Unsupervised re-scoring of observation probability in Viterbi based on reinforcement learning by using confidence measure and HMM neighborhood

Carlos Molina, Nestor Becerra Yoma, Fernando Huenupan, Claudio Garreton
Department of Electrical Engineering, Universidad de Chile, Santiago, Chile

Abstract

This paper proposes a new paradigm to compensate for mismatch condition in speech recognition. A two-step Viterbi decoding based on reinforcement learning is described. The idea is to strengthen or weaken HMM’s by using Bayes-based confidence measure (BBCM) and distances between models. If HMM’s in the N-best list show low BBCM, the second Viterbi decoding will prioritize the search on neighboring models according to their distances to the N-best HMM’s. As shown here, a reduction of 6% in WER is achieved in a task which results difficult for standard MAP and MLLR adaptation.

Index Terms: speech recognition, reinforcement learning, confidence metrics and telephone speech

1. Introduction

The mismatch between the training and testing database has widely been studied in the field of automatic speech recognition (ASR) due to its relevance in practical applications of the technology. This problem has been addressed with model adaptation/compensation techniques or with methods for noise removal from the corrupted signals. Conventional adaptation/compensation techniques (e.g. ML, MAP and MLLR), although effective in many circumstances, dramatically degrade when just a few adapting utterances are available [1][2]. This problem has been addressed by grouping the HMM’s on a finer acoustic subspaces level [2][3]. However, another possibility to reduce the number of parameters to be adapted would be to directly re-estimate the observation probabilities instead of their parameters (mean vectors and covariance matrices) [4].

The effectiveness of unsupervised adaptation is also significantly degraded with respect to supervised adaptation [2][5]. This result should also be due to the fact that the number of parameters to be re-estimated is dramatically increased because there is not a 100% certainty about the models to adapt. Unsupervised adaptation performance should be improved by mean of the use of confidence measures. The applicability of confidence measure in model adaptation has already been addressed in the literature [6][7]. In model adaptation, confidence measure is usually employed just as a heuristic function (e.g. for frame selection) because it has not been defined as a probability [6][7][8]. In [9] a Bayes Based Confidence Measure (BBCM) is proposed to address some of the limitation of ordinary confidence metrics. For instance, BBCM is a probability itself and incorporate information of speech recognition performance.

This paper addresses the problem of unsupervised adaptation with limited data. Instead of re-estimating the parameters of observation probabilities, this paper proposes to reduce the number of parameters to be adapted by re-estimating the observation probability itself. The method makes use of BBCM and the distance of neighboring HMM’s to N-best models.

The contribution of this paper concerns: a) a confidence based re-estimation of observation probability to address the problem of unsupervised adaptation with limited data; b) applicability of reinforcement learning on model adaptation/compensation; and, c) a model of the relationship between recognized and non recognized HMM’s. The method described here requires only one adapting utterance and can lead to reductions in WER equal to 6%. Compared with conventional approaches the results reported here are promising.

2. Correction component based on reinforcement learning

The motivation is to apply the reinforcement learning principle to correct or re-estimate observation probabilities [4]. Basically, the idea is to strengthen an action that produces a satisfactory state of the system, otherwise this action should be weakened [10]. Confidence measure is certainly a mean to asses the stability or reliability of the output delivered by recognizers. Accordingly, if the output presents a low confidence in the first search, the method should weaken the recognized HMM’s and prioritize other models in the second search. In contrast, if the delivered output presents a high confidence in the first decoding step, the method should strengthen the recognized HMM’s and penalize neighboring models in the second search. The method is shown in Fig. 1.

