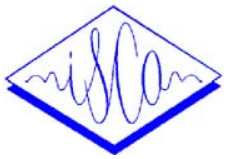

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

ENHANCED LIKELIHOOD COMPUTATION USING REGRESSION

Peter de Souza, Bhuvana Ramabhadran, Yuqing Gao, Michael Picheny

IBM T. J. Watson Research Center P. O. Box 218,
Yorktown Heights, NY 10598
(desouza,bhuvana,yuqing,picheny)@us.ibm.com

10.21437/Eurospeech.1999-305

ABSTRACT

In a rank based large vocabulary continuous speech recognition system [1], the correct leaf is expected to occupy the top rank positions. An increase in the number of times the correct leaf occurs in the top rank positions translates to an increase in word accuracy. In order to achieve low error rates, we need to discriminate the most confusable incorrect leaves from the correct leaf by lowering their ranks. Therefore, the goal here is to increase the likelihood of the correct leaf of a frame, while decreasing the likelihoods of the confusable leaves. In order to do this, we use the auxiliary information from the prediction of the neighboring frames to augment the likelihood computation of the current frame. We then use the residual errors in the predictions of neighboring frames to discriminate between the correct (best) and incorrect leaves of a given frame. In this paper, we present a new algorithm that incorporates prediction error likelihoods into the overall likelihood computation to improve the rank position of the correct leaf. Experimental results on the Wall Street Journal task and an in-house large vocabulary continuous speech recognition task show a relative accuracy improvements in speaker-independent performance of 10%.

1. INTRODUCTION

The IBM Continuous speech recognition system used here uses a set of phonetic baseforms and context dependent models. These models are are built by constructing decision tree networks that query the phonetic context to arrive at the appropriate models for the given context. A decision tree is constructed for every arc (sub-phonetic unit that corresponds to a state of the three state HMM). Each terminal node (leaf) of the tree represents a set of phonetic contexts, such that the feature vectors observed in these contexts were close together as defined by how well they fit a diagonal Gaussian model. The feature vectors at each terminal node are modeled using a Gaussian mxture density with each Gaussian having a diagonal covariance matrix. The IBM system also uses a rank-based decoding scheme [1]. The rank $r(l,t)$ of a leaf l at time t is the rank order of the likelihood given the mixture model of this leaf in the sorted list of likelihoods computed using all the models of all the leaves in the system and sorting them in **descending order**. In a rank-based system the output distributions on the state transitions of the model are expressed in terms of the rank of the leaf. Each transition with arc label a has a probability distribution on ranks which typically has a peak at rank one and rapidly falls off to low probabilities for higher ranks. The probability of rank $r(l,t)$ for arc a is then used as the probability of generating the feature vector at time t on the transition with arc a.

The more number of times a correct leaf appears in the top rank positons, the better the recognition accuracy. In order to improve the rank of the correct leaf, its likelihood score has to be boosted up relative to other leaves. This implies that the likelihood score for the correct leaf will be increased while those of the incorrect leaves will be decreased. A scheme to increase the likelihood of the correct leaf that captures the correlation between adjacent vectors using correlation models was introduced in [2]. In this paper, we describe a regression scheme to capture the same correlation. The regression predicts the neighboring frames of the current frame of speech. We then *smooth* the prediction error likelihoods into the overall likelihood computation to improve the rank position of the correct leaf, without increasing the complexity of the HMMs.

2. ALGORITHM

The idea in [2] was to do away with the assumption that given the output distribution at time t, the acoustic observation at time t is independent of that at time $t-1$, or depends only on the transition take at time t ($P(y_t|s_t)$). The manner in which y_{t-1} differs from the mean of the output distribution from which it is generated, influences the way that y_t differs from the mean of the output distribution from which it is generated. This is achieved by conditioning the probability of generating y_t on the transition at time t, the transition at time $t-1$ and y_{t-1},

$$P(y_t|s_t, s_{t-1}, y_{t-1}) \quad (1)$$

Incorporating this into a HMM would in effect square the number of output distributions and also increase the number of parameters in each output distribution. When the training data is not sufficient, the benefit of introducing the correlation concept may not be seen. Alternatively the probability could be conditioned only on the transition taken at time t and the output at y_{t-1},

