A New Approach to Utterance Verification Based on Neighborhood Information in Model Space

Hui Jiang, *Member, IEEE,* and Chin-Hui Lee, *Fellow, IEEE*

***Abstract*—In this paper, we propose to use neighborhood information in model space to perform utterance verification (UV). At first, we present a nested-neighborhood structure for each underlying model in model space and assume the underlying model's competing models sit in one of these neighborhoods, which is used to model alternative hypothesis in UV. Bayes factors (BF) is first introduced to UV and used as a major tool to calculate confidence measures based on the above idea. Experimental results in the Bell Labs communicator system show that the new method has dramatically improved verification performance when verifying correct words against mis-recognized words in the recognizer's output, relatively more than 20% reduction in equal error rate (EER) when comparing with the standard approach based on likelihood ratio testing and anti-models.**

***Index Terms*—Bayes factors, Bayesian predictive density, confidence measure, neighborhood in model space, utterance verification, Viterbi Bayesian Predictive Classification (VBPC).**

I. Introduction

WE KNOW THAT today's automatic speech recognition (ASR) systems are always fraught with recognition errors even in very constrained conditions. Recently, as more and more ASR systems are deployed in real-world applications, it becomes extremely urgent to equip the ASR system with the capability to evaluate the reliability of speech recognition results. Based on the reliability measurements, a series of further actions can be taken after recognition, e.g., to smartly reject nonspeech noises, detect/reject out-of-vocabulary words, even detect/correct some potential recognition mistakes, guide the system to perform unsupervised learning, and provide side information to assist high level speech understanding, and so on and so forth (refer to [18] for more other applications). It becomes very clear that a good reliability measurement is one of the most crucial techniques to make today's ASR systems more "intelligent". In the beginning of this paper, we first briefly explain the reason why a reliability measurement is missing from the conventional ASR procedure. Then we review many different methods which have been proposed to derive a reliability measurement for ASR in the literature. Finally we will present and focus on a completely different approach which computes this kind of reliability measures based on the "neighborhood" information in model space.

Manuscript received September 23, 2002; revised April 21, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jerome R. Bellegarda.

H. Jiang is with the Department of Computer Science, York University, Toronto, ON, M3J 1P3, Canada (e-mail: hj@cs.yorku.ca).

C. H. Lee is with the is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (chl@ece.gatech.edu).

Digital Object Identifier 10.1109/TSA.2003.815821

It is well known that conventional ASR algorithms are usually formulated as a pattern classification problem using the *maximum a posteriori* (MAP) decision rule to find the most likely sequence of words which achieves the maximum *posteriori* probability $p(W|X)$, i.e.,

$$\begin{aligned}\hat{W} &= \arg\max_{W\in\Gamma} p(W|X) \\ &= \arg\max_{W\in\Gamma} \frac{p(X|W)\cdot p(W)}{p(X)} \\ &= \arg\max_{W\in\Gamma} p(X|W)\cdot p(W) \end{aligned} \tag{1}$$

where X is the sequence of input feature vectors representing the underlying utterance, W is a word sequence, Γ is the set of all permissible sentences, $p(W)$ is the probability of W evaluated with a language model, $p(X)$ is the probability of observing X, and $p(X|W)$ is the probability of observing X under the assumption that W is the underlying word sequence for X. In theory, the posterior probability $p(\hat{W}|X)$ is a good confidence measure for the recognition result $\hat{W}$ given the acoustic input X. However, as shown in the above (1), most practical ASR systems simply ignore the term $p(X)$ in decision-making because it is constant across different words W. Therefore, the raw ASR scores (only representing relative differences) become inadequate as confidence measures to judge recognition reliability because the raw score can not tell how well the match is unless it is normalized by $p(X)$. In fact, without any model constraint, it is extremely difficult to have an accurate estimate of $p(X)$ for a given acoustic input X. In practice, many different heuristic methods must be used to approximate it.

During the past years, a lot of research works have been done in this field to seek for a reliability measurement for ASR, mainly driven by an increasing number of dialogue applications. Based on this sort of reliability measurement, machines will be able to handle the error-prone ASR outputs more intelligently. Generally speaking, the related works reported in the literature can be classified into two major categories. Firstly, under the name of Confidence Measures (CMs), various methods have been proposed to calculate the probability of a word W being correctly recognized by an ASR system, such as [3], [5], [13], [16], [17], [24], [25], [27], [28], and so on. Some of these works are based on how to calculate the *posterior* probability $p(\hat{W}|X)$ for the ASR output $\hat{W}$. As we mentioned above, the posterior probability is a good candidate for CM but it is very hard to estimate the distribution $p(X)$ in a precise manner. In practice, many heuristic methods are proposed to approximate it. The first ones are the so-called *filler-based*

methods which try to calculate $p(X)$ from a set of general filler models, i.e., all-phone recognition [28], catch-all model [13], the highest score in recognizing the word from decoder [3], etc. And the second one is called *lattice-based* method which attempts to calculate $p(X)$ from a word lattice (or graph) based on the forward-backward algorithm, such as in [16], [27]. Sometimes, in place of word lattice (or graph), an N-Best list can also be used for this purpose for the sake of simplicity [26]. So far, to our knowledge, comparing with other methods to approximate the posteriori probability, the method based on a word-graph gives the best performance since the word graph (or lattice) represents the information of all other possible competing paths during Viterbi search in a fairly accurate way. However, both the generation of word graphs and the computation of the posteriori probabilities over a word graph are relatively complicated. Besides the posterior-probability-based confidence measures (CMs), many people have also proposed some informal ways to derive CMs, i.e., the combination method, where a bunch of the so-called feature predictors are first collected from the speech recognition procedure, such as acoustic stability [4], hypothesis density [16], language model (LM) backoffs, duration, and many others (see [2], [12], [24], [25]). Then all these features are combined with a linear model or neural networks to derive a CM score for every recognized word. Secondly, in another major category, mostly motivated by speaker verification, some people [20], [23] have proposed the utterance verification (UV) approach which attempts to verify the claimed content of a spoken utterance. The content can be hypothesized by a speech recognizer or keyword detector or human transcriber. Under the framework of utterance verification (UV), the problem can be formulated as a statistical hypothesis testing problem [20], [23]. According to Neyman-Pearson Lemma, under certain conditions, the optimal solution is based on a likelihood ratio testing (LRT). The LRT-based utterance verification provides a good theoretical formulation to attack the tough problem of ASR reliability measurement. The major difficulty with LRT in utterance verification is how to model the alternative hypothesis, where the true distribution of data is unknown and alternative hypothesis usually represents a very complex and composite event. In [20], [23], the same HMM model structure is adopted to model the alternative hypothesis, they are commonly named as *anti-models*. Some significant successes have been made in using *anti-models* to model the alternative hypothesis when *anti-models* are trained from either discriminative training procedure as shown in [20], [23], or some smart data selection procedure as in [9]. On the other hand, if we study utterance verification problems from a Bayesian viewpoint, the final solution ends up with evaluating the so-called *Bayes factors* [10]. As shown in [10], *Bayes factors* is a powerful tool to model composite hypotheses, which can be used to solve many different verification problems. The same speaker verification method in [10] is also equally applicable to UV problems. As far as the relation between CMs and UV is concerned, as discussed in [18], there is a close link between CM and UV. Any verification scores in UV can be transformed into a CM score. Basically, in the speech recognition area, as we have known so far, the most effective way to measure the reliability (or confidence) of any recognition decision is mainly based on how much the underlying decision can overtake other possible competitors. The larger the difference is the more confident we will consider the decision to be. The various CM or UV methods attempt to explore this discrepancy in different ways (direct or indirect). For example, in the posteriori probability method based on a word graph, if the recognition result significantly overtakes other competing choices in the word graph, the contribution of the recognized path will dominate the total posteriori probability computed based on forward-backward algorithm. In this case, the derived CM (i.e., the normalized posteriori probability) will be large (close to 1). If other competing paths in the word graph come very close to the recognized results, the contribution of the recognized path will be relatively small when computing posteriori. Thus, the derived CM will be small (close to 0). Similar in UV, if the recognized result overtakes other competitors, the likelihood under the null hypothesis will be significantly larger than that of the alternative hypothesis. As a result, the likelihood ratio will be large. On the other hand, the likelihood ratio will be small if the competing sources from the alternative hypothesis gives comparable results with the recognized one in the null hypothesis. Therefore, it becomes very important to know the properties of competing source distributions in order to optimize the performance of utterance verification or confidence measures. In this paper, we are going to investigate a novel idea to perform utterance verification based on neighborhood information in model space. We first introduce a structure of "nested-neighborhoods" around the underlying model in model space. Then we conceptually explain the physical meaning of these nested neighborhoods with different sizes and we argue that one of these neighborhoods with a properly-selected size includes all possibly potential competing models of the original underlying model. Then we will also show how to use these "nested neighborhoods" to compute confidence scores for utterance verification. In this work, *Bayes factors* serves as a major computing vehicle to implement this idea. In order to examine the viability of the proposed approach, we have applied it to recognition error detection in the Bell Labs Communicator system, where we verify correct words against mis-recognized words in the decoder's outputs. The experimental results show that the new method has dramatically improved verification performance, relatively more than 20% reduction in equal error rate (EER) when comparing with the standard approach, which is likelihood ratio testing based on anti-models.