As was mentioned above, reinforcement learning could be applied as a correction in the observation probabilities. This correction factor can be interpreted from the classic adaptation/compensation techniques point of view. In fact, the result of HMM adaptation or compensation is modeled as an additive component of logarithm observation probabilities:

$$
\log[\Pr(\hat{O}(t)/\hat{\lambda}_s)] = \log[\Pr(\hat{O}(t)/\hat{\lambda}_s)] + \Delta(O,\hat{\lambda}_s,\text{compensation technique})
$$

(1)

where $\hat{\lambda}_s$ and $\hat{\lambda}_s'$ correspond to state $s$ in HMM $i$ before and after adaptation, respectively; $O(t)$ denotes frame $t$ in the observation sequence $O = [O(1),...,O(i),...,O(T)]$; and, $\Delta(\cdot)$ is the additive correction component in the log domain that is a function of $O$, HMM $\hat{\lambda}_s$ and the adaptation technique. In a similar way, the effect of a noise cancellation technique could also be expressed as:

$$
\log[\Pr(\hat{O}(t)/\hat{\lambda}_s)] = \log[\Pr(\hat{O}(t)/\hat{\lambda}_s)] + \Delta(O,\text{noise},\hat{\lambda}_s,\text{cancellation technique})
$$

(2)
where $\hat{O}(t)$ is the frame $t$ that results from the cancellation method; and, $\Delta(\cdot)$ is a function of $O$, noise, HMM $\lambda_n$, and the noise removal technique. As a consequence, the problems of HMM adaptation/compensation and noise removal could be interpreted as the adequate estimation of $\Delta(\cdot)$.

In this paper, a two-step procedure is explained to estimate the additive correction component of the observation probabilities in the log domain, $\Delta(\cdot)$. First, Viterbi based N-best analysis delivers a set of hypotheses, recognized words, HMM’s and confidence measures. Finally, a second Viterbi decoding provides the sequence of recognized words by making $\Delta(\cdot)$ be a function of confidence measures and proximity depending, respectively, on whether the HMM appears or not in the first-step N-best analysis. This second Viterbi allows re-estimating the recognized words by using the principle of reinforcement learning: if the recognized models show a high confidence, they will be strengthened in a second Viterbi; otherwise, they will be weakened.

If $Pr^{[1]}[O(t)|\lambda,S]$ and $Pr^{[2]}[O(t)|\lambda,S]$ are the observation probabilities of frame $O(t)$ given state $S$ in HMM $\lambda$ in the first and second Viterbi decoding step, respectively, then:

$$\log \left\{ Pr^{[2]}[O(t)|\lambda,S] \right\} = \log \left\{ Pr^{[1]}[O(t)|\lambda,S] \right\} + \Delta(\lambda,S,t) \quad (3)$$

### Figure 1: Diagram block of the method described here.

### 2.1. Penalizing HMM’s in N-best

Consider that $h_n$ is the $n^{th}$ hypothesis in the N-best Viterbi list that is composed of $N$ hypotheses. Every hypothesis corresponds to an alignment that allocates frames to a state in a given HMM. The list of words that appear in the N-best hypotheses is denoted by $W_{\gamma,best} = [w_{\gamma,best}(1),...,w_{\gamma,best}(T)]$, which in turn defines the set of HMM’s $\Lambda_{\gamma,best} = \{\lambda_{\gamma,n,best}(t,n)\}$ that are contained in the N-best list, where $1 \leq t \leq T$ and $1 \leq n \leq N$. The pair $(t,n)$ defines the frame $t$ and the hypothesis $n$. Every word in $W_{\gamma,best}$ can be associated to a confidence measure. The idea is to penalize N-best HMM’s according to their confidence measure: if the recognition output is not reliable, HMM’s in the neighborhood should have more priority in the second Viterbi decoding. In this paper Bayes Based Confidence Measure, BBCM, is employed. BBCM is defined as [9]:

$$BBCM(WF) = P(w_{\gamma,best}(i) \text{ is OK } \mid WF) = P(w_{\gamma,best}(i) \text{ is OK }) P(w_{\gamma,best}(i) \text{ is OK }) P(WF) \quad (4)$$

where “OK”, that substitutes “correct” in [9], denotes the fact that a word was properly recognized; $WF$, denotes the word feature associated to word $w_{\gamma,best}(i)$.

