of Gaussian components and satisfy the constraint $\sum_i\omega_{ik}=1$. Obviously, the function $p_\sigma(\boldsymbol{x}|\lambda_k)$ is not a probability density since its value is less than 1. The function is only defined for training purpose and must be distinguished from the class conditional density $p(\boldsymbol{x}|\lambda_k,\sigma)$, which is the output of the summation layer of the σ PNN model(after training). The function $p(\boldsymbol{x}|\lambda_k,\sigma)$ is computed by (2) or (3). The parameter

σ is only involved in the training procedure. In the training process, what we used is not parameter σ but σ's function β (for definition of β refer to formulae (16(a))), which decides the mixture model to approximate either the PNN model or SPNN model.

All the feature space of the σ PNN model can be expressed as $\Lambda=\{\lambda_k\}_{k=1}^{S}$, where λ_k denotes the kth speaker's model parameters set. There are many available approaches to train the σ PNN model, such as ML estimation. Now suppose that the training samples are drawn independently from the feature space set T, and they are further separated into S subsets T_i, $i=1,\cdots,S$, which belong to S speakers, respectively. For the computational efficiency, generally we will maximize the equivalent log-likelihood for the ML estimation of the parameters set

$$\Lambda^* = \arg\max_{\Lambda}\sum_{k=1}^{S}\sum_{x\in T_k}\log[p_\sigma(\boldsymbol{x}\mid\lambda_k)] \tag{6}$$

The maximization can be viewed as a nonlinearized optimization problem. Fortunately, the EM approach[9] can be used to solve this problem. The EM approach can be utilized to achieve the maximum-likelihood estimation via iterative computation when the observations are viewed as incomplete data. There are two major steps in this approach: the expectation (E) step and maximization (M) step. The E step extends the likelihood function to the unobserved variables, and then computes an expected value of the complete data log-likelihood function with respect to them using the current estimate of the parameters set. In the M step, the new parameter set is obtained by maximizing the resultant expectation function. These two steps are iterated until convergence is reached. In order to apply the EM approach to the σ PNN model, we must first present the unobserved variable, which is a random variable indicating the kernel that generates the observation variable. If the feature $\boldsymbol{x}$ comes from the ith Gaussian component of the kth speaker, $z_{i|k}(\boldsymbol{x})=1$, otherwise $z_{i|k}(\boldsymbol{x})=0$. All $z_{i|k}$ belong to the set Z. The observation feature set T is generally called the incomplete data set while the set $Y=\{T,Z\}$ is called the complete data set, since the missing information has been added to. So we compute the log-likelihood of the complete data set as

$$L(Y\mid\Lambda)=\sum_{k=1}^{S}\sum_{x\in T_k}\log p_\sigma(\boldsymbol{y}\mid\lambda_k)=$$
$$\sum_{k=1}^{S}\sum_{x\in T_k}\log\Big(\sum_{i=1}^{M}z_{i|k}(\boldsymbol{x})p_\sigma(\boldsymbol{x}\mid z_{i|k}(\boldsymbol{x})=1,\lambda_k)p(z_{i|k}(\boldsymbol{x})=1)\Big) \tag{7}$$

For the sake of simplicity of calculation, PNN is viewed as a special case of SPNN that some kernels are only associated with one speaker. By setting all the prior probabilities of a kernel equal to zero, the connection with other speakers are separated. Using (2), (3) and (5), we have

$$L(Y\mid\Lambda)=$$
$$\sum_{k=1}^{S}\sum_{x\in T_k}\Big\{\sum_{i\in L_s}z_{i|k}(\boldsymbol{x})\log p(\boldsymbol{x},\mu_i,\Sigma_i)\omega_{ik}+\sum_{i\in L_{pk}}z_{i|k}(\boldsymbol{x})\log p(\boldsymbol{x},\mu_{ik},\Sigma_{ik})\omega_{ik}\sigma\Big\}=$$
$$\sum_{k=1}^{S}\sum_{x\in T_k}\Big\{\sum_{i\in L_s}z_{i|k}(\boldsymbol{x})\log p(\boldsymbol{x},\mu_i,\Sigma_i)\omega_{ik}+\sum_{i\in L_{pk}}z_{i|k}(\boldsymbol{x})\log p(\boldsymbol{x},\mu_i,\Sigma_i)\omega_{ik}\sigma\Big\} \tag{8}$$