The remainder of the paper is organized as follows. At first, in Section II, we introduce the structure of nested neighborhoods in model space and how to use the neighborhood information to perform utterance verification. Next, in Section III, we briefly review how to use Bayes factors as a general tool to do statistical hypothesis testing. Then, based on the above general formulation, we will investigate two particular neighborhood definitions for HMM, a parametric neighborhood in Section IV and a nonparametric one in Section V respectively. In Section VI, we will report our experimental results on the Bell Labs communicator system, where we verify mis-recognized words against correctly recognized words in ASR outputs. Finally, we conclude the paper with our findings in Section VII.

II. UV Based on Neighborhood Information

A. Nested Neighborhoods in Model Space

First of all, let us look at the model space $\mathcal{T}$ of HMM. Suppose we have N different HMMs in the recognizer, denoted as $\{\lambda_i | 1 \le i \le N\}$. Each λ_i can be viewed as a point in the model space $\mathcal{T}$. Intuitively, for every given model λ_i, we are able to enumerate a set of nested neighborhoods in $\mathcal{T}$ which are all surrounding the underlying model λ_i. For a given model λ_i, we can define a set of nested neighborhoods $\Lambda_0^{(i)}, \Lambda_0^{(i)}, \cdots$, with increasing neighborhood sizes as follows:

1) **Zero neighborhood** $\Lambda_0^{(i)}$: $\Lambda_0^{(i)}$ consists of the center λ_i only.
2) **Tight neighborhood** $\Lambda_1^{(i)}$: $\Lambda_1^{(i)}$ is a very small neighborhood which tightly surrounds the model λ_i. As indicated in [6], this kind of neighborhood serves as a robust representation of the original model λ_i. In other words, due to estimation errors and any mismatched condition during testing phase, the optimal model for any given test utterance could slightly shift from the original position of the estimated model, but it generally is considered that the optimal model still resides somewhere within $\Lambda_1^{(i)}$ since the resultant model shift can not be too large.
3) **Medium neighborhood** $\Lambda_2^{(i)}$: $\Lambda_2^{(i)}$ has a medium size and is significantly larger than $\Lambda_1^{(i)}$. Thus, $\Lambda_2^{(i)}$ possibly includes all of λ_i's potential competing models, which are by definition close to λ_i in model space, no matter whether they are used by the recognizer or not.
4) **Large neighborhood** $\Lambda_3^{(i)}$: $\Lambda_3^{(i)}$ is even larger in size and should cover all related speech models in model space. Because the size of $\Lambda_3^{(i)}$ is much larger than the distance among all models $\{\lambda_i | 1 \le i \le N\}$, the large neighborhood of different models λ_i should overlap with each other. On the other hand, a different model λ_i should have its own $\Lambda_0^{(i)}$, $\Lambda_1^{(i)}$ and $\Lambda_2^{(i)}$.
5) **Infinity neighborhood** $\Lambda_4^{(i)}$: $\Lambda_4^{(i)}$ has an infinity size and it actually covers the entire model space. Therefore, $\Lambda_4^{(i)}$ should include all models in model space which represents nonspeech events. In concept, these models are far away from the original model λ_i.

The whole picture is illustrated in Fig. 1. A neighborhood with a relatively small size, $\Lambda_1^{(i)}$, contains all variants of the original model due to estimation errors and possible mismatches in testing. As the neighborhood size increases further, it starts to cover all of its competing models in the model space, which by definition should be close to the original model in some sense. Then a larger neighborhood can include all meaningful models in the model space, i.e., $\Lambda_3^{(i)}$. Eventually it can cover the whole model space, like $\Lambda_4^{(i)}$.

B. UV Based on Neighborhood Information

In utterance verification, we usually have several different scenarios, e.g., to detect recognition errors or to reject out-of-vocabulary words or to reject no-speech noises. Based on the above flexible "nested-neighborhood" structure, we will be able to select some neighborhoods with different sizes for different verification purposes. In this work, we only concentrate on the first scenario, namely detecting recognition errors in ASR.

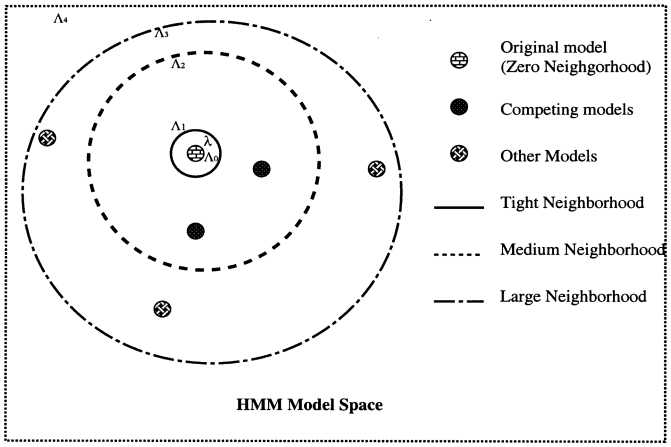

Fig. 1. Illustration of the structure of nested neighborhoods in HMM model space.