In this paper two word features are used to compose the BBCM metric. One of them is known as Word Density Confidence Measure, WDCM [9][11]. In order to improve the accuracy of the confidence metric, maximum hypothesis log-likelihood within the N-best list, $ML_n$, is also included as a word feature. $ML_i$ is defined as:

$$ML_i = \log \left[ \max_{r} \left\{ Q(h_i) \right\} \mid E(w_i,H) \right] \quad (5)$$

where $Q(h_i) = P(h_i)P(O|h_i)$; $Q(h_i)$ is the likelihood score of hypothesis $h_i$ given by the Viterbi search; $P(h_i)$ and $P(O|h_i)$ are the language model and observation probability of $h_i$ respectively. $\gamma$ is the language model scaling factor; $h_i$ is a hypothesis in the N-best Viterbi list; $E(w_i,H)$ defines the hypotheses where $w_i$ is contained; and finally, $H$ denotes all hypotheses obtained from the Viterbi decoding. Then, the probabilities $BBCM(ML_i)$ and $BBCM(WDCM)$ are estimated according to (4). As mentioned in [9], $BBCM(ML_i)$ and $BBCM(WDCM)$ could be approximated with:

$$BBCM(WDCM,ML_i) = BBCM(WDCM) \cdot BBCM(ML_i) \quad (6)$$

Observe that, in contrast to ordinary confidence measures found in the specialized literature [8], BBCM is a probability. The a priori distributions in (4) are estimated as in [9] with a development database different from the testing and training data. By employing the principle of reinforcement learning [10] and using the fact that BBCM is a probability, $\Delta(\cdot)$ in (3) can be expressed as [4]:

$$\Delta(O,t,\lambda_{\gamma,n,best}(t,n),h_n,W_{\gamma,best}) = \alpha \cdot \log \left\{ Pr(\lambda_{\gamma,n,best}(t,n) \text{ is OK } \mid O,W_{\gamma,best}) \right\} \quad (7)$$

where $Pr(\lambda_{\gamma,n,best}(t,n) \text{ is OK } \mid O,W_{\gamma,best})$ is equal to $BBCM(ML_i,WDCM)$; and, $\alpha$ is a scaling factor.
2.2. On including neighboring HMM’s in the second viterbi search

Neighboring HMM’s are defined at a given instant \( t \) and hypothesis \( h_n \) and are denoted by \( \Lambda \text{Neighboring} = \{ \lambda_{\text{Neighboring}}(t,n,p) \} \), where \( \lambda_{\text{Neighboring}}(t,n,p) = \{ \text{HMM } \lambda_i \neq \lambda_{\text{Neighboring}}(t,n) \} \) and \( 1 \leq p \leq N_{\text{HMM}} - 1 \).

Notice that the total number of models in the task. Figure 2 shows the relationship between the model allocated by Viterbi, \( \lambda_{\text{Viterbi}}(n) \), and the neighboring models, \( \lambda_{\text{Neighboring}}(t,n,p) \). By employing the principle of reinforcement learning [10], \( \Delta(\cdot) \) in (3) can be expressed as [4]:

\[
\Delta(O,t,\lambda_{\text{Neighboring}}(t,n,p),h,W) = 
\alpha \cdot \log \left[ \frac{\Pr(\lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid \lambda_{\text{Viterbi}}(t,n))}{\Pr(\lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid h,W)} \right]
\]

(8)

where (8) takes into consideration the fact that the probability of finding a correct model in the neighborhood of \( \lambda_{\text{Neighboring}}(t,n) \), given that \( \lambda_{\text{Viterbi}}(t,n) \) is not OK, depends on the distance between \( \lambda_{\text{Neighboring}}(t,n) \) and \( \lambda_{\text{Viterbi}}(t,n,p), d(\lambda_{\text{Viterbi}}(t,n,p), \lambda_{\text{Neighboring}}(t,n)) \); and,

\[
\Pr[\lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid \lambda_{\text{Viterbi}}(t,n)) = 
\frac{\Pr[\lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid \lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid h,W)}{\Pr[\lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid h,W)}
\]

