UTTERANCE VERIFICATION BASED ON
NEIGHBORHOOD INFORMATION AND BAYES FACTORS

Hui Jiang and Chin-Hui Lee

Dialogue Systems Research, Multimedia Communication Research Lab,
Bell Labs, Lucent Technologies, Murray Hill, NJ 07974
Email: hui@research.bell-labs.com and chl@comp.nus.edu.sg

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 Bell Labs communicator system show that the new method has dramatically improved verification performance when verifying correct words against mis-recognized words in 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.

1. INTRODUCTION

Utterance verification (UV) becomes extremely important when an automatic speech recognition (ASR) system is actually employed in a real-world application. Utterance verification is a procedure to verify how reliable speech recognition results are.[5] During the past a few years, the most significant progress in utterance verification is to cast UV as a statistical hypothesis testing problem. According to Neyman-Pearson Lemma, under certain conditions, the optimal solution is based on a likelihood ratio testing (LRT). The major difficulty with LRT in utterance verification is how to model alternative hypothesis, where the true distribution of data is unknown and alternative hypothesis usually represents a very complex and composite event. In [7, 9], the same HMM model structure is adopted to model the alternative hypothesis; they are commonly named as anti-models. On the other hand, if we study utterance verification problem from a Bayesian viewpoint, the final solution ends up with evaluating the so-called Bayes factors[3]. As shown in [3], Bayes factors is a powerful tool to model composite hypothesis, which can be used to solve many different verification problems. 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 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. We have examined the proposed approach to detect recognition errors in Bell Labs Communicator system, where we verify correct words against mis-recognized words in decoder output. 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.

2. UV BASED ON NEIGHBORHOOD INFORMATION

2.1. Nested Neighborhoods in Model Space

Let’s look at the model space $T$ of HMM. Suppose we have $N$ different HMM’s in the recognizer, denoted as $\{\lambda_i | 1 \leq i \leq N \}$. Each $\lambda_i$ can be viewed as a point in the model space $T$. Intuitively, for every given model $\lambda_i$, we are able to enumerate a set of nested neighborhoods in $T$ which are all surrounding the model $\lambda_i$:

1. Tight neighborhood $\Lambda_1^{(i)}$: $\Lambda_1^{(i)}$ is a very small neighborhood which tightly surrounds the model $\lambda_i$. As indicated in [1], this kind of neighborhood serves as a robust representation of the original model $\lambda_i$.

2. 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 $\lambda_i$’s potential competing models, which are close to $\lambda_i$ in model space, no matter whether they are used by the recognizer or not.

3. 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 $\{\lambda_i | 1 \leq i \leq N \}$, the large neighborhood of different models $\lambda_i$ should overlap with each other. On the other hand, different model $\lambda_i$ should have its own $\Lambda_0^{(i)}$, $\Lambda_1^{(i)}$ and $\Lambda_2^{(i)}$.

The whole picture is illustrated in Figure 1.

2.2. UV based on Neighborhood Information

For a given speech segment $X$, assume an ASR system recognize it as word $W$ which is represented by an HMM model $\lambda_W$. Now we are interested in examining the reliability of this decision in order to accept or reject it. Usually we formulate it as a statistical hypothesis testing problem. We test the null hypothesis $H_0$: $X$ is truly from model $\lambda_W$ against the alternative hypothesis $H_1$: $X$ is NOT from model $\lambda_W$. 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.

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. In this work, we concentrate on detecting recognition errors in ASR. Given the decision that $X$ is recognized as model $\lambda_W$, it is reasonable to consider that $X$ probably comes from some competing model of $\lambda_W$. Based on the concept described in section 2.1, we define two nested neighborhoods in model space around the underlying model $\lambda_W$: i) tight neighborhood $\Lambda_1$: as a robust representation of original model $\lambda_W$; ii) medium neighborhood $\Lambda_2$: including all potential competing models of $\lambda_W$. Therefore, we can translate the above hypothesis testing ($H_0$ vs. $H_1$) into the following ones:

$$H_0^0: \text{The true model of } X \text{ lies in tight neighborhood } \Lambda_1$$
$$H_1^0: \text{The true model of } X \text{ lies in the region } \Lambda_2 - \Lambda_1 \quad (1)$$

where $\Lambda_2 - \Lambda_1$ denotes the region inside medium neighborhood $\Lambda_2$ but excluding tight neighborhood $\Lambda_1$. Now we formulate utterance verification as a new hypothesis testing problem where we verify $H_0^0$ against $H_1^0$ to decide the reliability of the original recognition result. Please note that here both hypotheses $H_0^0$ and $H_1^0$ are composite which make it hard to solve this verification problem under the traditional framework of likelihood ratio testing (LRT). Besides LRT, there are several other tools available to solve verification problem. In this paper, we will investigate how to use Bayes factors to solve the above hypothesis testing problem.