In E-step, we take the expectation of the complete data log-likelihood function based on current parameter set Λ^{old} and observation set T, *i.e.*,

$$Q(\Lambda\mid\Lambda^{\text{old}})=$$
$$E(L(Y\mid\Lambda)\mid T,\Lambda^{\text{old}})=\sum_{k=1}^{S}\sum_{x\in T_k}\Big\{\sum_{i\in L_s}E(z_{i|k}(\boldsymbol{x})\mid T,\Lambda^{\text{old}})\log p(\boldsymbol{x},\mu_i,\Sigma_i)\omega_{ik}+$$

$$\sum_{i\in L_{pk}} E(z_{i|k}(\boldsymbol{x}) \mid T, \Lambda^{\text{old}}) \log p(\boldsymbol{x}, \mu_i, \Sigma_i)\omega_{ik} \Big\} \tag{9}$$

According to the definition of $z_{j|i}(\boldsymbol{x})$, we can calculate

$$E(z_{i|k}(\boldsymbol{x}) \mid T, \Lambda^{\text{old}}) = P_\sigma(z_{i|k}(\boldsymbol{x}) = 1 \mid T, \Lambda^{\text{old}}) = \frac{p_\sigma(\boldsymbol{x} \mid z_{i|k}(\boldsymbol{x}) = 1, \lambda_i^{\text{old}}) p_\sigma(z_{i|k}(\boldsymbol{x}) = 1, \lambda_i^{\text{old}})}{p_\sigma(\boldsymbol{x}, \lambda_i^{\text{old}})} =$$

$$\begin{cases} \dfrac{\omega_{ik}^{\text{old}} p^{\text{old}}(\boldsymbol{x}, \mu_i, \Sigma_i)}{p_\sigma^{\text{old}}(\boldsymbol{x} \mid \lambda_k)} = E_s(\boldsymbol{x}), & i \in L_s \\ \dfrac{\sigma\omega_{ik}^{\text{old}} p^{\text{old}}(\boldsymbol{x}, \mu_i, \Sigma_i)}{p_\sigma^{\text{old}}(\boldsymbol{x} \mid \lambda_k)} = \sigma E_{pk}(\boldsymbol{x}), & i \in L_{pk} \end{cases} \tag{10}$$

and the constraint $\sum_{i=1}^{M} E(z_{i|k}(\boldsymbol{x}) \mid T, \Lambda^{\text{old}}) = \sum_{i\in L_s} E_s(\boldsymbol{x}) + \sigma \sum_{i\in L_{pk}} E_{pk}(\boldsymbol{x}) = 1$ is satisfied.

In M-step, the new parameter set Λ is obtained by maximizing the resultant expectation function $Q(\Lambda \mid \Lambda^{\text{old}})$. If we write the function $Q(\Lambda \mid \Lambda^{\text{old}})$ as $Q_1(\Lambda \mid \Lambda^{\text{old}}) + Q_2(\Lambda \mid \Lambda^{\text{old}})$ where

$$Q_1(\Lambda \mid \Lambda^{\text{old}}) = \sum_{k=1}^{S} \sum_{x\in T_k} \Big\{ \sum_{i\in L_s} E_s^{\text{old}}(\boldsymbol{x}) \log\omega_{ik} + \sigma \sum_{i\in L_{pk}} E_{pk}^{\text{old}}(\boldsymbol{x}) \log\omega_{ik} \Big\} \tag{11}$$

$$Q_2(\Lambda \mid \Lambda^{\text{old}}) = \sum_{k=1}^{S} \sum_{x\in T_k} \Big\{ \sum_{i\in L_s} E_s^{\text{old}}(\boldsymbol{x}) \log p(\boldsymbol{x}, \mu_i, \Sigma_i) + \sigma \sum_{i\in L_{pk}} E_{pk}^{\text{old}}(\boldsymbol{x}) \log p(\boldsymbol{x}, \mu_i, \Sigma_i) \Big\} \tag{12}$$

then we can maximize separately the above quantities since they do not contain common parameters. In order to maximize $Q_1(\Lambda \mid \Lambda^{\text{old}})$, we must take account of the constraints $\sum_i \omega_{ik} = 1$. We use Lagrange multipliers and so the function $Q_1^L(\Lambda \mid \Lambda^{\text{old}})$ can be maximized as

$$Q_1^L(\Lambda \mid \Lambda^{\text{old}}) = Q_1(\Lambda \mid \Lambda^{\text{old}}) - \sum_{k=1}^{S} \kappa_k \Big(\sum_i \omega_{ik} - 1 \Big) \tag{13}$$