(9)

where \( 1 \leq p \leq N_{\text{HMM}} - 1 \), and \( f[\cdot] \) models the probability distribution of \( \lambda_{\text{Neighboring}}(t,n,p) \) being the correct model given that \( \lambda_{\text{Viterbi}}(t,n) \) is not OK and \( g[\cdot] \) the neighboring model. From (7), \( \Pr[\lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid \lambda_{\text{Viterbi}}(t,n)] \) in (9) can be written as,

\[
\Pr[\lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid \lambda_{\text{Viterbi}}(t,n)] = 
1 - \Pr[\lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid \lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid h,W] = 1 - \text{BBCCM}(ML, WDCM)
\]

(10)

where index \( i \) denotes a word in the N-best list.

The distance \( d(\lambda_{\text{Neighboring}}(t,n,p), \lambda_{\text{Neighboring}}(t,n)) \) is defined in the context of the Kullback–Leibler (K-L) metrics and the distribution \( f[\cdot] \) in (9) is estimated by assuming that it does not depend on \( t \) nor on \( n \). Considering \( f[\cdot] \) as independent of \( t \) and \( n \) is an approximation that is necessary to reliably estimate \( f[\cdot] \) with a manageable amount of database. Then, by simplifying the notation and employing the Bayes theorem, this distribution can be written as:

\[
f[\lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid \lambda_{\text{Neighboring}}(t,n,p) = \text{OK}, \lambda_{\text{Viterbi}}(t,n,p)] = 
\frac{g[d(\lambda_{\text{Viterbi}}(t,n), \lambda_{\text{Viterbi}}(t,n,p) = \text{OK}, \lambda_{\text{Viterbi}}(t,n,p) = \text{OK})]}{g[d(\lambda_{\text{Viterbi}}(t,n), \lambda_{\text{Viterbi}}(t,n,p) = \text{OK}, \lambda_{\text{Viterbi}}(t,n,p) = \text{OK})]}
\]

(11)

where distributions \( g[\cdot] \) and \( h[\cdot] \) are estimated with the development database by employing the following procedure: first, each utterance is recognized and the first hypothesis given by the Viterbi decoding is stored; second, both the reference and the recognized utterance transcriptions are used to deliver forced Viterbi alignments; then, the allocated reference and recognized HMM’s are compared frame-by-frame; if reference and recognized HMM’s are different, the K-L distance \( d(\cdot) \) between those models is used to update histogram \( g[\cdot] \) and the K-L distances between the recognized model and all the other HMM’s are employed to update histogram \( h[\cdot] \). The distributions \( g[\cdot] \) and \( h[\cdot] \) are shown in Fig. 3. The distribution \( \Pr(\lambda_{\text{Viterbi}}(t,n) = \text{OK} \mid \lambda_{\text{Viterbi}}(t,n) = \text{OK}) \) is supposed uniformly distributed and equal to \( \frac{1}{N_{\text{HMM}} - 1} \).

As mentioned above, given two HMM’s, \( \lambda_1 \) and \( \lambda_2 \), the distance between these HMM’s, \( d(\lambda_1, \lambda_2) \), employed in p.d.f. \( f[\cdot] \), is defined in the context of the (K-L) metrics. In this paper, each HMM corresponds to a triphone that is modeled with a three-state left-to-right topology without skip-state transition. The K-L metrics between HMM’s is defined as the average distance between the state that compose the HMM’s [12][13]. It is worth emphasizing that the K-L distance could have been estimated using any other method [12][13]. However, the estimation of the K-L distance is not the main focus of this paper and the procedure adopted was employed as an approximation to save computational complexity.

