Speaker Identification by BYY Automatic Local Factor Analysis based Three-Level Voting Combination

Lei Shi[†], Dingsheng Luo[‡], Lei Xu[†]

[†] Department of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

[‡] School of Electronics Engineering & Computer Science, Peking University, Beijing, 100871, China

shil@cse.cuhk.edu.hk; dsluo@cis.pku.edu.cn; lxu@cse.cuhk.edu.hk

***Abstract*— Local Factor Analysis (LFA) is known as more general and powerful than Gaussian Mixture Model (GMM) in unsupervised learning via local subspace structure analysis. In the literature of text-independent speaker identification, GMM has been widely used and investigated, with some preprocessing and postprocessing approaches, while there still lacks efforts on LFA for this task. In pursuit of fast implementation for LFA modeling, this paper focuses on the Bayesian Ying-Yang automatic learning with data smoothing based regularization (BYY-A), which makes automatic model selection during parameter learning. Furthermore for sequence classification, we design and analyze a three-level voting combination, namely sequence, classifier and committee, respectively. Different combination approaches are designed with variant sequential topologies and voting schemes. Experimental results on the KING speech corpus demonstrate the proposed approaches' effectiveness and potentials.**

1 Introduction

Speaker recognition, well known as one of the most important topics in biometrics, is a process to determine speakers' identities based their voice, whose two most focused studies are speaker identification and verification [2, 3, 4, 9]. Speaker identification is to classify an unknown voice token as one out of reference speakers, whereas speaker verification is to accept or reject an identity claim. Moreover, a speaker recognition system often works in either of two operating modes: text dependent, where the same text is required for both training and testing, and text independent, where arbitrary text is allowed to utter without restriction [2, 4]. In this paper, we focus on the task of text-independent speaker identification.

In the literature, there have been many efforts using variant models for this task [2, 3, 4, 9], such as multilayer perceptrons (MLP), k-nearest-neighbors (KNN), Vector Quantization (VQ) or K-means algorithm, Gaussian Mixture Model (GMM), etc., some of which are also combined together with other techniques including committee voting, Mixture of Experts (ME), boosting methods, and so forth. Out of them, from the view of unsupervised modeling, GMM has been widely investigated and shown advantages over some simple models like K-means algorithm with more accurate parametric modeling and better results.

Local Factor Analysis (LFA) combines GMM with one well-known dimension reduction approach named Factor Analysis (FA) [5, 10, 13]. Instead of describing each local component roughly as Gaussian, LFA tries to further model each component by a local lower-dimensional subspace for more accurate description and better generalization [5, 12, 13]. Fixed its model configuration including the component number k and local hidden dimensions $\{m_l\}_{l=1}^k$, LFA can be efficiently trained via a maximum-likelihood (ML) way, usually implemented by the expectation-maximization (EM) algorithm [5, 7].

One significant problem for LFA is to select both component number k and hidden dimensions $\{m_l\}_{l=1}^k$, which is a typical model selection problem. In the literature of LFA model selection, the conventional two-phase procedure performs with the help of one of typical statistical criteria via maximum-likelihood learning, while usually suffering a greatly huge computational cost due to multiple implementations through all candidate models [13]. Under the motivation of saving costs, we focus on the automatic Bayesian Ying-Yang harmony learning with data smoothing based regularization, shortly denoted BYY-A [13], which starts with a large enough number of components and automatically implements model selection during parameter learning.

To make sequence classification, we adopt and design three cascading combination levels. In the first point, collecting one trained LFA model on each speaker's data, one classifier forms one combination. In the second, with different initializations and implementations, different classifiers are obtained to group a stochastic committee combination. Furthermore, to make the sequence classification, results on all samples along the sequence are combined [3, 8]. Importantly, although the sequential topology between classifier's and committee's combination is fixed, i.e. the former should be in advance of the latter, different schedules for the sequence combination result in variant sequential hybrid schemes and structures. In this paper, we list three sequential topologies and investigate ten different combination approaches, facilitated with several voting schemes. Compared with the work in [3], the proposed approaches are applied on the KING database, a benchmark speech corpus designed especially for text-independent speaker identification.

The rest of this paper is organized as follows. In Section 2, we review LFA and its model selection problem.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

Section 3 briefly describes BYY automatic harmony learning algorithm with data smoothing based regularization on LFA modeling (BYY-A). Consideration of different three-level combinations is illustrated and analyzed in Section 4. After comparative speaker identification experiments on KING corpus database in Section 5, we draw concluding remarks and prospect future work in Section 6.