For a given speech segment X, assume that an ASR system recognizes it as word W which is represented by an HMM model λ_W. We are interested in examining the reliability of this decision in order to accept or reject it. Under the framework of UV, we usually formulate it as a statistical hypothesis testing problem. We test the *null* hypothesis H_0 against the alternative hypothesis H_1 as

$$\begin{aligned} H_0 &: X \text{ is truly from model } \lambda_W \\ H_1 &: X \text{ is NOT from model } \lambda_W. \end{aligned} \tag{2}$$

The major difficulty with this conventional hypothesis testing is that it is quite hard to model the alternative hypothesis H_1 which obviously is composite and not well-defined.

Given the decision that X is recognized as model λ_W, if X is not from the model λ_W, it is reasonable to consider that X probably comes from some competing model of λ_W. Based on the concept described in Section II-A, we define two nested neighborhoods in model space around the underlying model λ_W: i) tight neighborhood Λ_1: as a robust representation of the original model λ_W; ii) medium neighborhood Λ_2: including all potential competing models of λ_W. Therefore, based on the above discussions, we can translate the above hypothesis testing (H_0 vs. H_1) into the following ones:

$$\begin{aligned} \mathbf{H_0'} &: \text{The true model of } X \text{ lies in the} \\ &\quad \text{tight neighborhood } \Lambda_1 \\ \mathbf{H_1'} &: \text{The true model of } X \text{ lies in the} \\ &\quad \text{region } \Lambda_2 - \Lambda_1 \end{aligned} \tag{3}$$

where $\Lambda_2 - \Lambda_1$ denotes the holed region inside medium neighborhood Λ_2 but excluding tight neighborhood Λ_1, as shown in Fig. 2. Now we formulate utterance verification as a new hypothesis testing problem where we verify $\mathbf{H_0'}$ against $\mathbf{H_1'}$ to decide the reliability of the original recognition result. Note that here both hypotheses $\mathbf{H_0'}$ and $\mathbf{H_1'}$ are composite which makes it hard to solve this verification problem under the traditional

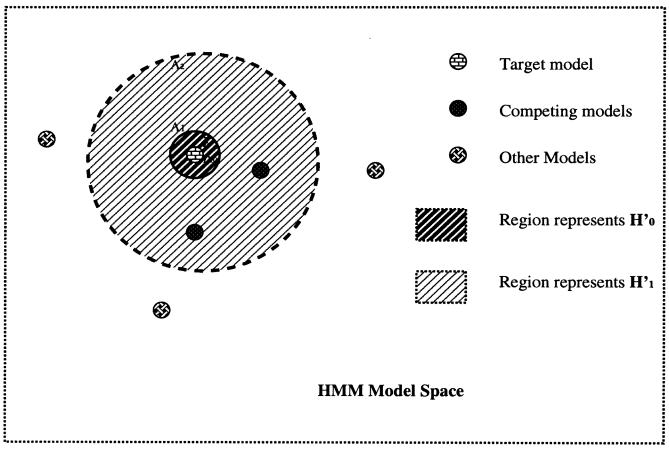

Fig. 2. Illustration of hypothesis testing in the scenario of detecting speech recognition errors based on the neighborhood information.

framework of likelihood ratio testing (LRT). Besides LRT, there are several other tools available to solve verification problems. In this paper, we will investigate how to use *Bayes factors* to solve the above hypothesis testing problem.

III. Bayes Factors: A Bayesian Tool for Verification Problems

Bayes factors has its solid foundation from Bayesian theory. As shown in [15], the Bayesian approach to hypothesis testing involves the calculation and evaluation of the so-called *Bayes factors*. Given the observation data X along with two hypotheses H_0 and H_1, Bayes factors is computed as

$$BF = \frac{\hat{p}(X|H_0)}{\hat{p}(X|H_1)} = \frac{\int f(X|\lambda_0, H_0) \cdot p(\lambda_0|H_0)\mathrm{d}\lambda_0}{\int f(X|\lambda_1, H_1) \cdot p(\lambda_1|H_1)\mathrm{d}\lambda_1} \tag{4}$$

where, for $k = 0, 1$, λ_k is the model parameter under H_k, $p(\lambda_k|H_k)$ is its prior density, and $f(X|\lambda_k, H_k)$ is the likelihood function of λ_k under H_k.

Bayes factors offers a way to evaluate evidence in favor of the *null* hypothesis H_0 because the Bayes factors is the ratio of the posterior odds of H_0 to its prior odds, regardless of the value of the prior odds [15].[1] Therefore, Bayes factors can be used to compare with a threshold, just like the likelihood ratio in Neyman-Pearson lemma, to make a decision with regards to H_0. In other words, if $BF > \tau$, where τ is a pre-set critical threshold, then we accept H_0, otherwise reject it.

In the above, we have presented a general framework to perform utterance verification based on neighborhood information in HMM model space. As shown in Fig. 2, according to certain distance measurement between two HMMs conformable with the decision rules used by speech recognition, we can define two nested neighborhoods around the underlying model. The small one is viewed as a robust representation of the original model and it contains all possible variants from the original model due to mismatches and other estimation errors [6]. On the other hand, the large neighborhood includes all potential competing models of the original model. And Bayes factors becomes an ideal computation tool to calculate the average contributions from all models in these two neighborhoods. However, in order to use Bayes factors to solve the hypothesis testing problem, i.e., $\mathbf{H_0'}$ vs. $\mathbf{H_1'}$ in (3), two important issues must be addressed first: i) how to quantitatively define neighborhoods Λ_1 and Λ_2; ii) how to properly choose prior distribution $p(\cdot)$ of HMM model parameter for each hypothesis. Because of high dimension in HMM model space, it is not straightforward to define a proper neighborhood form for HMM. In this paper, we have investigated two different ways to define neighborhood form in HMM model space: i) Parametric (C, ρ) neighborhood, previously used for robust speech recognition [6], [21], where two parameters C and ρ must be specified manually beforehand to control the size and shape of the neighborhood. In this case, a uniform distribution is chosen as prior p.d.f. to compute Bayes factors; ii) Non-parametric neighborhood: we plot all HMM models of the recognizer in the model space and adjust the neighborhood size to include a certain number of models. In this case, a mixture delta function is selected as the prior distribution.

[1]Any probability can be converted to the odds scale, i.e., $\mathrm{odds} = \mathrm{probability}/(1 - \mathrm{probability})$. Thus, $(Pr(H_0|y))/(Pr(H_1|y))$ is called the posterior odds in favor of H_0, and $(Pr(H_0))/(Pr(H_1))$ is prior odds in favor of H_0.

IV. Case I: (C, ρ) Neighborhood and Constrained Uniform Priors

Assume each HMM λ is an N-state continuous density HMM (CDHMM) with parameter vector $\lambda = (\pi, A, \theta)$, where π is an initial state distribution, A is a state transition matrix, and θ is a parameter vector composed of mixture parameters $\theta = \{\theta_i | i = 1, 2, \cdots, N\}$, where θ_i denotes the parameters of i-th state in HMM. θ_i consists of several Gaussian mixtures, i.e., $\theta_i = \{\omega_{ik}, m_{ik}, r_{ik}\}(k = 1, 2, \cdots, K)$, where k indicates the mixture number in the state. The state observation p.d.f. is assumed to be a mixture of multivariate Gaussian distributions with diagonal precision matrix

$$p(\mathbf{x}|\theta_i) = \sum_{k=1}^{K} \omega_{ik} \prod_{d=1}^{D} \sqrt{\frac{r_{ikd}}{2\pi}} e^{-\frac{1}{2} r_{ikd}(x_d - m_{ikd})^2} \tag{5}$$

where the mixture weights ω_{ik}'s satisfy the constraint $\sum_{k=1}^{K} \omega_{ik} = 1$.

