An Efficient, Fast Matching Approach Using Posterior Probability Estimates in Speech Recognition

Sherif Abdou, Michael S. Scordilis

Department of Electrical and Computer Engineering, University of Miami
Coral Gables, Florida 33124, U.S.A.

Abstract

Acoustic fast matching is an effective technique to accelerate the search process in large vocabulary continuous speech recognition. This paper introduces a novel fast matching method. This method is based on the evaluation of future posterior probabilities for a look-ahead number of timeframes in order to exclude unlikely phone models as early as possible during the search. In contrast to the likelihood scores used by more traditional fast matching methods these posterior probabilities are more discriminative by nature as they sum up to unity over all the possible models. By applying the proposed method we managed to reduce by 66% the decoding time consumed in our time-synchronous Viterbi decoder for a recognition task based on the Wall Street Journal database with virtually no additional decoding errors.

1. Introduction

Decoding the best word sequence in large vocabulary continuous speech recognition is a computationally demanding search process. For performing this search in reasonable time a rapid method of reducing the search space must be invoked. One of the effective techniques used for this purpose is *fast match.* The basic idea of this technique is to look-ahead in time to identify, using a computationally cheap method, some future search extensions with poor acoustic score that can be discarded before applying the expensive detailed match evaluation. Fast match techniques are characterized by the approximations that are made in the models in order to reduce the computation. The essential features required of the fast match are that it be accurate and that it requires a small amount of computation. There is an obvious trade-off between these two objectives.

In [1] a phoneme level fast match was proposed. In that approach each time a hypothesis is formed about a new phoneme arc to be started, it is first checked whether it can survive the pruning steps that would be performed for the next future time frames. For this purpose an approximate probability estimate is computed using a context independent (CI) version of the acoustic models. Although using CI models for computing the fast match score is less expensive, and also less precise, this approach is still computationally expensive. Moreover, the computations made for the fast match evaluation cannot be reused in detailed matching. It has also been shown that the discrimination between the likelihood scores is often very week for short term estimation [2].

Another fast match approach was introduced for hybrid recognition systems under the name *Phone Deactivation Pruning*. While standard HMM systems are based on the estimation of probabilistic distribution functions for the observations' likelihood, hybrid systems [3] utilize neural networks (NN) to estimate the HMM states' probability given an acoustic observation as input. These posterior probabilities are discriminative by nature as they sum up to unity over all HMM states. Phone Deactivation Pruning simply prunes phones whose posterior probability falls under a specific threshold. This technique was reported as one of the major advantages of the hybrid systems.

In this work, we introduce an efficient approach for estimating posterior probabilities for the conventional HMM-based decoders. Then we introduce a fast matching technique that utilizes these posterior probabilities to speed up the decoding time for a time-synchronous Viterbi decoder. The proposed technique is similar to the phone deactivation pruning of hybrid systems but it includes a novel posterior look-ahead pruning (PLP) technique that excludes the unlikely HMM states as early as possible during the search. The organization of this paper is as follows. Section 2 includes description for the methods that we developed to estimate posterior probability for HMM-based speech recognition systems. In Section 3 the proposed PLP technique is introduced. Experimental results for the application of this fast matching technique in our Viterbi-decoder using a large-scale corprus are included in Section 4. Section 5 includes the conclusion and future work directions.

2. Estimating Posterior Probabilities For HMM Based Systems

Standard HMM systems are based on the estimation of the observations' likelihood functions $p(x|q)$ for each HMM state q, with x denoting an arbitrary feature vector extracted from the acoustic observation. Bayes' rule can be applied to derive local posterior state probabilities from the state likelihoods according to

$$p(q/x) = \frac{p(x/q)p(q)}{p(x)} \quad (1)$$

While HMM systems provide a model for the state likelihood probabilities $p(x/q)$, they do not have a dedicated model for the state priors $p(q)$ or the observations' probabilities $p(x)$.

2.1 State Prior Estimate

Quantity $p(q)$ can be estimated from speech training data by running a Viterbi search to generate the best alignment of these data with their reference transcriptions up to the state level. From this alignment, a state prior probability $p(q)$ can be estimated by the fraction of time this state was active in the total data set duration. The accuracy of this estimate depends on the size of speech data used.

