TRAINING OF STREAM WEIGHTS FOR THE DECODING OF SPEECH USING PARALLEL FEATURE STREAMS

Xiang Li and Richard M. Stern

Department of Electrical and Computer Engineering and School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213 USA
{xiangl, rms}@cs.cmu.edu

ABSTRACT

In speech recognition systems, information from multiple sources such as different feature streams can be combined in many different ways to yield better recognition accuracy. In general, information may be combined at the level of the incoming feature vectors, at the level of the decoding process, or after hypothesis generation. In this paper we focus on the specific case where parallel streams of features are used simultaneously during search to generate a hypothesis, or a set of hypotheses. In this case the contributions of the individual features to the score associated with a frame of speech must be weighted appropriately during search. In this paper we present an offline data-driven algorithm for determining the weights to be associated with each feature stream for combining acoustic likelihoods for each frame. Experimental results show that the word error rates (WERs) obtained using the proposed algorithm are lower than those obtained using conventional schemes for parallel feature combination.

1. INTRODUCTION

The general motivation for using information from parallel feature streams in speech recognition systems has been extensively discussed in literature. The algorithms currently used to combine information from parallel feature sets can be broadly viewed as belonging to one of three types: those that concatenate different feature vectors together to form a larger feature vector (with independent or correlated feature streams) and perform recognition based on the values of these combined features (*e.g.* [1]), those that combine information from various feature streams by combining the probability scores of recognition classes via some combination functions (*e.g.* [2][3]), and those that generate independent hypotheses from separate features, which are later combined in various ways to generate the final hypotheses (*e.g.* [4][5][6]). In any particular application which must use parallel information sources, the choice of the specific type of combination method to be used depends on the type of application and the resources (time, computation etc.) available to it.

In this paper we first consider the difference between fusion of information at the recognizer hypothesis level (*e.g.* [5][6]) and fusion of information at the level at which state acoustic likelihoods are computed. We then focus on the specific case of acoustic likelihood combination where information from multiple independent feature streams is combined at the state level in a judicious manner during the decoding process, keeping in mind that the contributions from each of the various feature streams may not be equally helpful for returning a good recognition hypothesis. In literature associated with this problem, there has been relatively little work associated with choosing appropriate weights for combining the contributions of the feature streams during decoding. In this paper we propose an offline data-driven algorithm to train such weights for a speech recognition system based on Hidden Markov Models (HMMs). The proposed algorithm uses a set of training data and, for a particular feature type, computes the value of a “loss” function over the entire observation sequence. This loss is a function of the differences between the desired state sequences obtained using forced alignments of training speech to its orthographical representation and the states obtained from the conventional (“blind”) decoding process. The aggregate losses for each feature type are accumulated over all observation sequences in the training data set, and a final set of normalized losses is computed. These normalized losses are then used to determine the weights to be associated with each feature type during recognition of speech from the same or similar acoustic domains. We call this algorithm *loss-based weighted combination.*

While the algorithm is based on simplistic considerations, it must be noted that the effect of acoustic likelihood combination on the recognition performance depends critically on the combination function that is used during probability updating. In this paper we use simple weighted combinations, with weights for the feature streams trained explicitly on heldout development data or other training data.

In the following section we describe our motivation for focusing on state acoustic likelihood as the combination target by briefly considering differences between combining information at the level of the recognition hypothesis versus the state acoustic likelihood. In Section 3 we discuss various methods of combining likelihoods during decoding. In Section 4 we propose the algorithm for training feature stream weights. In Section 5 we present experimental results and we present our conclusions in Section 6.

2. COMBINATION OF INFORMATION FROM PARALLEL FEATURE STREAMS

While the best stage for fusion of information in pattern classification system depends on the specific application and resources available, in the specific case of continuous speech recognition system, the pruning that is inherent in the search stage causes combination at the level of state acoustic likelihoods to outperform combination of recognizer hypotheses in many situations.

Consider a simple example where we must recognize the given

0-7803-7663-3/03/$17.00 ©2003 IEEE

signal as one of a set of words, $W_1, W_2 \ldots, W_J$. We are given two sources of information in the form of two different features of the signal, f_1 and f_2 . The score of each word from each feature is represented as S_{ij} , where *i* and *j* represent the word index and feature index respectively. The score S_{ij} can either be the overall score for the entire duration of word W_i , corresponding to the recognition result; or it can be the probability of the state associated with the word W_i in each frame, which corresponds to the acoustic likelihood. The recognition task based on the single feature f_j can be stated as:

$$\hat{W} = W_k : \mathrm{k} = \mathrm{max}_i\{S_{ij}\} \tag{1}$$

where $\hat{W}$ is the recognized word. When we combine the information from the two features together, the recognition task becomes

$$\hat{W} = W_k : \mathrm{k} = \mathrm{max}_i\{S_i'\} \tag{2}$$

where S_i' is the combined score for word W_i generated by using individual scores from features f_1 and f_2 for the *same* target W_i via a combination function *Func*:

$$S_i' = Func(S_{i1}, S_{i2}) \tag{3}$$

The term *same* target in combination simply implies that the new score S_i' can only be generated based on original scores from the same word S_{i1} and S_{i2} , instead of S_{i1} and some S_{j2} .