At first, following the work in [21], we define the neighborhood form for both Λ_1 and Λ_2 as

$$\begin{aligned} \Lambda(\lambda) = \{\lambda | \pi = \pi^*,\ A = A^*,\ \omega_{ik} = \omega_{ik}^*, r_{ik} = r_{ik}^*, \\ |m_{ikd} - m_{ikd}^*| \le C d^{-1} \rho^d,\ 1 \le i \le N, \\ 1 \le k \le K,\ 1 \le d \le D\} \end{aligned} \tag{6}$$

where $\lambda^* = \{\pi^*, A^*, \omega_{ik}^*, r_{ik}^*, m_{ikd}^*\}$ denotes the original model parameter which is the central point of the neighborhood, and $C(C > 0)$ and $\rho(0 \le \rho \le 1)$ are used to control the shape and size of the neighborhood. As shown in [21], the neighborhood in (6) is derived based on an upper bound in cepstral domain of any small perturbation of speech signals. The parameter C is used to control the absolute neighborhood size across all vector dimensions. The acceptable dynamic region for C in speech recognition is usually [1, 10]. The parameter ρ is an exponential shrinking scale used to reflect the shrinkage

of dynamic range of speech cepstral in high dimensions. The acceptable dynamic range of ρ in speech recognition is [0.1, 0.9]. Generally speaking, the larger the absolute values of C and ρ are, the larger the size of the underlying neighborhood will be. For medium neighborhood Λ_2, we choose larger values for C and ρ. And for tight neighborhood Λ_1, we choose smaller values for C and ρ. Secondly, given the neighborhood, we assume that the prior distribution of HMM parameter is a uniform p.d.f. constrained in the neighborhood. Based on these assumptions, the calculation of Bayes factors can be simplified as

$$BF_1 = \frac{\hat{p}_1(X)}{\hat{p}_2(X)} = D \cdot \frac{\int_{\Lambda_1} f(X|\lambda)\mathrm{d}\lambda}{\int_{\Lambda_2-\Lambda_1} f(X|\lambda)\mathrm{d}\lambda} = D \cdot \frac{\int_{\Lambda_1} f(X|\lambda)\mathrm{d}\lambda}{\int_{\Lambda_2} f(X|\lambda)\mathrm{d}\lambda - \int_{\Lambda_1} f(X|\lambda)\mathrm{d}\lambda} \tag{7}$$

where $D = \int_{\Lambda_2-\Lambda_1} \mathrm{d}\lambda / \int_{\Lambda_1} \mathrm{d}\lambda$ is the normalization factor. Obviously, Bayes factors is a ratio between two Bayesian predictive densities so that we can calculate numerator and denominator separately. The VBPC (Viterbi Bayesian predictive classification) algorithm in [6] is used to compute each Bayesian predictive density $\hat{p}(X)$, e.g., $\hat{p}_1(X) = \int_{\Lambda_1} f(X|\lambda)\mathrm{d}\lambda$ and $\hat{p}_2(X) = \int_{\Lambda_2-\Lambda_1} f(X|\lambda)\mathrm{d}\lambda = \int_{\Lambda_2} f(X|\lambda)\mathrm{d}\lambda - \int_{\Lambda_1} f(X|\lambda)\mathrm{d}\lambda$.

Given an utterance $X = \{x_1, x_2, \cdots, x_T\}$, under the above definition of prior distribution, the Bayesian predictive density $\hat{p}(X)$ for the neighborhood Λ is computed as follows:

$$\hat{p}(X) = \int_{\Lambda} f(X|\lambda)\mathrm{d}\lambda = \int_{\Lambda} \sum_{s,l} f(X,s,l|\lambda)\mathrm{d}\lambda = \sum_{s,l} \int_{\Lambda} f(X,s,l|\lambda)\mathrm{d}\lambda \approx \max_{s,l} \int_{\Lambda} f(X,s,l|\lambda)\mathrm{d}\lambda \tag{8}$$

where s and l denote a state sequence and mixture component label sequence corresponding to X. We term the path s and l which maximize this integral as the optimal path, denoted as $\{\bar{s}, \bar{l}\} = \{\bar{s}_1, \cdots, \bar{s}_T, \bar{l}_1, \cdots, \bar{l}_T\}$, i.e.,

$$\{\bar{s}, \bar{l}\} = \arg\max_{s,l} \int_{\Lambda} f(X,s,l|\lambda)\mathrm{d}\lambda. \tag{9}$$

Given the input utterance X, the underlying CDHMM λ, and uniform prior distribution in the neighborhood, the optimal path $\{\bar{s}, \bar{l}\}$ can be obtained by using the VBPC recursive search algorithm described in [6]. The VBPC algorithm is a modified version of the Viterbi search in speech recognition to implement Viterbi approximation of calculating Bayesian predictive density in (8). In the VBPC algorithm, for every time instant, the Bayesian predictive density, i.e., the integral in (8), is calculated for all active partial paths survived in the search. Then, similar to the Viterbi search, all partial paths are propagated in the search network and their Bayesian predictive densities are re-computed until the end of the utterance. In this way, we can get an approximate method to calculate Bayesian predictive density for HMM. Meanwhile, the optimal paths $\{\bar{s}, \bar{l}\}$ can also be obtained by backtracking search results. The readers can refer to [6] for more details about VBPC algorithm. In this paper, in order to balance contribution from different models in the neighborhood, we introduce an exponential scale factor $\alpha(\alpha > 0)$ into the integral calculation. The exponential scale factor α is important to equalize the contributions from different models in the neighborhood during the computation of Bayes factors. For example, if we choose $\alpha > 1$, the models with large likelihood values are emphasized. On the other hand, if $\alpha < 1$, the models with smaller likelihood values will be put more weights. Therefore, given the optimal path $\{\bar{s}, \bar{l}\}$, the approximate Bayesian predictive density $\hat{p}(X)$ is computed as follows:

$$\begin{aligned}\hat{p}(X) &\approx \left\{ \int_{\Lambda} \left[f(X, \bar{s}, \bar{l}|\lambda) \right]^{\alpha} \mathrm{d}\lambda \right\}^{\frac{1}{\alpha}} \\ &= \prod_{i=1}^{N} \prod_{k=1}^{K} \prod_{d=1}^{D} \left(\frac{r^*_{ikd}}{2\pi} \right)^{\frac{n_{ik}}{2}} \frac{e^{-\frac{1}{2} r^*_{ikd} \left(\overline{x^2_{ikd}} - \bar{x}^2_{ikd} \right)}}{2Cd^{-1}\rho^d} \\ &\quad \times \left\{ \sqrt{\frac{2\pi}{\alpha r^*_{ikd} n_{ik}}} \right. \\ &\quad \times \left[\Phi\left(\sqrt{\alpha r^*_{ikd}} n_{ik} \left(m^*_{ikd} - \bar{x}_{ikd} + Cd^{-1}\rho^d \right) \right) \right. \\ &\quad - \Phi\left(\sqrt{\alpha r^*_{ikd} n_{ik}} \right. \\ &\quad \left. \left. \left. \times \left(m^*_{ikd} - \bar{x}_{ikd} - Cd^{-1}\rho^d \right) \right) \right] \right\}^{\frac{1}{\alpha}} \end{aligned} \tag{10}$$

where

$$n_{ik} = \sum_{t=1}^{T} \delta(\bar{s}_t - i) \cdot \delta(\bar{l}_t - k) \tag{11}$$

$$\bar{x}_{ikd} = \frac{1}{n_{ik}} \sum_{t=1}^{T} x_{td} \cdot \delta(\bar{s}_t - i) \cdot \delta(\bar{l}_t - k) \tag{12}$$

$$\overline{x^2_{ikd}} = \frac{1}{n_{ik}} \sum_{t=1}^{T} x^2_{td} \cdot \delta(\bar{s}_t - i) \cdot \delta(\bar{l}_t - k). \tag{13}$$