2.2. Observations Probability Estimation

Quantity $p(x)$ can be estimated as the sum over all the HMM states' likelihood functions weighted by the states' priors according to

$$p(x) = \sum_q p(q)\, p(x/q) \ . \qquad (2)$$

This computation requires the evaluation of the density functions of all states, which is computationally very expensive. In HMM systems the states' likelihood functions $p(x/q)$ are modeled as weighted sums of basic mixture distribution functions, usually Gaussian functions, and defined as

$$p(x/q) = \sum_{i=1}^{C_q} w_{qi} g_{qi}(x) \ , \qquad (3)$$

where w_{qi} and g_{qi} are the weights and mixture components for state q respectively and C_q is the number of mixture components for state q. For systems with large number of tied states, the estimation of $p(x)$ can be simplified according to

$$\begin{aligned} p(x) &= \sum_q p(q)p(x/q) = \sum_q p(q) \sum_{i=1}^{C_q} w_{qi}\, g_{qi}(x) \\ &= \sum_{j=1}^{C} \left(\sum_{all\ q\ tied\ to\ g_j} p(q)\, w_{qj} \right) g_j(x) = \sum_{j=1}^{C} W_j\, g_j(x) \end{aligned} \ , \qquad (4)$$

where C denotes the total number of Gaussian mixture components used for all the tied states [4]. Hence, $p(x)$ can be regarded as a weighted sum of the mixtures density functions. As the number of mixture components increases and the less tied the whole system gets, the more expensive the computation of $p(x)$ becomes. However with careful exploration of the acoustic space a large amount of the computation effort can be avoided. For this purpose we introduce a new model, which we name *Catch-All* model. This model can reduce the number of Gaussian mixtures that need to be evaluated by a large factor without serious reduction in the accuracy of the estimated observation likelihood.

2.3. The Catch-All Model

The design target of this model is to find those, and preferably only those, Gaussian mixtures that have a non-negligible effect on the total observation likelihood, and this should be done in a time that is negligible in comparison to the time required to evaluate the whole set of mixtures. The initial version of this model is created by pooling the mixture Gaussian components from the entire set of states' models. This initial model is then optimized using a two steps procedure. The first step is a clustering technique that reduces the local redundancy of the mixtures. The second step uses the resulting model to build a data structure that can be used for fast selection of the significant mixtures that cover a large fraction of the observation likelihood. The following sections describe the details of this procedure.

2.3.1 Clustering technique

The idea of this technique involves an iterative bottom-up clustering process for finding the most similar pair of Gaussians and then combining them. During each iteration, the two Gaussians that are most similar to each other are found and combined into a new Gaussian. For the similarity measure we used a weighted *Bhattacharyya* distance [5]. This distance measure has been used in several speech-related tasks [6], leading to good results. The specific implementation of the *Bhattacharyya* distance metric for the Gaussian distributions is

$$B_{distance} = \frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \ln \frac{\left| (\Sigma_1 + \Sigma_2)/2 \right|}{\left| \Sigma_1 \right|^{1/2} \left| \Sigma_2 \right|^{1/2}} \qquad (5)$$

Here, μ_1 and μ_2 are the means of the Gaussians and Σ_1 and Σ_2 are their variances. This distance behaves as a measure of the overlap between two Gaussians. After calculating the $B_{distance}$ between every pair of Gaussians, the pair with the lowest $B_{distance}$ is combined to form a new Gaussian. The parameters for the new Gaussian are derived from the parameters of the Gaussians from which it is born. The weight of the new Gaussian is equal to the sum of the weights of its Gaussian parents, the new mean and variance is a weighted sum of the means and variances of the parent Gaussians normalized by the sum of the weights of the parent Gaussians. After the new Gaussian is defined, it is added to the set of Gaussians describing the *Catch-All* model. The parents from which it was created are removed, resulting in reducing the number of Gaussians by one. The process is repeated as long as $B_{distance}$ is less than a clustering threshold TH_C. The model resulting from this clustering procedure is used to build the VQ codebook and the neighborhood information as described in the next section.