Expressing the derivatives of $Q_1^L(\Lambda \mid \Lambda^{\text{old}})$ with respect to ω_{ik}, we obtain the following update equation

$$\omega_{ik} = \begin{cases} \dfrac{\sum_{x\in T_k} E_s^{\text{old}}(\boldsymbol{x})}{\sum_{x\in T_k} 1}, & i \in L_s \\ \sigma \dfrac{\sum_{x\in T_k} E_{pk}^{\text{old}}(\boldsymbol{x})}{\sum_{x\in T_k} 1}, & i \in L_{pk} \end{cases} \tag{14}$$

Also the differentiation of $Q_2^L(\Lambda \mid \Lambda^{\text{old}})$ with respect to priors the kernel parameters μ_i, Σ_i leads to the following update equations

$$\mu_i = \frac{\sum_{k=1}^{S} \sum_{x\in T_k} E_s^{\text{old}}(\boldsymbol{x})\boldsymbol{x} + \sigma \sum_{x\in T_k} E_{pk}^{\text{old}}(\boldsymbol{x})\boldsymbol{x}}{\sum_{k=1}^{S} \sum_{x\in T_k} E_s^{\text{old}}(\boldsymbol{x}) + \sigma \sum_{x\in T_k} E_{pk}^{\text{old}}(\boldsymbol{x})} \tag{15(a)}$$

$$\Sigma_i = \frac{\sum_{k=1}^{S} \sum_{x\in T_k} E_s^{\text{old}}(\boldsymbol{x})(\boldsymbol{x} - \mu_i)(\boldsymbol{x} - \mu_i)^{\mathrm{T}} + \sigma \sum_{x\in T_k} E_{pk}^{\text{old}}(\boldsymbol{x})(\boldsymbol{x} - \mu_i)(\boldsymbol{x} - \mu_i)^{\mathrm{T}}}{\sum_{k=1}^{S} \sum_{x\in T_k} E_s^{\text{old}}(\boldsymbol{x}) + \sigma \sum_{x\in T_k} E_{pk}^{\text{old}}(\boldsymbol{x})} \tag{15(b)}$$

The convergence property of this two-step iterative procedure is proved in [9]. The

main idea of the scheme is to estimate model parameters and make it more accurately represent the distribution of the speech feature space. Another important issue in realizing the scheme is to consider the balance of the contributions between the PNN and SPNN models. Let us rewrite (15) into

$$\theta_i = (1-\beta)\phi_s + \beta\phi_{pk} \tag{16}$$

where θ_i denotes μ_i or Σ_i,

$$\beta = \frac{\sigma \sum_{x\in T_k} E_{pk}^{\text{old}}(\boldsymbol{x})}{\sum_{k=1}^{S}\sum_{x\in T_k} E_s^{\text{old}}(\boldsymbol{x}) + \sigma \sum_{x\in T_k} E_{pk}^{\text{old}}(\boldsymbol{x})} \tag{17(a)}$$

$$\phi_{pk} = \frac{\sigma \sum_{x\in T_k} E_{pk}^{\text{old}}(\boldsymbol{x})\vartheta}{\sigma \sum_{x\in T_k} E_{pk}^{\text{old}}(\boldsymbol{x})} \tag{17(b)}$$

$$\phi_s = \frac{\sum_{k=1}^{S}\sum_{x\in T_k} E_s^{\text{old}}(\boldsymbol{x})\vartheta}{\sum_{k=1}^{S}\sum_{x\in T_k} E_s^{\text{old}}(\boldsymbol{x})} \tag{17(c)}$$

where ϑ denotes $\boldsymbol{x}$ or $(\boldsymbol{x}-\mu_i)(\boldsymbol{x}-\mu_i)^{\mathrm{T}}$. It can be concluded that (17(b)) denotes the updating rule in training PNN kernel parameters and (17(b)) denotes the updating rule in training SPNN kernel parameters. Further, β is used to control the contributions of the two types of training. The value β is totally decided by the parameter σ, input feature and original parameter set. If $\beta=0$, the mixture model reduces to the SPNN model in which all speakers are influenced by the pattern layer kernels and the training process utilizes the SPNN training approach. If $\beta=1$, the mixture model reduces to the PNN model in which each speaker is only influenced by a set of kernel and the training process utilizes the PNN training approach. In order to decrease the number of the pattern layer kernel and optimize the network configuration, we define a constant factor $\beta_{\max}$ and require $\beta\leqslant\beta_{\max}$ to prevent the σ PNN model from becoming the PNN model.

Based on the above discussion and some added necessary inspective steps, the proposed σ PNN model and the improved EM algorithms are given as follows.