In the above, $\delta(\cdot)$ denotes the Kronecker delta indicator with

$$\delta(a-b) = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}$$

and also $\Phi(y)$ represents the error function of Gaussian distribution

$$\Phi(y) = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{y} e^{-\frac{x^2}{2}} dx. \tag{14}$$

The only issue left here is how to specify the parameters (C, ρ) for various neighborhoods. In this work, we propose the following two different methods.

- **Global setting**: We manually select (C_1, ρ_1) for tight neighborhood Λ_1 and (C_2, ρ_2) for medium neighborhood Λ_2, where $C_1 \leq C_2$ and $\rho_1 \leq \rho_2$. In this case, we use the same tight (or medium) neighborhood for all different states and Gaussian mixtures in all HMMs.

- **State-dependent setting**: Given a HMM state with parameter θ_i, we assume the neighborhood for this state θ_i is defined as in (6), we calculate the maximum deviation from the central point within the neighborhood in terms of euclidean distance as

$$\mathcal{D} = C^2 \cdot \prod_{d=1}^{D} \left[(d^{-1}\rho^d)^2 \cdot \max_{k=1}^{K} r_{ikd} \right]. \tag{15}$$

In reverse, when fixing ρ, we can calculate C based on a deviation distance $\mathcal{D}$ from the above equation. Thus, we first manually select ρ_1 and ρ_2 for tight neighborhood Λ_1 and medium one Λ_2, then we define maximally allowed deviation distances for tight and medium neighborhoods, i.e., $\mathcal{D}_1$ and $\mathcal{D}_2$ (usually $\mathcal{D}_1 < \mathcal{D}_2$), finally we can calculate C_1 and C_2 for different HMM states from $\mathcal{D}_1$ and $\mathcal{D}_2$ based on (15). In this case, we can use different tight (or medium) neighborhoods for different states in all HMMs.

V. Case II: Delta Priors

Given a model λ^*, we still define two neighborhoods around λ^*: tight neighborhood Λ_1 and medium neighborhood Λ_2. Then for each neighborhood, to say Λ_1, we construct a prior distribution as a mixture of delta functions. These delta functions are centered at other models in the recognizer, which are located inside neighborhood Λ_1. That is,

$$\rho_1 = \frac{1}{N_t} \sum_{\lambda_i \in \Lambda_1} \delta(\lambda - \lambda_i) \tag{16}$$

where N_t denotes the total number of models inside Λ_1.

Similarly, we can build a prior distribution for the region $\Lambda_2 - \Lambda_1$

$$\rho_2(\lambda) = \frac{1}{N_m} \sum_{\lambda_i \in \Lambda_2 - \Lambda_1} \delta(\lambda - \lambda_i) \tag{17}$$

where N_m denotes the total number of models located in the region $\Lambda_2 - \Lambda_1$.

Based on these two priors, Bayes factors to verify hypotheses $\mathbf{H_0'}$ and $\mathbf{H_1'}$ in (3) can be simplified as

$$BF_2 = \frac{\sum_{\lambda_i \in \Lambda_1} \frac{f(X|\lambda_i)}{N_t}}{\sum_{\lambda_i \in \Lambda_2 - \Lambda_1} \frac{f(X|\lambda_i)}{N_m}}. \tag{18}$$

To balance the contribution from different models, we can similarly introduce a scale factor $\alpha(\alpha > 0)$ in the above summation. Then we have

$$BF_2 = \frac{\left\{ \sum_{\lambda_i \in \Lambda_1} \frac{[f(X|\lambda_i)]^\alpha}{N_t} \right\}^{\frac{1}{\alpha}}}{\left\{ \sum_{\lambda_i \in \Lambda_2 - \Lambda_1} \frac{[f(X|\lambda_i)]^\alpha}{N_m} \right\}^{\frac{1}{\alpha}}}. \tag{19}$$

At present, in many large-scale ASR systems, we usually use state-tied CDHMMs as fundamental acoustic models for speech recognition, which basically consists of a pool of distinct HMM states, from which all HMMs in the system share their states. Therefore, instead of building delta priors for each HMM model, we also can set up delta priors in the level of HMM states. In other words, for all distinct HMM states in the recognizer, we can build the above delta priors separately for each state by using the delta functions centering at all other HMM states in the pool. Then Bayes factors is calculated for each state segment independently and these scores are combined to obtain a verification score for the HMM model, each recognized word, or even the entire utterance, for the final utterance verification purpose.

VI. Experiments

To examine the viability of the proposed methods, we evaluate them on the Bell Labs Communicator system [19] to detect recognition errors in final recognition results from the decoder. The Bell Labs Communicator system is a travel reservation system developed at Bell Labs under the sponsor of the DARPA Communicator project. Users can talk and negotiate with the system through telephone line with free spontaneous speech to make their own travel plans, such as flight tickets reservation, hotel booking, etc. In this work, we consider how to reliably detect recognition errors, such as city-name and other keywords, from the recognition output of the decoder. The detection results and the resultant confidence scores will be used to facilitate and improve the performance of speech understanding and dialogue management in the later stages. In our experiments, the newly proposed utterance verification method is compared with the traditional method, i.e., likelihood ratio testing (LRT) based on standard anti-models, which are trained from training data with the fixed segmentation (generated from forced alignment against reference transcriptions).

A. Baseline System

In our recognition system, we used a 38-dimension feature vector, consisting of 12 Mel LPCCEP, 12 delta CEP, 12 delta-delta CEP, delta and delta-delta log-energy. In the baseline system, the best acoustic HMMs are trained by using the standard Baum-Welch ML estimation on a total of 46 hours of task-dependent speech data. The acoustic models are state-tied, tri-phone CDHMM models, which consist of roughly 4K distinct HMM states with an average of 13.2 Gaussian mixture components per state. Besides, a class-based, tri-gram language model including 2600 words is used for decoding in the system. The baseline system achieves 15.8% word error rate (WER) in our independent evaluation set, which includes in total 1395 utterances. In the experiments, we are interested in detecting recognition errors from the decoder's outputs. We verify correctly recognized words against mis-recognized words (only substitution and insertion errors). The recognition result is first aligned against the reference transcription by a standard dynamic programming procedure to label each recognized word as either correct or wrong. In all recognizer's recognition outputs of the evaluation set, we totally have 3257 words labeled as *correct* and 520 words as *wrong* (not including deletion errors). Based on the word and phoneme segmentations generated by the recognizer, we calculate a confidence score for every recognized word. According to the computed confidence scores, we verify all correctly recognized words against other mis-recognized words. As our baseline verification system, we

use LRT based on a standard mono-phone *anti-models*, which are trained from all training data with fixed forced alignment phoneme segmentation, generated from the standard forced alignment procedure. That is, all phone segments of a phone are collected to train the positive model for this phone and all other phone segments are used to train an anti-model for this phone. As shown in the first row of Table I, we achieve 40.0% equal error rate (EER) with this standard method in our evaluation data set. The ROC curve is also shown in Fig. 3 with the label *baseline*.