2 Local Factor Analysis Model

Local Factor Analysis (LFA), or also named Mixture of Factor Analyzers (MFA), is a useful multivariate unsupervised learning model, exploring not only clusters but also local subspaces with wide applications in pattern recognition, bio-informatics, and financial engineering [5, 13]. Provided d-dimensional observed vector $\mathbf{x}$, LFA assumes that its distribution follows a mixture of k underlying components $p(\mathbf{x}) = \sum_{l=1}^{k} \alpha_l p(\mathbf{x}|l)$, where $p(\mathbf{x}|l)$ is the probability density of the l-th component, and α_l is the l-th component's prior with $\alpha_l \geq 0$ and $\sum_{l=1}^{k} \alpha_l = 1$. Furthermore, each component, instead of regarded roughly as Gaussian in Gaussian Mixture Model (GMM), is assumed as a Factor Analysis (FA) [5, 10], where observed data are regarded as generated from lower-dimensional hidden independent Gaussian factors $\mathbf{y}$ via linear transform [7]:

$$p(\mathbf{x}|\mathbf{y}, l) = G(\mathbf{x}|\mathbf{A}_l\mathbf{y} + \mu_l, \mathbf{\Psi}_l), \quad p(\mathbf{y}|l) = G(\mathbf{y}|\mathbf{0}, \mathbf{I}_{m_l}),$$
$$p(\mathbf{x}|l) = \int p(\mathbf{x}|\mathbf{y}, l)p(\mathbf{y}|l)d\mathbf{y} = G(\mathbf{x}|\mu_l, \mathbf{A}_l\mathbf{A}_l^T + \mathbf{\Psi}_l), \quad (1)$$

where $\mathbf{y}$ is the m_l-dimensional unobservable latent vector, $\mathbf{A}_l$ is the $d \times m_l$-dimensional loading matrix, μ_l is the d-dimensional mean vector, $\mathbf{\Psi}_l$ is the diagonal covariance matrix for noises. On a set of observed data $\{\mathbf{x}_t\}_{t=1}^{N}$, given the number of individual Gaussian components k and the local factor numbers $\{m_l\}_{l=1}^{k}$, one widely used method to estimate the unknown parameters $\mathbf{\Theta} = \{\alpha_l, \mathbf{A}_l, \mu_l, \mathbf{\Psi}_l\}_{l=1}^{k}$ is the maximum-likelihood (ML) learning, which can be effectively implemented by expectation-maximization (EM) algorithm [5, 7, 10].

One important remaining problem for LFA is to determine both the component number k and local hidden dimensions $\{m_l\}_{l=1}^{k}$, which is a typical model selection problem. Conventionally, this model selection task can be addressed in a *two-phase* procedure in help of $J(\hat{\mathbf{\Theta}}, k, \{m_l\}_{l=1}^{k})$, i.e., one of typical existing statistical model selection criteria such as AIC, CAIC, BIC which coincides with MDL, and cross-validation, etc [13]. In the first phase, both a range $[k_{min}, k_{max}]$ for k and a range $[m_{min}, m_{max}]$ for m_l are predetermined to set up a domain $\mathcal{M}$ assumed to contain the optimal $k^*, \{m_l^*\}_{l=1}^{k^*}$. Then for each model scale $\{k, \{m_l\}_{l=1}^{k}\}$, parameters $\mathbf{\Theta}$ are estimated usually by EM algorithm [5, 7, 10]. In the second phase, model selection is made through obtained candidate models such that: $\hat{k}, \{\hat{m_l}\} = \arg\min_{k,\{m_l\}}\{J(\hat{\mathbf{\Theta}}, k, \{m_l\}), \{k, \{m_l\}\} \in \mathcal{M}\}$. Thus, totally at least $\sum_{k=k_{min}}^{k_{max}}(m_{max} - m_{min} + 1)^k$ times estimates are needed.

3 BYY Automatic LFA Learning

Since the two-phase model selection procedure is computationally extensive and thus impractical in many real applications, alternatively, several efforts have been made on fast seeking model selection. One type of fast implementation is in a sense that with its scale initialized large enough to include the correct one, learning on a model will not only determine parameters but also automatically shrink its scale appropriately, while discarding those extra substructures. One representative effort following this type is Bayesian Ying-Yang (BYY) harmony automatic learning [13, 14, 15].

Firstly proposed as a unified statistical learning framework firstly in 1994 and systematically developed in the past decade [14], BYY harmony learning consists of a general BYY harmony system and a fundamental harmony learning principle as a unified guide for developing new regularization techniques, a new class of criteria for model selection, and a new family of algorithms that perform parameter learning with automatic model selection. Readers are referred to [13] for a recent systematic review. Applied on LFA, to remove the rotational indeterminacy, BYY harmony learning considers the following alternative but probabilistically equivalent model with l-th component's distribution as follows:

$$p(l) = \alpha_l, \quad p(\mathbf{y}|l) = G(\mathbf{y}|\mathbf{0}, \mathbf{\Lambda}_l),$$
$$p(\mathbf{x}|\mathbf{y}, l) = G(\mathbf{x}|\mathbf{U}_l\mathbf{y} + \mu_l, \mathbf{\Psi}_l), \quad \mathbf{U}_l^T\mathbf{U}_l = \mathbf{I}_{m_l},$$
$$p(\mathbf{x}|l) = G(\mathbf{x}|\mu_l, \mathbf{U}_l\mathbf{\Lambda}_l\mathbf{U}_l^T + \mathbf{\Psi}_l), \quad (2)$$

where $\mathbf{y}$ is the m_l-dimensional hidden factor, μ_l is the d-dimensional mean, both $\mathbf{\Lambda}_l$ and $\mathbf{\Psi}_l$ are diagonal covariance matrices, while loading matrix $\mathbf{U}_l$ is constrained on the Stiefel manifold $\mathbf{U}_l^T\mathbf{U}_l = \mathbf{I}_{m_l}$.

Furthermore, data smoothing based regularization [11, 13] combining parametric model with Parzen window nonparametric model [1, 6] is adopted, i.e., $p_h(\mathbf{x}) = \frac{1}{N}\sum_{t=1}^{N} G(\mathbf{x}|\mathbf{x}_t, \mathbf{\Xi}_h)$. As $\mathbf{\Xi}_h \to \mathbf{0}$, $p_h(\mathbf{x})$ becomes the empirical distribution. There have been some previous efforts applying the data smoothing technique onto GMM modeling [15], where the kernel bandwidth matrix usually took the simplest form $\mathbf{\Xi}_h = h^2\mathbf{I}_d$, whose limitation is the data smoothing effect has to be identical and orthogonal, thus unable to precisely model the distribution with high correlation. In this paper, we consider a general form where the kernel can take any symmetric positive definite bandwidth matrix $\mathbf{\Xi}_h$.

By the nature of BYY harmony learning, for LFA the target is to maximize $H_s(\mathbf{\Theta}, \mathbf{\Xi}_h)$, the harmony function with data smoothing, as shown in Fig. 1. After initialization with large enough $k = k_{init}$, $m_l = m_{l,init}$ for $l = 1, \ldots, k$, model selection is implemented automatically during parameter learning, where the whole algorithm, shortly named BYY-A [13], iterates by three steps named **Yang-Step**, **Ying-Step** and **Smoothing-Step**, respectively. The adaptive learning process is stopped when there is no further improvement on harmony function and the parameter estimation converges.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

Targeted harmony function $H_s(\Theta, \Xi_h)$:

$H_s(\Theta, \Xi_h) = L_h(\Theta) + Z(\Xi_h), \quad Z(\Xi_h) = -\ln \sum_{t=1}^{N} p_h(\mathbf{x}_t)$

$L_h(\Theta) = \sum_{l=1}^{k} \int p_h(\mathbf{x}) p(l|\mathbf{x}) p(\mathbf{y}|\mathbf{x}, l) \ln[\alpha_l p(\mathbf{y}|l) p(\mathbf{x}|\mathbf{y}, l)] d\mathbf{x} d\mathbf{y},$

Initialization:

Select large enough k_{init} and m_{init} and randomly initialize a k_{init} component LFA model with m_{init} local hidden dimensions for each component, where $m_{init} \leq d$.

Yang-Step:

After selecting a sample $\mathbf{x}_t$, for each component $l = 1, \ldots, k$, calculate

$\mathbf{e}_{t,l} = \mathbf{x}_t - \mu_l$, $\hat{\mathbf{y}}_l(\mathbf{x}) = \arg\max_{\mathbf{y}} \ln[p(\mathbf{x}|\mathbf{y}, l) p(\mathbf{y}|l)] = \mathbf{W}_l(\mathbf{x} - \mu_l)$, $\mathbf{W}_l = \mathbf{\Lambda}_l \mathbf{U}_l^T \mathbf{M}_l$,

$\mathbf{M}_l = (\mathbf{U}_l \mathbf{\Lambda}_l \mathbf{U}_l^T + \mathbf{\Psi}_l)^{-1} = \mathbf{\Psi}_l^{-1} - \mathbf{\Psi}_l^{-1} \mathbf{U}_l (\mathbf{\Lambda}_l^{-1} + \mathbf{U}_l^T \mathbf{\Psi}_l^{-1} \mathbf{U}_l)^{-1} \mathbf{U}_l^T \mathbf{\Psi}_l^{-1}$,

$$p(l|\mathbf{x}_t) = \begin{cases} 1, & \text{if} \quad l = l_t, \\ 0, & \text{otherwise.} \end{cases}, \quad l_t = \arg\max_l \ln[\alpha_l p(\hat{\mathbf{y}}_l(\mathbf{x}_t)|l) p(\mathbf{x}_t|\hat{\mathbf{y}}_l(\mathbf{x}_t), l)].$$