1) Specify the number of kernels M and the initial parameter values;

2) Set the parameter σ to a fixed value, the number of shared kernels L_s and separated kernels L_{pk}, $k=1,\cdots,S$;

3) E-step: for each training sample set T_k, $k=1,\cdots,S$, use (10) and current parameters set Λ^{old} to calculate a posterior probability $E(z_{i|k}(\boldsymbol{x})\,|\,T,\Lambda^{\text{old}})$, $i=1,\cdots,M$;

4) Use (10) and (17(a)) to calculate β. If $\beta>\beta_{\max}$, let $\beta=\beta_{\max}$. This step ensures that there are shared kernels in the σ PNN model;

5) M-step: Calculate the new parameters set Λ by (14) and (16), respectively;

6) The above steps are performed iteratively until convergence is reached.

4 Experimental results

We will use the closed-set text-independent speaker identification experiment results to verify the validity of the proposed model and algorithms. The mel frequency cepstral coefficients (MFCC) reflect the feature of the short-time speech spectra based on non-linearity psychological apperceive to different frequencies for our ears. Obviously, the recognition performance and antinoise capability of the MFCC are superior to the traditional linear predictive coding coefficients (LPCC). Besides the static cepstral obtained from the short-

time speech, due to the slow-changing character of speech signal, the static cepstral will produce the dynamic cepstral which changes slowly with time. The combination of the two cepstral will fully express the speaker's track model. The twelfth-order MFCC and its first-order difference vector Δ MFCC will be used as the feature parameters. The 10 speakers' (5 men and 5 women) speech data are used in the experiment. The speech contents are arbitrary Mandarin. The sampling frequency is 16 kHz. For each speaker there is 30 conversations, each about 15 seconds. The first 10 conversations are used for training and the other conversations are used for testing. To compare the system time robustness of the different models and algorithms the test data set contain 10 conversations sampled in the same time and other 10 conversations sampled in other different time. A frame size of 20 ms and a frame shift of 10 ms are used in the MFCC calculation. In order to guarantee accuracy and validity of the feature, we first normalize the dynamic range of the speech signal. We use the alterable and multi-threshold regulations based on the short-time energy, shot-time cross-zero rate and 1-order difference energy to effectively eliminate the silent part and the noise part from the speech signal. The PNN, SPNN and σ PNN models are used as the classifier. The objective is to find the speaker model which has the maximum a posteriori probability for a given testing speech feature sequence. All the parameters are decided by the experiment. The performance of the three PNN classifiers will be evaluated in the same population and speech condition. In the following three experiments, the network which involves 24 input pattern features, 10 output pattern classes and the different neurons of the pattern layer is set up.

In the first experiment, the performances of the three PNN classifiers are evaluated under the premise that the network used to represent each speaker is the same. The size is $24\times32\times1$. The neurons of the pattern layer in the PNN, SPNN and σ PNN models are 320, 32 and 176, respectively. The pattern layer of the σ PNN model contains 16 separated kernels and 16 shared kernels. The final results are given in Table 1.

Table 1 Identification correct rate comparison among PNN, SPNN and σ PNN models

	PNN(%)	SPNN(%)	σ PNN
testing data from the same time	97.1	80.4	89.4
testing data from different time	93.2	78.7	84.5

In the second experiment, the performances of the three PNN classifiers are evaluated under the premise that the network is the same. The size is $24\times320\times10$. So with regard to the whole network, the neurons of the pattern layer are 320. Each speaker is represented by 32 kernels in the PNN model, 320 kernels in the SPNN model and 16 separated kernels and 160 shared kernels in the σ PNN model. The final results are given in Table 2.

Table 2 Identification correct rate comparison among PNN, SPNN and σ PNN models

	PNN(%)	SPNN(%)	σ PNN(%)
testing data from the same time	97.1	99.4	98.7
testing data from different time	93.2	98.7	95.5

In the third experiment, the size of the PNN and σ PNN classifiers is evaluated under the premise that the testing data are sampled at the same time and the performance is selected as 97.1%. For PNN model, each speaker is represented by 32 kernels and the neurons of the pattern layer are 320. For the σ PNN model, each speaker is represented by 80 kernels and the neurons of the pattern layer are 224 with 16 separated kernels and 64 shared kernels.

From the view of the model identification rate, on condition that the network size is

the same, the proposed σ PNN model is superior to the PNN model by 2%. From the view of the network structure, on condition that the network performance is same, the proposed σ PNN model is smaller that the PNN model.