$$P(y_t|s_t, y_{t-1}). \tag{2}$$

The output distribution for Eq. 2 has the form,

$$P(y_t|s_t, y_{t-1}) = \det W^{1/2} \frac{1}{(2\pi)^{n/2}} exp\left[-\frac{1}{2}(Z'WZ)\right] \tag{3}$$

where Z is given by

$$(y_t - (\mu_t + C(y_{t-1} - \mu_{t-1}))) \tag{4}$$

and C is the regression matrix given by

$$C = \sum (y_t . y_{t-1})/|y_t|^2 \tag{5}$$

This form only increases the number of parameters in each output distribution and not the number of output distributions, making it computationally attractive. However, from a modeling perspective, it is less accurate than Eq. 1 because the distribution from which y_{t-1} was generated and its deviation from its mean are unknown. There is an important trade-off between the complexity of an acoustic model and the quality of the parameters in that model. The greater the number of parameters in a model, the more variance there will be in the estimates of the probabilities of these acoustic events.

The fundamental idea behind the regression scheme introduced here is that if a leaf is the correct leaf for a particular frame, it will be correlated with its immediate neighbors. Therefore, we should be able to use the leaf that describes the current frame of speech to also predict its neighboring frames. In other words, s_t and s_{t-1} are assumed to be the same. If it is the correct leaf, then the prediction error vectors will be small, otherwise it will be large. The error vectors can then be modeled with a diagonal Gaussian mixture and the error likelihoods can be weighted and used to augument the overall likelihood computation of the leaf. As can be seen from our experiments, this results in improved ranks. The regression predicts the neighboring frames of the current frame of speech. We then *smooth* the prediction error likelihoods into the overall likelihood computation to improve the rank position of the correct leaf, without increasing the complexity of the HMMs. The resulting output distribution has a form similar to that of Eq. 2. However, Z is now given by,

$$(y_t - (\mu_t + wC(y_{t-1} - \mu_{t-1}) - w(y_{t-1} - \mu_{t-1}))) \tag{6}$$

where w is the weighting factor used to smooth the error likelihoods with the usual Gauusian mixture likelihoods. Computationally, this scheme has the same number of parameters as Eq. 2. However, from a modelling point of view it is more accurate because of the presence of the additional term, which is the deviation of y_{t-1} from its mean. In going from Eq. 1 to Eq. 2, the dependence on s_{t-1} was dropped. Here, we are retaining the dependence, but constraining it to be the same as s_t.

3. IMPLEMENTATION

At the time of training the feature vectors are tagged with the Gaussian that best describes the state that they align to. The tagging of the vectors to the gaussian that models it the best, ensures that we are using the best possible data for prediction. The prediction is done at the gaussian level, hence, the training data vectors that are tagged with the same gaussian are used to estimate the regression coefficients (See Section 3.1). This will subsequently be used in the prediction of the neighboring frames. Forward and backward regression coefficients are estimated for each gaussian. The residual error vectors are then computed. In addition to the conventional set of gaussian mixture models, we now use these error vectors to build two sets of gaussian mixture models, one each for the forward and backward prediction. In our implementation, instead of directly modeling Eq. 6, we have broken it into two distributions, the usual Gaussian mixture distribution on the feature vectors and the Gaussian distribution on the prediction error vectors and combined them using the smoothing factor w. We have also extended Eq. 6 to incorporate the errors in the prediction of the both the previos and succeeding frames.During recognition, we use these three sets of gaussian models to calculate three sets of likelihoods for each frame of speech and use their weighted combination to get the final likelihood score. Because of the incorporation of the forward and backward prediction likelihoods into the final likelihood, the likelihood for the correct gaussian has been increased while the likelihoods for the incorrect gaussians have been decreased. This translates to increasing the rank of the correct leaf while decreasing the rank of the incorrect leaf, thereby introducing discrimination.

3.1. Computation of regression coefficients

The details of the algorithm are as follows (Refer Figure 1). Since the regression is done at the Gaussian level, there are as many regression coefficients as there are gaussians.

Regression coefficients are estimated for each gaussian as follows:

$$C_{b,i} = \sum (y_t . y_{t-1}) / |y_t|^2$$

$$C_{f,i} = \sum (y_t . y_{t+1}) / |y_t|^2$$

where the sum runs over all y_t which gaussian i models best, y_t, y_{t-1} and y_{t+1} are the cepstral vectors corresponding to the speech at time t, $t-1$ and $t+1$ respectively, and i is chosen from the mixture-gaussian system containing M gaussians built from all of the training data. Note that these coefficients are the same ones given in Eq. 5.