Ying-Step:

$$\alpha_l^{new} = \begin{cases} \frac{\alpha_l + \eta_0}{1 + \eta_0}, & \text{if} \quad l = l_t, \\ \frac{\alpha_l}{1 + \eta_0}, & \text{otherwise.} \end{cases} \qquad \mu_{l_t}^{new} = \mu_{l_t} + \eta_0 \mathbf{e}_{t,l_t}, \quad \varepsilon_{t,l} = \mathbf{e}_{t,l} - \mathbf{U}_l \hat{\mathbf{y}}_l(\mathbf{x}_t),$$

$\mathbf{\Lambda}_{l_t}^{new} = (1 - \eta_0) \mathbf{\Lambda}_{l_t} + \eta_0 diag[\mathbf{W}_{l_t} \mathbf{\Xi}_h \mathbf{W}_{l_t}^T + \hat{\mathbf{y}}_{l_t}(\mathbf{x}_t) \hat{\mathbf{y}}_{l_t}(\mathbf{x}_t)^T]$,

$\mathbf{\Psi}_{l_t}^{new} = (1 - \eta_0) \mathbf{\Psi}_{l_t} + \eta_0 diag[(\mathbf{I}_d - \mathbf{U}_{l_t} \mathbf{W}_{l_t}) \mathbf{\Xi}_h (\mathbf{I}_d - \mathbf{U}_{l_t} \mathbf{W}_{l_t})^T + \varepsilon_{t,l} \varepsilon_{t,l}^T]$.

Update $\mathbf{U}_{l_t}$ by gradient on the Stiefel manifold,

$\mathbf{U}_{l_t}^{new} = \mathbf{U}_{l_t} + \eta_0 (\mathbf{G}_{\mathbf{U}_{l_t}} - \mathbf{U}_{l_t} \mathbf{G}_{\mathbf{U}_{l_t}}^T \mathbf{U}_{l_t})$,

$\mathbf{G}_{\mathbf{U}_{l_t}} = \mathbf{M}_{l_t} \mathbf{e}_{t,l_t} \hat{\mathbf{y}}_{l_t}(\mathbf{x}_t)^T + \mathbf{M}_{l_t} \mathbf{\Xi}_h \mathbf{W}_{l_t}^T$.

Discard the l-th component if α_l approaches 0.

Discard the j-th factor of the l-th component if the j-th element of $\mathbf{\Lambda}_l$ approaches 0.

Smoothing-Step:

Update the smoothing parameter $\mathbf{\Xi}_h$ as follows:

$\mathbf{\Xi}_h^{new} = \mathbf{R}^{new^T} \mathbf{R}^{new}$, $\mathbf{R}^{new} = \mathbf{R} - \eta_0 \Delta \mathbf{R}$,

$\Delta \mathbf{R} = (\mathbf{R}^T)^{-1} - \alpha_{l_t} \mathbf{R} (\mathbf{U}_l \mathbf{\Lambda}_{l_t} \mathbf{U}_{l_t}^T + \mathbf{\Psi}_{l_t})^{-1}$

Stop Condition:

If parameters estimation converges, i.e., no further improvement on the harmony function.

Figure 1: Algorithm sketch for BYY automatic LFA learning approach with data smoothing based regularization (BYY-A).

4 Three-Level Voting Combination

To make classification without rejection allowed, w LFA models are independently initialized and trained on training data of each class s ($s = 1, \ldots, S$), and then decreasingly ordered by the harmony function. Thereafter, one stochastic classifier is formed by putting S selected LFA models together, with one LFA randomly selected from one speaker's f ***first*** models ($1 \leq f \leq w$). After repeating this selection procedure for C times, a stochastic classifier set of size C are determined, namely a stochastic committee, as shown in Fig. 2. In our experiments, we set and fix $w = 5$, $f = 3$, and $C = 20$.

For sequence classification like the speaker identification task, we adopt and design three cascading combination levels. In the first point, collecting one trained LFA model on one speaker's data, a classifier combination is formed. In the second, different classifiers are grouped into a committee combination. Furthermore, results on all samples along the sequence are combined. Importantly, although the sequential topology between the classifier's and committee's combination is fixed, i.e. the former should be in advance of the latter, different schedules for the sequence combination result in variant hybrid schemes and structures. In the sequel, we describe and investigate three different sequential topologies and ten different ***voting*** combination approaches, as shown in Fig. 3.

4.1 Classifier-Committee-Sequence Hybrid

To classify a sequence $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$, one usually used scheme is ***voting*** by each sample $\mathbf{x}_t$ along it, where the sequence combination is positioned at the last level. Since the classifier is combined ahead of committee, this three-level combination is a ***Classifier-Committee-Sequence*** hybrid. Under this sequential topology, we consider four voting combination methods caused by different operations.