B. New Approach With Settings in Case I

In this section, we first investigate the Bayes factors methods described in Section IV where we choose (C, ρ) neighborhood and constrained uniform prior distribution for Bayesian calculation. For all recognition results from the decoder, based on the segmentation information, we calculate Bayes-factors-based score BF_1 for every phoneme segment as shown in (7). Then these phone-level scores are combined to get the average score per frame for each recognized word. Based on these scores, we repeat the same UV experiments as in the baseline verification system. Since we use static, delta, and delta-delta features, following [6], [8], we slightly modify the (C, ρ) neighborhood definition in (6) to suit with delta and delta-delta features as

$$\begin{aligned}\Lambda(\lambda) = \{\lambda | &\pi = \pi^*,\ A = A^*,\ \omega_{ik} = \omega_{ik}^*,\ r_{ik} = r_{ik}^*,\\ &|m_{ikd} - m_{ikd}^*| \le Cd^{-1}\rho^d,\ 1 \le i \le N,\\ &\left|m_{ik(\frac{D}{3+d})} - m_{ik(\frac{D}{3+d})}^*\right| \le Cd^{-1}\rho^d,\\ &\left|m_{ik(\frac{2D}{3+d})} - m_{ik(\frac{2D}{3+d})}\right| \le Cd^{-1}\rho^d,\\ &1 \le i \le N,\ 1 \le k \le K,\ 1 \le d \le \frac{D}{3}\Big\} \end{aligned} \tag{20}$$

where for $1 \le d \le D/3$, m_{ikd}'s correspond to the static feature part, $m_{ik(D/3+d)}$'s the delta feature part and $m_{ik(2D/3+d)}$'s the delta-delta feature part. Obviously, in the above definition, we use the same neighborhood bound for static, delta and delta-delta features.

In the case of global setting, we manually check the range: $C_1, C_2 \in [1.0, 10.0] (C_1 \le C_2)$ and $\rho_1, \rho_2 \in [0.1, 0.9] (\rho_1 \le \rho_2)$. The best performance as well as its corresponding setting is shown in the second row of Table I. We can see that the verification score BF_1 based on global setting obtains 36.7% in term of EER, which is slightly better than our LRT-based baseline performance (40.0% of EER). The performance is limited because in this case we are using the same neighborhoods for all different HMM states.

As for the state-dependent setting, we first set up ρ_1 to a small value, to say 0.1, and ρ_2 to a large value, to say 0.9. According to (15), once we fix the maximum deviation distance $\mathcal{D}$, we can derive the parameter C for different HMM states from (15), where only static feature components are considered in this calculation. In this case, we have manually checked the range $\mathcal{D}_1 \in [0, 10.0]$, and $\mathcal{D}_2 \in [100.0, 250.0]$. The best performance is shown in the third row of Table I, where we achieve 32.4% in terms of EER using the state-dependent setting BF_1. This is a big improvement from the global setting. One possible reason why state-dependent setting gives much better performance is that we can use different neighborhood sizes for different HMM states based on certain deviation distance measurement. Another reason is that it is much easier and more accurate to specify the neighborhood size by using the distance deviation introduced in (15) rather than using the original control parameters C and ρ. For both settings, we also plot the ROC curves in Fig. 3, which also clearly show that both methods significantly outperform the baseline system.

TABLE I
VERIFICATION PERFORMANCE COMPARISON (EQUAL ERROR RATE IN %) OF BASELINE UV METHOD (LRT + ANTI-MODELS) WITH THE PROPOSED NEW APPROACH IN SEVERAL DIFFERENT SETTINGS. IN EACH CASE, THE BEST PERFORMANCE OF THE NEW APPROACH AND ITS CORRESPONDING PARAMETER SETTING ARE GIVEN. HERE WE ALWAYS FIX $\alpha = 1.2$

method	EER	parameter setting
anti-model	40.0	–
caseI-global	36.7	$\rho_1 = 0.1,\ \rho_2 = 0.7,\ C_1 = C_2 = 3.0$
caseI-state	32.4	$\rho_1 = 0.1,\ \rho_2 = 0.9,\ \mathcal{D}_1 = 3,\ \mathcal{D}_2 = 200$
caseII-A	31.5	$N_t = 1,\ N_m = 1000$
caseII-B	31.0	$\mathcal{D}_t = 12.5,\ \mathcal{D}_m = 40.0$

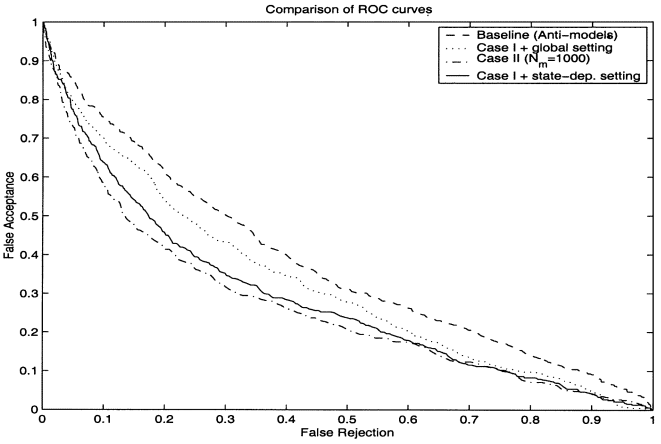

Fig. 3. Comparison of ROC curves for different methods when verifying mis-recognized words against correctly recognized words in ASR outputs.

C. New Approach With Settings in Case II

In this part, we choose delta priors in (16) and (17) in the level of HMM state for Bayes factors. At first, for each distinct state in acoustic models, we calculate its distance from all other states (roughly 4K). The distance between two HMM states is computed as the minimum euclidean distance between every possible pair of Gaussian components from these two states. For each state, we sort all other states according to their distances from the underlying state. Then we save the sorted list and all corresponding distances for every state in order to determine the sizes of neighborhoods. In the first case, denoted as *Case II-A*, for each underlying HMM state, we choose neighborhood sizes to include exactly N_t other states in Λ_1 and N_m in $\Lambda_2 - \Lambda_1$. In the second case, denoted as *Case II-B*, from the top 1500 states, we choose neighborhood sizes for Λ_1 to include all other

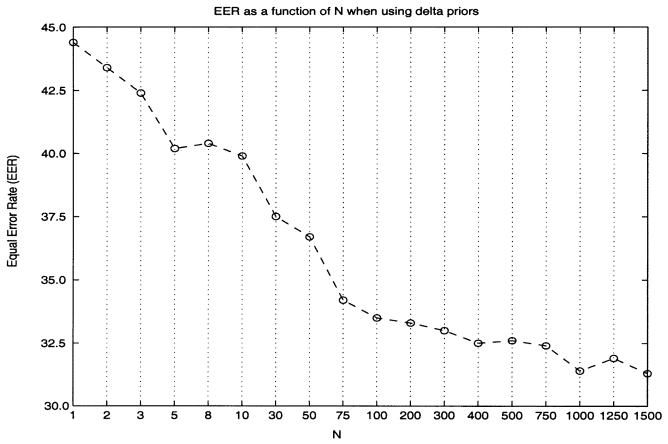

Fig. 4. Verification performance, in terms of EER (in %), of the method CaseII-A is shown as a function of parameter N_m when we fix $N_t = 1$ and $\alpha = 1.2$.