The residual error vectors are then computed as

$$e_{b,t} = C_{b,i} y_t - y_{t-1}$$

$$e_{f,t} = C_{f,i} y_t - y_{t+1}$$

where $C_{b,i}$ is the backward regression coefficient for the gaussian i that best models y_t, $C_{f,i}$ is the forward regression coefficient for the gaussian i that best models y_t.

Each of the residual error vectors, $e_{b,t}$ and $e_{f,t}$ also has the same Gaussian tag i, as each x_t. Each regression coefficient is a m X n matrix, where m is the dimensionality of the feature vector being predicted, and n is the dimensionality of the feature vector used to predict it plus a constant term, to produce a regression of the form $\hat{y} = Ay + B$. We now model these error vectors. All the $e_{b,t}$ vectors which have the same tag are used to build a single diagonal Gaussian with $\mu_{b,i}$ mean and $U_{b,i}$ variance. This way we ensure that the number of gaussians used to model the $e_{b,t}$ vectors are the same as in the original system. It is important to note here that since, the regression is done at the Gaussian level, the regression coefficients are also tied to the Gaussians. During recognition, the Gaussian i is used to determine which regression coefficient has to be used for predicting the neighboring frames. By using the same number of Gaussians to model the prediction error vectors, the likelihood computation can be simplified to a one-to-one weighted linear combination of the three sets of likelihoods.

3.2. Likelihood Computaion

During recognition, we use the three sets of gaussian models built above to calculate three sets of likelihoods for each frame of speech and use their weighted combination to get the final likelihood score.

$$L_{t,i,c} = logN(x_t; \mu_i, U_i)$$

is the standard gaussian likelihood of a hidden-markov model system, where N(.) denotes the Gaussian density function with with mean μ_i and variance U_i, and x_t is the test data vector.

The backward-residual-error likelihoods are computed as :

$$L_{t,i,b} = logN(e_{test,b,t}; \mu_{b,i}, U_{b,i})$$

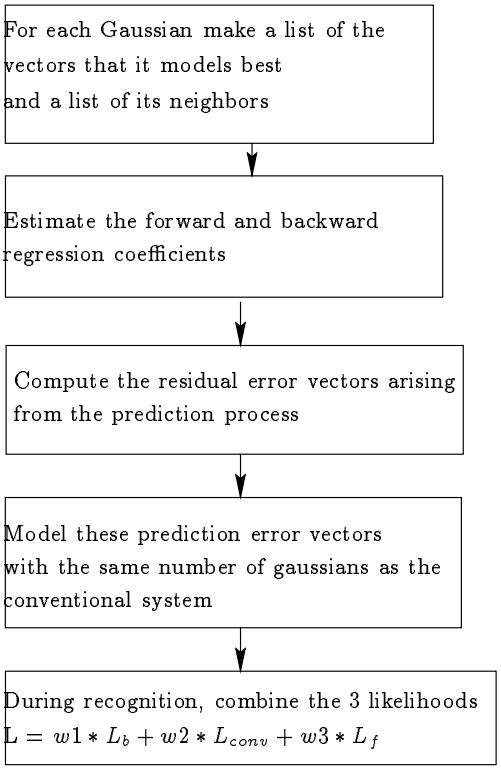

Figure 1: Enhanced Likelihood Computation

The forward-residual-error likelihoods are computed as :

$$L_{t,i,f} = logN(e_{test,f,t}; \mu_{f,i}, U_{f,i})$$

where $e_{test,b,t} = C_{b,i} . y_t - y_{t-1}$, $e_{test,f,t} = C_{f,i} . y_t - y_{t+1}$.

The final likelihood is given by,

$$L_{t,i} = w_1 L_{t,i,c} + w_2 L_{t,i,b} + w_3 L_{t,i,f}$$

where w_1, w_2 and w_3 are the weights assigned to the three likelihoods.

Because of the incorporation of the forward and backward prediction likelihoods into the final likelihood, the likelihood for the correct gaussian has been increased while the likelihoods for the incorrect gaussians have been decreased, thereby improving the rank of the correct leaf.