In the first method, supposing in one classifier the s-th ($s = 1, \ldots, S$) class has one LFA model with k_s components, as a testing data $\mathbf{x}_t$ comes, the likelihood $p(\mathbf{x}_t|s, l)$ is computed for each component $l = 1, \ldots, k_s$ and $s = 1, \ldots, S$. Then the κ largest valued components are listed out to form a subset $\mathbf{K}$, where the s-th class has $\mathbf{K}_s \subseteq \mathbf{K}$. Then $\mathbf{x}_t$ is classified to the class $\hat{s}(\mathbf{x}_t) = \arg\max_s |\mathbf{K}_s|$, where actually we just need to count the number of components in each $\mathbf{K}_s$. This idea can be regarded as an extension of the well-known kNN approach in the application of LFA for a classification problem, namely *kNN voting*.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

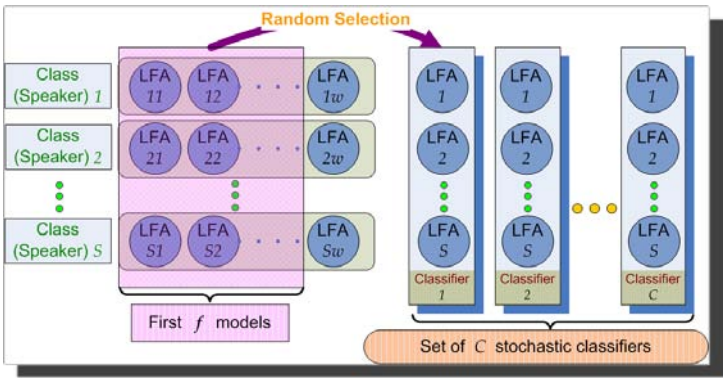

Figure 2: Flowchart of the acquisition of stochastic classifier and committee. On each class training data, w LFA models are trained, the first f of which ordered by harmony function and colored pink form the candidate model set. One stochastic classifier is obtained by randomly select one LFA model for each class from the set. Afterwards a stochastic classifier set, a committee, is formed by collecting C classifiers.