3. BAYES FACTORS: A BAYESIAN TOOL FOR VERIFICATION PROBLEM

Bayes factors has its solid foundation from Bayesian theory. As shown in [4], 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:

$$BF = \frac{\hat{p}(X \mid H_0)}{\hat{p}(X \mid H_1)} = \int \frac{f(X \mid \lambda_0, H_0) \cdot p(\lambda_0 \mid H_0) \, d\lambda_0}{\int f(X \mid \lambda_1, H_1) \cdot p(\lambda_1 \mid H_1) \, d\lambda_1} \quad (2)$$

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

Bayes factors offer 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[4]. Therefore, Bayes factors can be used to compare with a threshold, just like likelihood ratio in Neyman-Pearson lemma, to make a decision with regards to $H_0$. In other words, if $BF > \tau$, where $\tau$ is a pre-set critical threshold, then we accept $H_0$, otherwise reject it.

In order to use Bayes factors to solve the hypothesis testing problem, i.e., $H_0^0$ vs. $H_1^0$ in eq.(1), two important issues have to be addressed first: i) how to quantitatively define neighborhoods $\Lambda_1$ and $\Lambda_2$; ii) how to properly choose prior distribution $p(\cdot)$ of HMM model parameter for each hypothesis.

4. CASE I: $(C, \rho)$ NEIGHBORHOOD AND CONSTRAINED UNIFORM PRIORS

Assume HMM model $\lambda$ is an $N$-state continuous density HMM (CDHMM) with parameter vector $\lambda = (\pi, A, \theta)$, where $\pi$ is initial state distribution, $A$ is state transition matrix, and $\theta$ is parameter vector composed of mixture parameters $\theta = \{\omega_{ik}, m_{ik}, r_{ik}\}$ ($k = 1, 2, \cdots, K$) for state $i$. The state observation $p.d.f.$ is assumed to be a mixture of multivariate Gaussian distribution with diagonal precision matrix:

$$p(x|\theta_i) = \sum_{k=1}^{K} \omega_{ik} \int_{\mathcal{D}_{ikd}} e^{-\frac{1}{2} r_{ikd}(x_d-m_{ikd})^2} \, d\lambda$$

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

At first, following the work in [8], we define the neighborhood form for both $\Lambda_1$ and $\Lambda_2$ as:

$$\Lambda(\lambda) = \{ \lambda \mid \pi = \pi^*, A = A^*, \omega_{ik} = \omega_{ik}^*, r_{ik} = r_{ik}^*, m_{ikd} = m_{ikd}^*, 1 \leq i \leq N, 1 \leq k \leq K, 1 \leq d \leq D \}$$

where $\lambda^* = (\pi^*, A^*, \omega_{ik}^*, r_{ik}^*, m_{ikd}^*)$ denotes the original model parameter which is the central point of the neighborhood, and $C > 0$ and $\rho (0 \leq \rho \leq 1)$ are used to control the shape and size of the neighborhood. For medium neighborhood $\Lambda_2$, we choose larger values for $C$ and $\rho$. And for tight neighborhood $\Lambda_1$, we choose smaller values for $C$ and $\rho$. Secondly, given the neighborhood, we assume the prior distribution of HMM model parameter is a uniform p.d.f. constrained in the neighborhood. Based on these assumptions, the calculation of Bayes factors can be simplified as:

$$BF = \frac{\hat{p}_1(X)}{\hat{p}_2(X)} = D \cdot \frac{\int_{\Lambda_2} f(X|\lambda) \, d\lambda}{\int_{\Lambda_1} f(X|\lambda) \, d\lambda}$$

where $D = \int_{\Lambda_2 - \Lambda_1} \, d\lambda / \int_{\Lambda_1} \, d\lambda$ 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 [1] is used to compute each Bayesian predictive density $\hat{p}(X)$, e.g., $\hat{p}_1(X) = \int_{\Lambda_1} f(X|\lambda) \, d\lambda$ and $\hat{p}_2(X) = \int_{\Lambda_2 - \Lambda_1} f(X|\lambda) \, d\lambda = \int_{\Lambda_2} f(X|\lambda) \, d\lambda - \int_{\Lambda_1} f(X|\lambda) \, d\lambda$.