2.3.2 Vector Quantization (VQ)

VQ is used extensively in many ASR systems as a means of reducing the computational cost of the acoustic model evaluation [7]. Here VQ is used in a similar manner. The motivation behind this technique is as follows. If an input vector is an outlier with respect to a component distribution, that is, it lies on the tail of the distribution, then the likelihood of that component producing the input vector is very small. This results in the components of a likelihood mixture having a large dynamic range, with only few of them tending to dominate the likelihood for a particular input vector. Hence the observation likelihood could be computed solely from those components without a noticeable loss in accuracy. The VQ technique attempts to efficiently select these components, or a subset containing them at each frame. A VQ codebook is generated by clustering all the mixtures in the model set in an unsupervised manner [8]. Then, for each code-vector in the codebook, a subset of the mixtures from the model set is identified as containing only significant contributors. This procedure is performed offline and the resulting *neighborhood information* is stored as part of the acoustic model. During model evaluation, each observation vector is quantized using the designed VQ codebook and the likelihoods are calculated only for those mixtures contained in the corresponding neighborhood. The compression ratio of the *Catch-All* model can be controlled by the value of the clustering threshold, TH_C, and the VQ codebook size, N_{VQ}. This constitutes a trade off between accuracy and efficiency. Increasing the merging threshold or decreasing the neighborhood cell size will improve efficiency because fewer Gaussian mixtures will be computed during model evaluation. However, this will be done on the price of the degradation in the model accuracy. The goal is to compress the acoustic space evenly so that the entire space is covered with reasonable

resolution. Table 1 shows the design parameters, N_{VQ} and TH_C, for six prototype *Catch-All* models with compression ratios ranging between 75% and 98%. The accuracy of each of these models can be judged by the average change in the estimated observation likelihood, $p(x)$, compared to a baseline model that use the whole set of Gaussians.

Model	N_{VQ}	Th_C	Mixtures	Compression	Δ likelihood
S0	--	0	79200	Baseline	0
S1	128	1.3	19300	75.0%	-0.048
S2	128	1.6	10296	87.0%	-0.083
S3	256	1.3	12434	84.3%	-0.076
S4	256	1.6	3880	95.1%	-0.097
S5	512	1.3	6098	92.3%	-0.091
S6	512	1.6	1900	97.6%	-.0.183

Table 1. The *CATCH-ALL* model design parameters

The displayed results in Table 1 illustrates the average loss in the likelihood score due to the effect of calculating only subsets of the whole Gaussians set. An important question is how much loss in likelihood score would be acceptable. Results in Section 4 give an estimate for this amount.

3. Posterior Look Ahead Pruning

In the previous section we showed that it is possible to get estimates for posterior probabilities in HMM based systems with efficient computation. These estimates can then be used to develop a fast matching technique similar to the way they were exploited in the Hybrid systems. In the proposed technique a global threshold *TH* is used to prune those states q for an observation vector x whose posterior probability falls below this threshold according to the rule

$$If \;\; p(q/x) < TH \;\; \Rightarrow \;\; prune\ state\ \ q \qquad (6)$$

However, the direct implementation of this rule in HMM based systems may not lead to an effective pruning. To illustrate this concept, Figure 1 shows the posterior probability estimates of the four HMMs phones /th/, /ae/, /t/ and /s/ in the utterance of the word *"That's"* extracted from an utterance of our speech corprus.

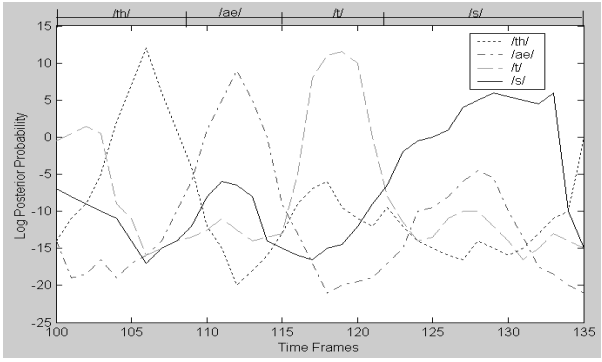

Figure 1. Normalized posterior probability estimates and alignment of the phones in an utterance "That's"

It is obvious that the posterior probability estimates reach their maximum values at the phone centers while close to the phonetical borders the discrimination among the posterior probability estimates is very week. This effect is caused by the physical constraints of the speech articulators where inertia prevents instantaneous transitions between states of the voice production system. For a three state HMM model, such as the one used for this work, the middle state represents some idealized articulator configuration in contrast to the left and right states which represent the articulation effect of the left and right contexts respectively. Thus, it may be concluded that state deactivation pruning cannot prevent an HMM node from being expanded at all, but only comes into effect after several frames have passed, during which the posterior probability has not drop below the threshold.