Hybrid Methods	Cascading Combination Levels		
	Level 1	Level 2	Level 3
1	classifier (kNN voting) $\hat{s}(\mathbf{x}_t) = \arg\max_s \lvert\mathbf{K}_s\rvert$	committee (voting) Each classifier votes $\hat{s}(\mathbf{x}_t)$	sequence (voting) Each $\mathbf{x}_t$ votes s
2	classifier (kNN approximation) $\hat{s}(\mathbf{x}_t) = \arg\max_s \tilde{p}(\mathbf{x}_t\lvert s)$ $\tilde{p}(\mathbf{x}_t\lvert s) = \sum_{i\in\mathbf{K}_s} \tilde{\alpha}_i p(\mathbf{x}_t\lvert i)$	committee (voting) Each classifier votes $\hat{s}(\mathbf{x}_t)$, and take the maximum s	sequence (voting) Each $\mathbf{x}_t$ votes s
3	classifier (kNN approximation & Bayesian) $\tilde{p}(\mathbf{x}_t\lvert s) = \sum_{i\in\mathbf{K}_s} \tilde{\alpha}_i p(\mathbf{x}_t\lvert i)$ $p(s\lvert\mathbf{x}_t) = \frac{\beta_s \tilde{p}(\mathbf{x}_t\lvert s)}{\sum_{i=1}^{S} \beta_i \tilde{p}(\mathbf{x}_t\lvert i)}$	committee (Bayesian voting) Each classifier votes $p(s\lvert\mathbf{x}_t)$, and take the maximum s	sequence (voting) Each $\mathbf{x}_t$ votes s
4	classifier (Bayesian) $p(s\lvert\mathbf{x}_t) = \frac{\beta_s p(\mathbf{x}_t\lvert s)}{\sum_{i=1}^{S} \beta_i p(\mathbf{x}_t\lvert i)}$ $p(\mathbf{x}_t\lvert s) = \sum_{i\in s} \alpha_i p(\mathbf{x}_t\lvert i)$	committee (Bayesian voting) Each classifier votes $p(s\lvert\mathbf{x}_t)$, and take the maximum s	sequence (voting) Each $\mathbf{x}_t$ votes s
5	classifier (kNN voting) $\hat{s}(\mathbf{x}_t) = \arg\max_s \lvert\mathbf{K}_s\rvert$	sequence (voting) $\hat{s}(\mathbf{X}) = \arg\max_s \sum_{t=1}^{T} \delta[\hat{s}(\mathbf{x}_t) - s]$	committee (voting) Each classifier votes $\hat{s}(\mathbf{X})$
6	classifier (kNN approximation) $\tilde{p}(\mathbf{x}_t\lvert s) = \sum_{i\in\mathbf{K}_s} \tilde{\alpha}_i p(\mathbf{x}_t\lvert i) + \epsilon$, ϵ is a small positive value to avoid 0	sequence (product & Bayesian) $p(\mathbf{X}\lvert s) = \prod_{t=1}^{T} \tilde{p}(\mathbf{x}_t\lvert s)$ $p(s\lvert\mathbf{X}) = \frac{\beta_s p(\mathbf{X}\lvert s)}{\sum_{i=1}^{S} \beta_i p(\mathbf{X}\lvert i)}$	committee (Bayesian voting) Each classifier votes $p(s\lvert\mathbf{X})$
7	classifier (kNN approximation & Bayesian) $\tilde{p}(\mathbf{x}_t\lvert s) = \sum_{i\in\mathbf{K}_s} \tilde{\alpha}_i p(\mathbf{x}_t\lvert i)$ $p(s\lvert\mathbf{x}_t) = \frac{\beta_s \tilde{p}(\mathbf{x}_t\lvert s)}{\sum_{i=1}^{S} \beta_i \tilde{p}(\mathbf{x}_t\lvert i)}$	sequence (Bayesian voting) Each sample $\mathbf{x}_t$ votes $p(s\lvert\mathbf{x}_t)$, and take the maximum as $\hat{s}(\mathbf{X})$	committee (voting) Each classifier votes $\hat{s}(\mathbf{X})$
8	classifier (Bayesian) $p(s\lvert\mathbf{x}_t) = \frac{\beta_s p(\mathbf{x}_t\lvert s)}{\sum_{i=1}^{S} \beta_i p(\mathbf{x}_t\lvert i)}$ $p(\mathbf{x}_t\lvert s) = \sum_{i\in s} \alpha_i p(\mathbf{x}_t\lvert i)$	sequence (Bayesian voting) Each sample $\mathbf{x}_t$ votes $p(s\lvert\mathbf{x}_t)$, and take the maximum as $\hat{s}(\mathbf{X})$	committee (voting) Each classifier votes $\hat{s}(\mathbf{X})$
9	sequence (product) $p(\mathbf{X}\lvert s) = \prod_{t=1}^{T} p(\mathbf{x}_t\lvert s)$ $p(\mathbf{x}_t\lvert s) = \sum_{i\in s} \alpha_i p(\mathbf{x}_t\lvert i)$	classifier (voting) $\hat{s}(\mathbf{X}) = \arg\max_s p(s\lvert\mathbf{X})$ $= \arg\max_s p(\mathbf{X}\lvert s)$	committee (voting) Each classifier votes s
10	sequence (product) $p(\mathbf{X}\lvert s) = \prod_{t=1}^{T} p(\mathbf{x}_t\lvert s)$ $p(\mathbf{x}_t\lvert s) = \sum_{i\in s} \alpha_i p(\mathbf{x}_t\lvert i)$	classifier (Bayesian) $p(s\lvert\mathbf{X}) = \frac{\beta_s p(\mathbf{X}\lvert s)}{\sum_{i=1}^{S} \beta_i p(\mathbf{X}\lvert i)}$	committee (Bayesian voting) Each classifier votes $p(s\lvert\mathbf{X})$

Figure 3: Comparative explanation of different combination approaches.

In the second level, committee's classification is voted by each classifier's selection on $\hat{s}(\mathbf{x}_t)$. Finally, the whole sequence $\mathbf{X}$'s classification is obtained by each $\mathbf{x}_t$'s voting from the committee selection.

In the second method, instead of kNN voting as the previous first level, more information is adopted from the

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

selected subset $\mathbf{K}$ via approximation of the probability $p(\mathbf{x}_t|s)$, i.e., $\tilde{p}(\mathbf{x}_t|s) = \sum_{i\in\mathbf{K}_s} \tilde{\alpha}_i p(\mathbf{x}_t|i)$, where $\tilde{\alpha}_s$ is the normalized prior in selected subset $\mathbf{K}_s$ for the s-th class, with $\tilde{p}(\mathbf{x}_t|s) = 0$ if $\mathbf{K}_s = \phi$. Following this *kNN approximation*, $\hat{s}(\mathbf{x}_t)$ with the maximum value is selected, i.e., $\hat{s}(\mathbf{x}_t) = \arg\max_s \tilde{p}(\mathbf{x}_t|s)$. Then the committee combines each classifier's $\hat{s}(\mathbf{x}_t)$ via voting to classify $\mathbf{x}_t$, followed by the same sequence voting combination as the first method.

In the third method, on the first level, besides the kNN approximation, a Bayesian is conducted via $p(s|\mathbf{x}_t) = \beta_s\tilde{p}(\mathbf{x}_t|s)/[\sum_{i=1}^{S}\beta_i\tilde{p}(\mathbf{x}_t|i)]$, where β_s is the prior of the s-th class, assumed either equally $1/S$ or proportional to that class's sample size. On the second level, each classifier votes through classes with its obtained Bayesian weights $p(s|\mathbf{x}_t)$, then the committee classifies $\mathbf{x}_t$ to the class with the maximum score. The sequence combination level is the same as the first method.