In Table 1, the performance of the SPNN model is inferior to the PNN model, which indicates the speaker speech feature space is separated, meanwhile it also has the overlapping regions. The former explains the reason why speaker can be recognized from the speech data and the latter is considered as the reason that speaker recognition rate is difficult to be satisfuctory. Another important conclusion is that better performance of the σ PNN model can be obtained from intermediate values of β, which is similar to the conclusion in [10]. The distinctions between the proposed model and the model in [10] are follows. First, the core networks of these two models are different, that is, the PNN and RBF network, respectively. So the theoretic foundations are also different. Second, the network structures are different. Our network model is based on the shared and separated kernels, while the model in [10] is based on the class kernels. So our model is more direct and reasonable to represent the speaker. Third, the mechanisms of the adjusted parameter are different. The parameter β in our model depends on data, while the adjusted parameter in [10] has to be decided beforehand. Moreover, our model inclines to enhance the effect of the SPNN model without decreasing the recognition performance of the network, which makes the network structure more reasonable and smaller. But [10] does not have this mechanism. Fourth, the above differences ultimately lead to the difference of the EM algorithm processing and the parameters updated rules.

5 Conclusion

Despite considerable progress in the PNN model, there is a room for improvement about network structure determination and performance. We have presented a novel SPNN model based on the idea of kernel shared by the pattern layer. Moreover, we further extended the above idea and proposed a novel σ PNN model based on the mixture of the PNN and SPNN models. We also expanded the EM approach for training the novel model efficiently. In conclusion, the proposed model leads to a quite small network structure and improves the classification performance. The model can also be used to other pattern classification domains. Further research will be focused on a dynamic approach to adjust the kernel number and a novel training approach based on the minimum classification error criterion or the genetic algorithm.

References

1 Furui H. Recent advances in speaker recognition. *Pattern Recognition Letters*, 1997, **18**(9): 859~872

2 Romero R D, Touretzky D S, Thibadeau G H. Optical Chinese character recognition using probabilistic neural networks. *Pattern Recognition*, 1997, **3**(8): 1279~1292

3 Tian B, Azimi-Sadjadi M R, Vonder-Haar T H, Reinke D L. Temporal updating scheme for probabilistic neural networks with application to satellite cloud classification. *IEEE Transactions on Neural Network*, 2000, **11**(7): 903~920

4 Ganchev T, Yopanoglou A T, Fakotakis N, Kokkinakis G. Probabilistic neural networks combined with gmms for speaker recognition over telephone channels. In: Proceeding of the 14th International Conference on Digital Signal Processing, 2002, **2**: 1081~1084

5 Specht D F. Probabilistic neural networks. *Neural Network*, 1990, **3**(1): 109~118

6 Streit R L, Luginbuhl T E. Maximum likelihood training of probabilistic neural networks. *IEEE Neural Network*, 1994, **5**(5): 764~783

7 Burrascano P. Learing vector quantization for the probabilistic neural networks. *IEEE Neural Network*, 1991, **2**(4): 456~461

8 Zaknich Z. A vector quantization reduction method for the probabilistic neural networks. In: Proceedings of IEEE Neural Network, NJ: Piscataway, 1997, 1117~1120

9 Dempsteer A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm. *Journal*

of Royal Statistical Society, Series B, 1977, **39**(1):1～38

10 MMichalis K T, Aristidis C L. Shared kernel models for class conditional density estimation. *IEEE Neural Network*, 2001, **12**(7):987～997

WANG Cheng-Ru Professor. His research interests include image and speech signal processing.

WANG Jin-Jia Teaching assistant. He received the master degree from Yanshan University in 2003. His research includes neural network and speaker recognition.

LIAN Qiu-Sheng Associate professor. He received the master degree from Yanshan University in 1996. His research includes image compression and coding.

一种新的用于说话人辨认的 PNN 分类器的研究

王成儒　　王金甲　　练秋生

（燕山大学通信与电子工程系　秦皇岛　066004）

（E-mail：01016888@sina.com）

摘　要　给出了一种新的类条件密度函数估计的 σ PNN 模型，它基于模式层共享的 PNN 和模式层分离的 PNN，即每个类不仅拥有一组只属于自己的模式层，还拥有所有类都共享的几个模式层，这里共享意味着每个核函数对所有类的条件密度估计都有贡献. 新模型的训练采用最大似然准则，并改进了 EM 算法来调整模型参数. 闭集文本自由说话人辨认试验证明了提出的模型及其算法的正确性.

关键词　概率神经网络，最大似然，期望最大化，说话人辨认

中图分类号　TP391