3.3. Variations in Regression Coefficients Computation

In our experiments, we predicted a 39 dimensional vector from a 13 dimensional vector. Hence each regression coefficient is a 39 x 14 dimensional matrix. It is worthwhile to note here that the goal is not to obtain perfect prediction but to get good dicriminant prediction between leaves. Other variations on increasing the number of dimensions used to predict the neighboring vectors were explored, but these variations did not provide a significant improvement

System	Baseline	New Scheme
WSJ	9.1%	8.1%
LVCSR task	12.8%	11.8%
After EM	12.54%	11.6%

Table 1: Recognition error rates using the usual Gaussian mixture likelihoods v/s Gaussian mixture likelihoods plus Error likelihoods

in accuracy over the current scheme. The regression was performed at the Gaussian level. If a gaussian was built from sparse data, then data was borrowed from other gaussians modelling the same context dependent arc (leaf). If this was still insufficient, data was borrowed from other leaves of the same arc.

4. EXPERIMENTS

The speech recognition system uses an alphabet of 52 phones. Each phone is modelled with a 3-state left-to-right HMM. The acoustic front-end uses a cepstral feature vector extracted every 10 ms, along with $\Delta +$ $\Delta\Delta$ and sentence based cepstra mean normalization.

The training data for the two systems were different. For the WSJ task, we used the wsj0 training data set. For the second large vocabulary continuous speech recognition task, we used an in-house data base consisting of 100 hours of training data collected from 2000 speakers. Both systems had approximately 3000 leaves.

The WSJ task had approximately 9 speakers and 6000 words in the test set and the LVSCR task had 10 speakersand 11000 words in the test set.

The use of different systems indicates that this algorithm provides consistent gains. However, it remains to be seen if this algorithm will provide similar gains on systems built with other techniques such as discriminative training or other model selection criteria.

The smoothing weights: $w1$, the weighting factor for the backward prediction error likelihood, $w2$, the weighting factor for the usual Gaussian mixture likelihood, and $w3$, the weighting factor for the forward prediction error likelihood, were chose to be 0.2, 0.6 and 0.2 respectively. A few variations on the choice of these factors were explored, again, without significant changes in the decoded results.

5. RESULTS

The results are tabulated in Table 1. It can be seen that there is approximately a 10% relative increase in the recognition accuracy when using the enhanced likelihood computation scheme over the usual Gaussian mixture likelihood computation.

When the regression coefficients are re-estimated using gaussians obtained from EM training, the resultant gaussians have better estimated parameters to model the data. The accuracy gets marginally better (See Table 1). This is because we are obtaining a better estimate of the regression coefficients and therefore predicting the neighboring frames better. This reinforces the likelihood computation in a manner that boosts the likelihood of the correct leaf for a given frame of speech.

The number of gaussians in the conventional system and the error gaussians are kept the same, so that the likelihoods from these gaussians can easily be incorporated into the overall likelihood. If the prediction was done at a higher level, i.e., at the leaf level or at the arc level, the number of gaussians would have been tied to the number of leaves or arcs. The rationale behind computations at gaussian level is that a finer resolution in modelling can be obtained in this manner.

6. CONCLUSIONS AND FUTURE WORK

It has been demonstrated that computing better ranks results in better recognition accuracy. In using this computation scheme, the average rank of the correct leaf increased from 21 to 15. Augumenting the conventional likelihood computation with prediction error likelihoods has resulted in increased accuracy. This algorithm can be extended such that the regression coefficients are computed for every leaf instead of every gaussian. This scheme can also be used in a rescoring framework when the first pass has been done using the conventional method of computing likelihoods. It can also be used as a confidence measure for estimating the prediction ability of the system, to decide if data can be used for unsupervised adaptation. The prediction process can be expanded to include more neighboring frames instead of just one. Also, instead of using cepstral feature vectors, feature vectors obtained using an LDA transormation can be used for the forward and backward prediction. Using this scheme of computing likelihoods has considerably increased the number of computations and the memory usage by a factor of three. Schemes that reduce the computations and memory cost using a vector quantization scheme are currently being explored.

7. REFERENCES

[1] Bahl et. al, "Robust-methods for using context-dependent features and models in a continuous speech recognizer," ICASSP 1994, Vol. 1, pp. 533-536.

[2] Brown, P. F., "The Acoustic-Modelling Problem in Automatic Speech Recognition", Ph. D. thesis, IBM RC 12750, 1987.