In the fourth method, the classifier level conducts Bayesian without kNN-like approximation, i.e., $p(s|\mathbf{x}_t) = \beta_s p(\mathbf{x}_t|s)/[\sum_{i=1}^{S}\beta_i p(\mathbf{x}_t|i)]$. In the committee level, each classifier votes with its $p(s|\mathbf{x}_t)$ weights, and then the maximum selection is the committee's classification result for $\mathbf{x}_t$. Sequence combination is the same as the first method.

4.2 Classifier-Sequence-Committee Hybrid

This subsection focuses on another sequential topology, ***Classifier-Sequence-Committee*** combination, where the sequence combination is put between the classifier level and committee level. In the fifth method, classifier's classification for sample $\mathbf{x}_t$ is made by kNN voting as $\hat{s}(\mathbf{x}_t) = \arg\max_s |\mathbf{K}_s|$. Afterwards the sequence classification by current classifier is made by all samples voting, i.e., $\hat{s}(\mathbf{X}) = \arg\max_s \sum_{t=1}^{T}\delta(\hat{s}(\mathbf{x}_t) - s)$, where $\delta(\bullet)$ is the Kronecker delta function used for voting. Finally in the committee level, all classifiers vote their selections of $\hat{s}(\mathbf{X})$ to assign the most preferred one as the final classification.

The sixth method realizes the classifier combination by the kNN approximation similar to the third method as $\tilde{p}(\mathbf{x}_t|s) = \sum_{i\in\mathbf{K}_s}\tilde{\alpha}_i p(\mathbf{x}_t|i) + \epsilon$, where the ϵ is a small positive value to avoid the potential multiplied zero result in the second level. Then the sequence combination level assumes each sample is conditional independent distributed and thus $p(\mathbf{X}|s) = \prod_{t=1}^{T}\tilde{p}(\mathbf{x}_t|s)$. Afterwards, Bayesian is conducted via $p(s|\mathbf{X}) = \beta_s p(\mathbf{X}|s)/[\sum_{i=1}^{S}\beta_i p(\mathbf{X}|i)]$, so as to obtain the Bayesian voting weight in the next level. The final committee's result is made by all classifiers' Bayesian voting.

The seventh method owns the same classifier level as the third method, i.e., a kNN approximation and Bayesian procedure. For the sequence combination, each sample $\mathbf{x}_t$ votes for each s with its Bayesian weights, and then the maximum scored class is selected as the classification result by current classifier for the whole sequence $\mathbf{X}$. Final committee voting is implemented from each classifier's sequence classification result $\hat{s}(\mathbf{X})$.

The eighth method shares the same first level with the fourth method, i.e. a Bayesian on $p(s|\mathbf{x}_t)$. Then the sequence voting is performed by each sample with the obtained Bayesian weights, so that a maximum scored class is taken as $s(\hat{\mathbf{X}})$ by current classifier. Final level on committee combination is the same as the seventh method.

4.3 Sequence-Classifier-Committee Hybrid

In this topology, the sequence combination is conducted on each LFA model first, and then each LFA model is combined in the classifier level, and finally the committee's result is combined from each classifier's, thus a ***Sequence-Classifier-Committee*** structure. In the first level of sequence combination for this topology, $\{\mathbf{x}_t\}_{t=1}^{T}$ are assumed independently distributed, so that $p(\mathbf{X}|s) = \prod_{t=1}^{T} p(\mathbf{x}_t|s)$. The variants in this topology come from the following two levels. We do not adopt the kNN idea into this sequential topology, because the sequence combination is conducted in the first level, merging all samples' information together, while the kNN technique is more meaningful for a single sample instead of a sample set.

In the ninth method, the second level of classifier combination is to classify $\mathbf{X}$ to the class with the maximum posterior probability, i.e. $\hat{s}(\mathbf{X}) = \arg\max_s p(s|\mathbf{X})$. In the third level, the committee lets each classifier select through classes and vote for one, resulting in a final decision.

For the tenth method, on the level of each classifier, Bayesian weights are obtained for all classes by $p(s|\mathbf{X}) = \beta_s p(\mathbf{X}|s)/[\sum_{i=1}^{S}\beta_i p(\mathbf{X}|i)]$. Finally the committee collects all classifiers' Bayesian weighted votes and makes selection to the maximum scored one.

5 Application on KING Corpus

The proposed approaches are applied on the KING database, a benchmark English speech corpus designed especially for text-independent speaker identification. It consists of wide-band (WB) and narrow-band (NB) sets, where WB was collected with a high-quality microphone in a quiet room, while NB by telephone handsets through various long distance telephone channels. In each set, all speakers are male and ten sessions for each speaker were recorded. To save space, the data and preprocessing are the same with and referred to [3] for detail.