Based on the previous finding we propose a novel Look-ahead Pruning Technique (PLP). This technique works as follows. At any possible start of a new HMM h, the decoder looks ahead in time for an HMM-specific number of frames n, in order to check the posterior probability for the frame that is most likely to be close to the specific HMM's center. Once the HMM's posterior probability at this point turns out to fall below a certain look-ahead threshold TH_L, the whole HMM is pruned. This means that HMM h must not start at the actual frame. In our recognition system, with three state HMMs, the HMM posterior probability is estimated by the second state. The start of HMM h is omitted if the HMM's posterior probability at time $t + n$ falls below the look-ahead threshold TH_L according to this rule:

$$If \;\; p(q_2/x_{t+n}) < TH_L \Rightarrow \; forbid \;\; start\ of \;\; HMM\ h\ at\ time\ t \qquad (7)$$

In this rule q_2 represents the second state in the HMM h and n represents the average duration of h's first state plus half the average duration of the second state. Thus, it is the average duration to the center of the second state in the standard linear three-state HMM system. A specific value of n can be estimated for each HMM from the acoustic model training data. It was also found that it is more effective to have HMM-specific thresholds rather than a single global threshold for all the models. These thresholds can be easily computed as the minimum posterior probabilities calculated for each model on the training data. Another advantage of these individual thresholds is that the state priors probability can be omitted in the calculation according to this modified rule:

$$If \;\; p(q_2/x_{t+n}) = \frac{p(x/q_2)p(q_2)}{p(x)} < TH_{Lq} \Rightarrow \;\; \frac{p(x/q_2)}{p(x)} < \frac{TH_{Lq}}{p(q_2)} \Rightarrow \qquad (8)$$

$$\frac{p(x/q_2)}{p(x)} < TH_{Lq} \Rightarrow \; forbid \;\; start\ of \;\; HMM\ h\ at\ time\ t$$

4. Experiments and Results

The proposed PLP technique was tested using the Wall Street Journal (WSJ) 5K word recognition task. The acoustic models used consisted of 4K cross word triphones and 1.4K left diphones. A three state decision tree clustered HMMs were trained using the WSJ-284 data set. Each state pdf used up to 16 Gaussian mixtures with individual variances for each Gaussian for a total of 79.2K Gaussians. The WSJ Nov 92 test-set of 330 sentences were used to evaluate the performance of the proposed PLP technique. Two evaluation measures were used; the test set Word Error Rate (WER) and a Time Factor (TF) that represents the decoder acceleration measured as the ratio between the decoding time of the test set using a system augmented with PLP pruning and the decoding time of a baseline system that does not use any look-ahead pruning. Table 2 lists WER and TF values for 6 experiments. Each experiment used one of the *Catch-All*

models *S1-S6*, as described in Table 1. The baseline system achieved 6.7% WER for this testing data set.

The first set of experiments tested if using a model-specific pruning threshold, according to equation (8), can provide any gain in performance. These pruning thresholds, TH_{Lq} , were initially selected as the minimum posterior probability values calculated for each state, *q*, in the training data. Preliminary experiments showed that this minimum posterior probability thresholds needed to be slightly increased for better performance. Table 2 lists the WER and TF results for using each of the *S1-S6 Catch-All* models. The first row displays the results of directly using the minimum posterior values to estimate the threshold TH_{Lq}. Rows 2 to 5 displays the results of adding fixed values, in the log domain, to this minimum estimate.

Model	TH_{Lq}	min	min+1	min+2	min+3	min+4
S1	WER	7.23	7.12	7.12	**7.10**	7.10
	TF	0.77	0.54	0.5	**0.41**	0.69
S2	WER	7.3	7.18	6.96	**6.94**	6.89
	TF	0.74	0.56	0.45	**0.36**	0.63
S3	WER	7.32	7.13	6.94	**6.91**	6.93
	TF	0.75	0.54	0.48	**0.37**	0.65
S4	WER	7.32	7.15	6.94	**6.93**	6.95
	TF	0.73	0.51	0.45	**0.34**	0.61
S5	WER	7.12	6.97	6.93	**6.91**	6.91
	TF	0.73	0.53	0.44	**0.36**	0.58
S6	WER	7.9	7.91	7.94	8.23	8.45
	TF	0.65	0.55	0.51	0.43	0.37

Table 2. PLP performance

Results in the fourth column of Table 2 show that by using model-specific thresholds, adjusted by addition of +3, we can get 66% acceleration in the decoding time while keeping almost the same decoding accuracy. Also the *Catch-All* models *S1-S5* showed very similar performance. Models *S6* showed significant degradation in WER. This leads us to the conclusion that while the loss in the likelihood score estimated by the *Catch-All* model is within the [0, 0.1] bound, the decoding accuracy is hardly affected.