![Figure 2: Definition of \( \lambda_{\text{Neighboring}}(t,n) \) and \( \lambda_{\text{Neighboring}}(t,n,p) \) given a Viterbi alignment.](image)

### 3. Experiments

The approach explained in this paper was tested with a Spanish database recorded on the telephone line. Users phoned to a ASR-based cinema enquiry system implemented with Galaxy II [14] at the Speech Processing and Transmission Lab., Universidad de Chile. The dialogue was the following: first, the system asked the user to choose one film from a list composed of 80 films; second, the system prompted for the name and neighborhood of the cinema; finally, the user had to say if he/she wanted to go to the cinema in the morning, afternoon or evening. The ASR employed a language model based on trigrams and allowed the user to employ natural language to input the information required by the system. The vocabulary is composed of 221 words. The training database corresponded to 13.897 utterances. All of the training signals were employed to train the CDHMMs. The a priori p.d.f.’s in (11), \( g[\cdot] \) and \( h[\cdot] \), shown in Fig. 3, are estimated with a development database composed of 1.036 utterances (1.437 words). The a priori distributions for BBCM in (4) were estimated with the same development database. The testing database corresponds to 793 utterances (1.127 words). N-best analysis was based on the ten best hypotheses (\( N = 10 \)) obtained from Viterbi alignment. The recognized sentence corresponded to the first hypothesis within the N-best list and the baseline system gave a WER equal to 13.83%. Thirty-three MFCC parameters per frame.
were computed. Cepstral Mean Normalization (CMN) was also employed. Standard MAP and MLLR vector mean adaptation were also implemented according to [15] and [16], respectively, to assess the difficulty of the task addressed here. In MAP the optimal learning constant was chosen. The results are shown in Tables 1-2.

4. Discussion and conclusions

As can be seen in Table 1, the propose method can lead to reductions in WER equal to 6%. This improvement takes place when $\alpha$ is equal to 0.1. Significance analysis with the McNamars testing [17] shows that this result is significant (p < 0.01). Observe that $g[\cdot]$, $h[\cdot]$, and $f[\cdot]$ are estimated with the development database.

Results with MAP and MLLR vector mean adaptation are presented in Table 2. As can be seen, MAP and MLLR vector mean adaptation do not present improvement with the task addressed here: unsupervised compensation with telephone speech by using only one short adaptation utterance. Instead of re-estimating the parameters of HMM’s or modifying the observation features, this paper proposes a new paradigm to reduce the number of parameters to be adapted by re-estimating or correcting the observation probability itself.

The results presented here are promising in conditions where standard MAP and MLLR vector mean adaptation have lead to not improvement. Finally, to optimize the computational load by tying the correction of observation probability across a number of models, to improve the accuracy of distance metrics between HMM’s, and the applicability of the RL model discussed in this paper to other problems in pattern recognition.

5. Acknowledgements

This work was funded by Conicyt-Chile under grants Fondef D05I-10243/D05I-10243 and Fondecyt 1070382.

6. References


<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>WER (%)</th>
<th>Reduction in WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>13.83</td>
<td>0.0</td>
</tr>
<tr>
<td>0.01</td>
<td>13.83</td>
<td>0.0</td>
</tr>
<tr>
<td>0.05</td>
<td>13.58</td>
<td>1.8</td>
</tr>
<tr>
<td>0.1</td>
<td>12.95</td>
<td>6.4</td>
</tr>
<tr>
<td>0.5</td>
<td>13.93</td>
<td>-0.7</td>
</tr>
</tbody>
</table>

Table 1: WER vs $\alpha$ defined in (7) and (8) for testing database.

<table>
<thead>
<tr>
<th>Method described</th>
<th>WER (%)</th>
<th>Reduction in WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLLR</td>
<td>14.55</td>
<td>-5.2</td>
</tr>
<tr>
<td>MAP</td>
<td>13.83</td>
<td>0.0</td>
</tr>
<tr>
<td>Method described here</td>
<td>12.95</td>
<td>6.4</td>
</tr>
</tbody>
</table>

Table 2: Comparison with standard MLLR and MAP vectors mean adaptation.

Figure 3: Estimated histograms corresponding to distributions $g[\cdot]$ and $h[\cdot]$ in (11).