For text-independent speaker identification problem, a sequence of feature vectors is divided into overlapping segments of T frames, as suggested by Reynolds [8] as follows.

$$\ldots, \overbrace{\mathbf{x}_l, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+T-1}}^{\text{segment } l}, \mathbf{x}_{l+T}, \ldots$$
$$\ldots, \mathbf{x}_l, \overbrace{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+T-1}, \mathbf{x}_{l+T}}^{\text{segment } l+1}, \ldots$$

For the l-th testing segment, its sequence classification is implemented by the discussed approaches. The correct

Hybrid Methods	Data Sets (Training Length) WB (70s)	NB (70s)
1	90.8 ± 2.8	67.8 ± 6.3
2	91.9 ± 3.1	70.1 ± 4.8
3	93.8 ± 4.1	75.2 ± 5.5
4	92.6 ± 3.3	74.3 ± 4.2
5	91.0 ± 4.5	68.9 ± 7.6
6	91.3 ± 2.6	72.9 ± 8.1
7	92.8 ± 3.2	72.6 ± 6.4
8	92.6 ± 3.7	74.1 ± 6.3
9	91.7 ± 3.6	69.4 ± 5.7
10	92.1 ± 3.2	73.2 ± 4.6
GGMM*	92.3 ± 3.9	73.6 ± 6.8

Figure 4: Experimental results of correct speaker identification rates on KING corpus database. The length of testing segment is 8s. (*): The result is taken from [3].

identification rate through all testing segments is obtained as the result for one trial. Same as [3], our simulations adopt multiple trials for obtaining the reliable performance, where in each trial two sessions are randomly selected from ten recoding sessions for training and the remaining eight for testing. The training data from two sessions are of duration 70s, while the testing data segments are set with length $T = 8$s. In total, ten trials have been performed and the overall performance is reported in Fig. 4. Again, in our experiments, we set and fix $w = 5$, $f = 3$, and $C = 20$, to compose the stochastic committee. The GGMM's results are directly collected from [3] for comparison, which is claimed as the best approach among their considered.

From the results we can find that, those methods adopting kNN approximation and Bayesian voting generally produce comparably good or better results than the referred GGMM [3], while others worse. Interestingly, the classification rate variances by the voting approaches are mostly less than GGMM.

6 Conclusion and Future Work

For the application of speaker identification, in this paper we adopt one fast Local Factor Analysis (LFA) modeling implementation approach named BYY-A, which makes automatic model selection during parameter learning. For the sequence classification task, a series of three-level combination structures are designed and analyzed to combine trained LFA models, mainly using variant voting mechanisms. This three-level cascading combination can be regarded as a pseudo-dimension-reduction from 4 dimensions, including models, classifiers, classes and samples along sequence, into 1 dimension, i.e. the classes. Based on the considered KING corpus database, the experimental results of correct identification rates show these LFA based voting approaches' advantage and potential in classifying and modeling high dimensional sequence data.

Furthermore, the experimental results also indicate that, in the post-processing, among these different combination and voting mechanisms some outperform the others, leaving mathematical stochastic analysis and comparison expected in future. Nevertheless, as indicated by the results, several results are better than the previously reported GGMM's result [3], which combines GMM models via a trained Mixture of Experts (ME) gating network. Thus, further investigation on comparing LFA based ME with these voting approaches is being carried out, with results expected later.

References

[1] A. W. Bowman and A. Azzalini. *Applied Smoothing Techniques for Data Analysis*. Oxford Statistical Science Series. Oxford Science Publications, 1997.

[2] J. P. Campbell. Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

[3] K. Chen. On the use of different speech representations for speaker modeling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 35(3):301–314, 2005.

[4] S. Furui. Recent advances in speaker recognition. *Pattern Recognition Letters*, 18:859–872, 1997.

[5] Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, 21 1996.

[6] E. Parzen. On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[7] R. Redner and H. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984.

[8] D. A. Reynolds. *A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification*. PhD thesis, Georgia Institute of Technology, 1992.

[9] D. A. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Commun.*, 17(1-2):91–108, 1995.

[10] D. Rubin and D. Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76, March 1982.

[11] A. Tikhonov and V. Arsenin. *Solutions of Ill-posed Problems*. Winston and Sons, 1977.

[12] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[13] L. Xu. Bayesian Ying Yang Learning. In *Scholarpedia p.10469, http://scholarpedia.org/article/Bayesian_Ying_Yang_Learning*.

[14] L. Xu. Bayesian-Kullback coupled Ying-Yang machines: Unified learnings and new results on vector quantization. In *ICONIP*, pages 977–988, 1995.

[15] L. Xu. Data smoothing regularization, multi-sets-learning, and problem solving strategies. *Neur. Netw.*, 16(5-6):817–825, 2003.