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

![HMM Model Space](image-url)
Bayesian predictive density \( \hat{p}(X) \) for the neighborhood \( \Lambda \) is computed as follows:

\[
\hat{p}(X) = \int_{\Lambda} f(X | \lambda) \, d\lambda = \int_{\Lambda} \sum_{s,l} f(X, s, l | \lambda) \, d\lambda
\]

\[
= \sum_{s,l} \int_{\Lambda} f(X, s, l | \lambda) \, d\lambda \approx \max_{s,l} \int_{\Lambda} f(X, s, l | \lambda) \, d\lambda
\]

(6)

where \( s \) and \( l \) denote 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 \( \{ \bar{s}, \bar{l} \} = \{ s_1, \ldots, s_T, l_1, \ldots, l_T \} \), i.e.,

\[
\{ \bar{s}, \bar{l} \} = \arg \max_{s,l} \int_{\Lambda} f(X, s, l | \lambda) \, d\lambda
\]

(7)

In order to balance contribution from different models in the neighborhood, we introduce an exponential scale factor \( \alpha (\alpha > 0) \) into the integral calculation. Given the optimal path \( \{ \bar{s}, \bar{l} \} \), which can be obtained by VBPC search algorithm in [1], the approximate Bayesian predictive density \( \hat{p}(X) \) is computed as follows:

\[
\hat{p}(X) \approx \{ \int_{\Lambda} [f(X, \bar{s}, \bar{l} | \lambda)]^\alpha \, d\lambda \}^{1/\alpha}
\]

\[
= \prod_{i=1}^{N} \prod_{k=1}^{K} \prod_{d=1}^{D} \left( \frac{C_{1d}^{\alpha_d} \sum_{j=1}^{M} (\frac{1}{2\pi})^{D/2} e^{-\frac{1}{2} ||x_{ikd} - \tau_{ikd}||^2}}{2C_{d}^{-1}\rho_d} \right)
\]

\[
\left\{ \left( \frac{2\pi}{\alpha_d} \right)^{D/2} \left[ \Phi\left( \sqrt{\alpha_d} m_{ikd} - \tau_{ikd} + C_{d}^{-1}\rho_d \right) \right] 
\]

\[\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{y}^{\infty} e^{-x^2/2} \, dx.\]

The only issue left here is to determine \((C, \rho)\) for neighborhoods. We propose the following two different methods:

- **Global setting**: We manually select \((C_1, \rho_1)\) for tight neighborhood \( \Lambda_1 \) and \((C_2, \rho_2)\) for medium neighborhood \( \Lambda_2 \), 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 HMM.

- **State-dependent setting**: Given a HMM state with parameter \( \theta_i \), we assume the neighborhood for this state \( \theta_i \) is defined as in eq.(4), we calculate the maximum deviation distance in the neighborhood from the central point as:

\[
D = C^2 \prod_{d=1}^{D} \left[ (d-1)\rho_d^2 \cdot \max_{k=1}^{K} r_{ikd} \right]
\]

In reverse, when fixing \( \rho \), we can calculate \( C \) based on a deviation distance \( D \) from the above equation. Thus, we first manually select \( \rho_1 \) and \( \rho_2 \) for tight neighborhood \( \Lambda_1 \) and medium one \( \Lambda_2 \), then we define maximally allowed deviation distances for tight and medium neighborhoods, i.e., \( D_1 \) and \( D_2 \)(usually \( D_1 < D_2 \)), finally we can calculate \( C_1 \) and \( C_2 \) for different HMM states based on \( D_1 \) and \( D_2 \). In this case, we can use different tight (or medium) neighborhoods for different states in HMM.

5. **CASE II: DELTA PRIORS**

Given a model \( \lambda^* \), we still define two neighborhoods around \( \lambda^* \): tight neighborhood \( \Lambda_1 \) and medium neighborhood \( \Lambda_2 \). Then for each neighborhood, to say \( \Lambda_1 \), 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 \( \Lambda_1 \). That is,

\[
\rho_1(\lambda) = \frac{1}{N_{\Lambda_1}} \sum_{\lambda_i \in \Lambda_1} \delta(\lambda - \lambda_i)
\]

(10)

where \( N_{\Lambda_1} \) denotes the total number of models inside \( \Lambda_1 \).

Similarly, we can build prior distribution for the region \( \Lambda_2 - \Lambda_1 \):

\[
\rho_2(\lambda) = \frac{1}{N_{\Lambda_2 - \Lambda_1}} \sum_{\lambda_i \in \Lambda_2 - \Lambda_1} \delta(\lambda - \lambda_i)
\]

(11)

Based on these two priors, Bayes factors to verify hypotheses \( H_0 \) and \( H_1 \) in eq.(1) can be simplified as:

\[
BF = \frac{\sum_{\lambda_i \in \Lambda_1} f(X | \lambda_i) / N_{\Lambda_1}}{\sum_{\lambda_i \in \Lambda_2 - \Lambda_1} f(X | \lambda_i) / N_{\Lambda_2 - \Lambda_1}}
\]

(12)

In many cases, we usually use state-tied HMMs for speech recognition. Thus, we case build the above delta priors separately for each distinct state. Then Bayes factors is calculated for each state segment independently and combine all of them to get verification score for the whole model for utterance verification.