Finally, an experiment was conducted to compare the performance of the proposed PLP technique with a fast-matching approach that utilizes a context independent (CI) model for estimating the look-ahead score [1]. We tested several variants of the CI models: 1-state with a total of either 175 or 675 Gaussian densities and 6-state models with a total of either 497 or 1926 Gaussian densities. Table 3 summarizes the results of this experiment.

Look-ahead Method	WER	TF
Baseline	6.70	---
PLP	6.93	0.34
CI 1-state 175-density	7.12	0.46
CI 1-state 675-density	6.82	0.51
CI 6-state 497-density	6.89	0.53
CI 6-state 1926-density	6.80	0.57

Table 3. Look-Ahead methods comparison

Results in Table 3 show that the best performing CI model is the 1-state and 675 densities model that can achieve a 49% reduction in decoding time with minor degradation in accuracy. Further acceleration can be achieved with the 1-state and 175 densities model but with the price of large lose in accuracy. The other CI models did not show any gain in performance. These results show that our proposed PLP technique clearly outperform the CI model based look-ahead approach.

5. Summary and Conclusions

In this work we have shown that estimates of the state posterior probabilities can effectively be computed and utilized to develop a fast matching technique for HMM-based recognition systems. We showed that a *Catch-All* model can reduce the computational effort for obtaining these probabilities by a factor of 20 without serious reduction in accuracy. Based on these posterior probabilities estimates, a novel Posterior Look-ahead Pruning (PLP) technique was proposed and proved to be capable of providing a remarkable speed-up for a Viterbi-decoder. With this technique we were able to triple the speed of our spontaneous speech recognition system with only 0.23% absolute loss in the decoder accuracy. This gain in the acceleration was achieved by using model-specific pruning thresholds in place of the single global threshold. For the WSJ data sets these individual thresholds had to be adjusted by adding some fixed values. Such heuristic adjustments resulted in a significant gain in performance. Some further work may be conducted on the estimation of these model-specific posterior thresholds. Several smoothing and clustering techniques could be applied in order to get better threshold estimates for sparsely represented HMM states to gain resistance to extreme outliers among the posterior probability estimates on the training data.

6. References

[1] S. Ortmanns, H. Ney, A. Eiden, and N. Coenen, "LookAhead Techniques for Fast Beam Search," *in Proceedings of ICASSP97*, pp. 1783-1786, 1997.

[2] H. Ney, R. Haeb-Umbach, B.H. Tran, and M. Oerder, "Improvements in Beam Search for 10000 Word Continuous Speech Recognition," *in Proceedings of ICASSP92*, vol. I, pp. 9-13. 1992.

[3] S. Renals and M. Hochberg, "Efficient Search Using Posterior Phone Probability Estimates, " *in Proceedings of ICASSP95*, Vol. 1, pp. 596 - 599, May 1995.

[4] M. Y. Hwang, X. Huang and F. Alleva, "Predicting unseen triphones with senones". *In Proc. ICASSP93*, pp 311-314. 1993.

[5] B. Mak, and E. Bocchieri, "subspace Distribution Clustering Hidden Markov Models," *IEEE Transactions On Speech And Audio Processing*, Vol. 9, No. 3, pp. 264-275, March 2001.

[6] P. C. Loizou and A. S. Spanias, "High-performance alphabet recognition," *IEEE Transactions On Speech And Audio Processing*, Vol. 4, pp. 430-445, Nov 1996.

[7] E. Bocchieri, "Vector quantization for the efficient computation of continuous density likelihoods," *in Proceedings of ICASSP93* ,Vol. 1, pp. 692-695. 1993.

[8] S. M. Herman and R. A. Sukkar, "Variable threshold vector quantization for reduced continuous density likelihood computation in speech recognition," *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 331-337. 1997.