states with distance less than $\mathcal{D}_t$ and ones' distance between $\mathcal{D}_t$ and $\mathcal{D}_m$ for $\Lambda_2 - \Lambda_1$. In Fig. 4, we plot the verification performance in term of EER as a function of the mixture number N_m in large neighborhood Λ_2 when we fix the number of other states in small neighborhoods Λ_1, i.e., $N_t = 1$ and the scale factor $\alpha = 1.2$. From the result, we can see that the verification performance significantly improves as N_m increases. But it requires a pretty large number, e.g., $\geq$1000, to achieve a reasonably good result. In the last part of Table I, we also give verification performance in terms of EER, 31.5% for *Case II-A* and 31.0% *Case II-B*. Generally speaking, delta prior gives a slightly better performance than uniform distribution based on (C, ρ). However, because it usually requires a very large number of delta components, the computational complexity is much more expensive than the parametric neighborhood in Case I. Finally, we also plot the ROC curve for *Case II-A* (with $N_m = 1000$) in Fig. 3, which clearly shows that it gives the best verification performance.

D. Discussions

Theoretically, the exponential scale factor α is important to equalize the contributions from different models in the neighborhood when calculating Bayes factors. If we choose $\alpha > 1$, the models with larger likelihood value are emphasized. On the other hand, if $\alpha < 1$, the models with smaller likelihood value will be put more weights. However, in our experiments, we find the verification performance is not very sensitive to α in a certain range, to say [0.6, 1.5]. Thus, in our above experiments, we have fixed $\alpha = 1.2$, which gives a slightly better performance.

As for the parametric neighborhood parameters C and ρ, similar to some previous works [6], [21], the use of (C, ρ) neighborhood suffers from the difficulty of how to automatically determine the control parameters C and ρ. The general behavior is that the method usually performs well in a certain range of (C, ρ), usually [1, 10] for C and [0.1, 0.9] for ρ. But optimal values of C and ρ usually depend on the particular data set. In Table I, we give the best performance and its corresponding parameter setting in the test set. For other C and ρ values in the above region, the performance will be slightly worse. To overcome this difficulty, we have proposed a nonparametric neighborhood (case II) in this paper. In this case, it is much easier to specify parameters. For example, how many other models are included in the neighborhood. As shown in Fig. 4, the larger the number is, the better the performance will be (of course, at expense of more computational cost). Generally, if N is larger than 500, the performance will be reasonably good for different data sets.

When we compare two investigated neighborhood definitions, namely parametric method (case I) and nonparametric one (case II), based on the above experiments, we can see that the nonparametric method (case II) slightly outperforms the parametric method (case I) in case we use a large number of delta mixture components in the nonparametric method. But the nonparametric method requires much higher computational complexity to calculate confidence scores because a large number of mixture components are involved in the delta prior p.d.f., where we must repeat likelihood calculation for every component in summation calculation of Bayes factors. On the other hand, the parametric definition of neighborhood (or the priors) is cheaper in terms of computational complexity because an approximate closed-form solution usually is available to the integral calculation in Bayes factors. The problem is that it is usually very hard to define a proper parametric neighborhood (or prior distribution) for HMM models due to the high dimension in HMM model space.

If we compare the proposed methods with the traditional anti-model-based LRT approach, the new method is simple to use because it does not require to build separate verification models. Once the neighborhood is defined, we can calculate confidence score immediately. In contrast, in LRT-based approaches, we usually have to follow a very complicated training procedure in order to obtain some reasonable verification models, such as in [9], [20]. When comparing with other confidence measures which are proved to work reasonably well, such as word-graph-based posterior methods [27], the neighborhood-based approach is much faster and instant. In other words, in [27], in order to obtain confidence scores of certain recognized words from a word graph, we have to wait until the whole recognition process ends, and then build a word graph for confidence measurement. The confidence score calculation based on the word graph is also very complex and relatively computationally expensive. On the other hand, the neighborhood-based approach can calculate the confidence score as soon as a recognition decision is made. For instance, like in [9], the new method can even be used to calculate confidence scores for all partial paths during the Viterbi search procedure. Based on these scores, we will be able to prune out some very unlikely hypotheses during the process of Viterbi search for the purpose of speed and accuracy [9]. Even though the lattice-based posterior probability could also be calculated during the search procedure once an intermediate recognition is made, backtracking a lattice and computing the posterior probability over lattice are both complicated and expensive when comparing with the methods proposed in this work (as was pointed out by one reviewer).

VII. Conclusions

In this work, we have examined how to perform utterance verification based on neighborhood information in model space. The basic idea is to assume that all competing models

of a given model sit inside one neighborhood of the underlying model. Based on definition of the neighborhood, Bayes factors is adopted as a major computation vehicle to calculate confidence measures for utterance verification. In this paper, we have investigated two particular neighborhood definitions: i) Parametric one: (C, ρ) neighborhood with constrained uniform prior p.d.f.; ii) Non-parametric one: mixture delta prior p.d.f. with a fixed number of mixtures included in the neighborhood. Based on these two neighborhood definitions, Bayes-factors-based confidence measures become quite easy to calculate for HMM. From the experimental results in the Bell Labs communicator system, we have found that it is a promising direction to use neighborhood information in model space to perform utterance verification for the purpose of confidence measurement. Some preliminary studies based on two particular neighborhood definition have shown that the new method works better than the traditional anti-model-based LRT approach if we define and choose the neighborhoods properly, which in turn proves that the neighborhood information is very useful in calculating confidence measures for speech recognition. In this paper, we have proposed a systematical framework to perform utterance verification based on the neighborhood information and Bayes factors. Although we have investigated two different neighborhood definitions, one parametric neighborhood and another nonparametric one, we believe much more research works are still needed to search for a better neighborhood definition in high-dimension HMM model space. As another possible research direction for future works, instead of Bayes factors, other statistical hypothesis testing tools, such as generalized likelihood ratio testing (GLRT), can also be used to implement the neighborhood based UV described in this paper. At last, in this work, the proposed methods are compared only with one traditional approach in the literature [20], [23]. It will be very interesting to compare with other popular methods, such as [16], [26], [27].

REFERENCES

[1] M. Aitkin, "Posterior Bayes factors," *J. R. Statist. Soc.*, ser. B, vol. 53, no. 1, pp. 111–142, 1991.

[2] L. Chase, "Word and acoustic confidence annotation for large vocabulary speech recognition," in *Proc. EuroSpeech-97*, Rhodes, Greece, Sept. 1997, pp. 815–818.

[3] S. Cox and R. C. Rose, "Confidence measures for the switchboard database," in *Proc. ICASSP-96*, 1996, pp. 511–514.

[4] M. Finke, T. Zeppenfeld, M. Maier, L. Mayfield, K. Ries, P. Zhan, J. Lafferty, and A. Waibel, "Switchboard April 1996 Evaluation Report," DARPA, Apr. 1996.

[5] L. Gillick, Y. Itou, and J. Young, "A probabilistic approach to confidence estimation and evaluation," in *Proc. ICASSP-97*, 1997, pp. 879–882.