6. **EXPERIMENTS**

To examine the viability of the proposed methods, we evaluate on Bell Labs Communicator system[6] to detect recognition errors in final recognition results from the decoder. The Bayes-factors-based verification method is compared with likelihood ratio testing (LRT), which is based on standard anti-models.

6.1. **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 HMM models are trained by using the standard Baum-Welch ML estimation on total 46 hours of speech data. The acoustic models are state-tied, tri-phone HMM models, which include 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 2,600 words is used in the system. The baseline system achieves 15.8% word error rate (WER) in our independent evaluation set, which includes 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). As our baseline verification system, we use LRT with normal mono-phone anti-models, which are trained from all training data with fixed force-alignment phone segmentation. 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 anti-model for this phone. As shown in first row of Table 1, we achieve 40.0% equal error rate (EER) with this method.
Then we sort all other states according to these distance measures.

Table 1: Verification performance comparison (Equal Error Rate in %) of baseline UV method (LRT-anti-models) with the proposed new approaches in several different settings. The corresponding parameter setting for each case is also given. Here we always fix $\alpha = 1.2$ for case I.

<table>
<thead>
<tr>
<th>Method</th>
<th>EER</th>
<th>Parameter setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>anti-model</td>
<td>40.0</td>
<td></td>
</tr>
<tr>
<td>caseI-global</td>
<td>36.7</td>
<td>$\rho_1 = 0.1, \rho_2 = 0.7, C_1 = C_2 = 3.0$</td>
</tr>
<tr>
<td>caseI-state</td>
<td>32.4</td>
<td>$\rho_1 = 0.1, \rho_2 = 0.9, D_1 = 4, D_2 = 200$</td>
</tr>
<tr>
<td>caseII</td>
<td>31.5</td>
<td>$N_1 = 1, N_m = 100$</td>
</tr>
</tbody>
</table>

Table 2: Verification performance comparison (Equal Error Rate in %) of the proposed new methods, when we properly choose the parameters, the new methods, both with global setting and state-dependent setting, can achieve better performance than anti-model. We get 36.7% EER with global setting and 32.4% with state-dependent setting. One possible reason why state-dependent setting gives much better performance is that we can use different neighborhood sizes for different HMM states.

6.2. Bayesian Approach With Settings in Case I

Here we repeat the same UV experiments with Bayes factors methods shown in section 4. We choose $(C, \rho)$ neighborhood and constrained uniform prior distribution. As shown in 2nd and 3rd rows of Table 1, when we properly choose the parameters, the new methods, both with global setting and state-dependent setting, can achieve better performance than anti-model. We get 36.7% EER with global setting and 32.4% with state-dependent setting. One possible reason why state-dependent setting gives much better performance is that we can use different neighborhood sizes for different HMM states.

6.3. Bayesian Approach With Settings in Case II

In this part, we choose delta priors in eqs.(10) and (11) for Bayes factors. At first, for each distinct state in acoustic models, we calculate its distance measures from all other states (roughly 4K). Then we sort all other states according to these distance measures. In this case, we choose neighborhood sizes to include exactly $N_t$ other states in $\Lambda_1$ and $N_m$ in $\Lambda_2 - \Lambda_1$. In the last row of Table 1, we show verification performance of the same experiments, where we have achieved 31.5% in EER for Case II. Generally speaking, delta prior give slightly better performance than uniform distribution based on $(C, \rho)$. But it usually requires a very large number of delta components, e.g., 1000, which is much more computationally expensive than Case I.

Finally, we also plot the ROC curves for different methods in Figure 2, which also clearly show that the new approaches significantly outperform the standard method.

7. 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 different neighborhood definitions: i) $(C, \rho)$ neighborhood with constrained uniform prior p.d.f.; ii) mixture delta prior p.d.f. with a fixed number of mixand included in the neighborhood. Based on these two definitions, Bayes-factors-based confidence measures are easy to calculate for HMM model. Experimental results in Bell Labs communicator system show that comparing with the conventional approach of likelihood ratio testing and anti-models the new method has dramatically improved verification performance when verifying correct words against mis-recognized words in the recognizer’s output. As a future work, instead of Bayes factors, other statistical hypothesis testing tools can also be used to implement the neighborhood based idea for UV.

8. REFERENCES