[6] H. Jiang, K. Hirose, and Q. Huo, "Robust speech recognition based on Bayesian prediction approach," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 426–440, July 1999.

[7] ——, "Improving Viterbi Bayesian predictive classification via sequential Bayesian learning in robust speech recognition," *Speech Commun.*, vol. 28, no. 4, pp. 313–326, Aug. 1999.

[8] ——, "A minimax search algorithm for robust continuous speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 688–694, Nov. 2000.

[9] H. Jiang, F. Soong, and C.-H. Lee, "Data selection strategy for utterance verification in continuous speech recognition," in *Proc. Eur. Conf. Speech Communication and Technology*, Aalborg, Denmark, Sept. 2001, pp. 2573–2576.

[10] H. Jiang and L. Deng, "A Bayesian approach to the verification problem:—applications to speaker verification," *IEEE Trans. Speech Audio Processing*, vol. 9, pp. 874–884, Nov. 2001.

[11] H. Jiang and C.-H. Lee, "Utterance verification based on neighborhood information and Bayes factors," in *Proc. ICSLP-2002*, Sept. 2002.

[12] A. Gunawardana, H.-W. Hon, and L. Jiang, "Word-based acoustic confidence measures for large vocabulary speech recognition," in *Proc. ICSLP-98*, Sydney, Australia, 1998, pp. 791–794.

[13] S. Kamppari and T. Hazen, "Word and phone level acoustic confidence scoring," in *Proc. ICASSP-2000*, 2000, pp. 1799–1820.

[14] T. Kawahara, C.-H. Lee, and B.-H. Juang, "Key-phrase detection and verification for flexible speech understanding," *IEEE Trans. Speech Audio Processing*, vol. 6, pp. 558–568, Nov. 1998.

[15] R. E. Kass and A. E. Raftery, "Bayes factors," *J. Amer. Statist. Assoc.*, vol. 90, no. 430, pp. 773–795, June 1995.

[16] T. Kemp and T. Schaaf, "Estimating confidence using word lattices," in *Proc. EuroSpeech-97*, 1997, pp. 827–830.

[17] M.-W. Koo, C.-H. Lee, and B.-H. Juang, "Speech recognition and utterance verification based on a generalized confidence score," *IEEE Trans. Speech Audio Processing*, to be published.

[18] C.-H. Lee, "Statistical confidence measures and their applications," in *Proc. ICSP '2001*, Dajeong, Korea, Aug. 2001.

[19] A. Potamianos, E. Ammicht, and J. Kuo, "Dialogue management in the Bell Labs Communicator System," in *Proc. ICSLP '2000*, Beijing, China, Oct. 2000.

[20] R. Sukkar and C.-H. Lee, "Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, Nov. 1996.

[21] N. Merhav and C.-H. Lee, "A minimax classification approach with application to robust speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 90–100, 1993.

[22] A. O'Hagan, "Fractional Bayes factors for model comparison," *J. R. Statist. Soc. B*, vol. 57, no. 1, pp. 99–138, 1995.

[23] M. G. Rahim, C.-H. Lee, and B.-H. Juang, "Discriminant utterance verification for connected digits recognition," *IEEE Trans. Speech Audio Processing*, vol. 5, May 1997.

[24] T. Schaaf and T. Kemp, "Confidence measure for spontaneous speech recognition," in *Proc. ICASSP-97*, Munich, Germany, Apr. 1997, pp. 875–878.

[25] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke, "Neural-network based measures of confidence for word recognition," in *Proc. ICASSP-97*, Munich, Germany, Apr. 1997, pp. 887–890.

[26] F. Wessel, K. Macherey, and H. Ney, "A comparison of word graph and N-best list based confidence measures," in *Proc. ICASSP-2000*, 2000, pp. 1587–1590.

[27] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 9, Mar. 2001.

[28] S. Young, "Detecting misrecognitions and out-of-vocabulary words," in *Proc. ICASSP-94*, Adelaide, Australia, Apr. 1994, pp. II-21–II-24.

Hui Jiang (M'00) received the B.Eng. and M.Eng. degrees from University of Science and Technology of China (USTC), and the Ph.D. degree from the University of Tokyo, Tokyo, Japan, in September 1998, all in electrical engineering.

From October 1998 to April 1999, he worked as a Researcher in the University of Tokyo. From April 1999 to June 2000, he was with Department of Electrical and Computer Engineering, University of Waterloo, Canada as a Postdoctoral Fellow. From 2000 to 2002, he worked in Dialogue Systems Research, Multimedia Communication Research Lab, Bell Labs, Lucent Technologies Inc., Murray Hill, NJ. Since fall 2002, he has been with Department of Computer Science, York University, Toronto, ON, Canada, as an Assistant Professor. His current research interests include all issues related to speech recognition and understanding, especially robust speech recognition, utterance verification, adaptive modeling of speech, spoken language systems, and speaker recognition/verification.

Chin-Hui Lee (M'82–SM'91–F'97) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1973, the M.S. degree in engineering and applied science from Yale University, New Haven, CT, in 1977, and the Ph.D. degree in electrical engineering with a minor in statistics from University of Washington, Seattle, in 1981.

After graduation, he joined Verbex Corporation, Bedford, MA, and was involved in research on connected word recognition. In 1984, he became affiliated with Digital Sound Corporation, Santa Barbara, CA, where he engaged in research and product development in speech coding, speech synthesis, speech recognition and signal processing for the development of the DSC-2000 Voice Server. Between 1986 and 2001, he was with Bell Laboratories, Murray Hill, NJ, where he became a Distinguished Member of Technical Staff and Director of the Dialogue Systems Research Department. His research interests include multimedia communication, multimedia signal and information processing, speech and speaker recognition, speech and language modeling, spoken dialogue processing, adaptive and discriminative learning, biometric authentication, information retrieval, and bioinformatics. His research scope is reflected in a best-seller, entitled *Automatic Speech and Speaker Recognition: Advanced Topics* (Norwell, MA: Kluwer, 1996). From August 2001 to August 2002, he was a Visiting Professor at the School of Computing, The National University of Singapore. In September 2002, he joined the Faculty of School of Electrical and Computer Engineering at Georgia Institute of Technology. He has published more than 250 papers and 25 patents on the subject of automatic speech and speaker recognition.

Dr. Lee has participated actively in professional societies. He is a member of the IEEE Signal Processing Society, Communication Society, and the European Speech Communication Association. He is also a lifetime member of the Computational Linguistics Society in Taiwan. In 1991–1995, he was an associate editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING AND TRANSCATIONS ON SPEECH AND AUDIO PROCESSING. During the same period, he served as a member of the ARPA Spoken Language Coordination Committee. In 1995–1998 he was a member of the Speech Processing Technical Committee of the IEEE Signal Processing Society (SPS) and chaired the Speech TC from 1997 to 1998. In 1996 he helped promote the SPS Multimedia Signal Processing (MMSP) Technical Committee in which he is a founding member. He received the SPS Senior Award in 1994 and the SPS Best Paper Award in 1997 and 1999, respectively. In 1997, he was also awarded the prestigious Bell Labs President's Gold Award for his contributions to the Lucent Speech Processing Solutions product. More recently he was named one of the six Distinguished Lecturers for the year 2000 by the IEEE Signal Processing Society.