Typically, because of the pruning that must be performed during search in a continuous speech recognition system, the space of hypotheses (which could be N-best word strings or word lattices) does not represent the complete search space that could have been considered during the decoding process. If we perform combination based on the actual recognition hypothesis spaces, we may face the problem that scores for some recognition candidates (which could be words or word fragments) are present for some feature streams but not for others. This situation that will either invalidate our *same-target* assumption in the combination process, or put a very strong constraint on choosing combination function *Func*. These two problems tend to vitiate the benefits of combination.

As an example, suppose that there are only four possible words W_1, W_2, W_3, W_4 which must be recognized, and that the recognition output from each individual feature is in the 2-best list format. Let Feature f_1 output word W_1 and W_2 with score S_{11} and S_{21} respectively, while Feature f_2 scores word W_3 with S_{32} and word W_4 with S_{42} . The scores for the words W_3 and W_4 from feature f_1 and for the words W_1 and W_2 from feature f_2 are missing. How can we choose a combination function *Func* that enables us to generate the updated score for all those word candidates even though half of the required inputs are missing?

Fortunately, one category of combination function, transitive operation (such as the Max operation [6]), does enable us to deal with the situation of missing scores. In the above example, the recognition task could be stated as Eq. (4) if we use the *"Max"* operation as combination function:

$$\begin{gathered} \hat{W} = W_k \\ k= \\ Max\{max(S_{11},S_{12}), max(S_{21},S_{22}), max(S_{31},S_{32}), max(S_{41},S_{42})\} \end{gathered} \tag{4}$$

Because of the transitive property of the "Max" operation, we can rewrite Eq. (4) as Eq. (5):

$$\begin{gathered} \hat{W} = W_k \\ k = Max\{max(S_{11}, S_{21}, S_{31}, S_{41}), max(S_{12}, S_{22}, S_{32}, S_{42})\} \\ k = Max\{(S_{11}, S_{21}), (S_{32}, S_{42})\} \end{gathered} \tag{5}$$

where the item inside () bracket is the recognition result from each individual feature. The transitive property of the *"Max"* operation will enable us to generate the combined results even though some scores are missing.

On the other hand, a transitive function like *"Max"* is just one category of many possible combination functions, and many other functions (*e.g.* summation, multiplication, etc.) may represent the relationship between different features much better than the *Max* operation. If we can develop some other combination scheme that can utilize those functions without the missing score problem, we will have much more flexibility, and consequently at least equal performance compared with using only the *"Max"* function.

While the discussion above has been couched in the language of combination of recognizer hypotheses (as in [5]), scores from multiple feature streams can also be combined at the decoder level, based on *a posteriori* probability scores on a state-by-state basis. Combining information at the level of the decoder states based in this fashion has the advantage that the missing score problem described above will vanish, since probability scores for the states are generated before the pruning stage in speech recognition systems.

3. COMBINING ACOUSTIC LIKELIHOODS AT THE STATE LEVEL

In state-level combination of acoustic likelihoods, we first update the overall acoustic likelihood S_i of each state *i* based on the likelihood values S_{ij} from each individual feature f_j using

$$S_i = Func(S_{i1}, S_{i2}, \ldots, S_{iN}) \tag{6}$$

where *Func* is the combination function. Once we generate the updated likelihood value, the normal pruned search and language modelling processes are applied to generate the final combination result.

For the given feature sets, the effect of state combination depends critically on the combination function *Func*. We explored the effects on WERs of the following simple combination functions:

1) Maximum:

$$S_i = Max(S_{i1}, S_{i2}, \ldots, S_{iN}) \tag{7}$$

2) Multiplication:

$$S_i = \prod_{j=1}^{N} S_{ij} \tag{8}$$

3) Summation:

$$S_i = \frac{1}{N}\sum_{j=1}^{N} S_{ij} \tag{9}$$

4) Weighted summation:

$$S_i = \sum_{j=1}^{N} W_{ij} \cdot S_{ij} \tag{10}$$

5) Weighted maximization:

$$S_i = Max(W_{i1}S_{i1}, W_{i2}S_{i2}, \ldots, W_{iN}S_{iN}) \tag{11}$$

Among those functions, the summation and max combination function as in Eqs. (7) and (9) are actually special cases of the weighted summation and maximization operations as in Eqs. (10) and (11). We know that the benefit of combination actually comes from the differing abilities of the different features in representing different correct recognition classes that occur at the different states of decoding. Each recognition class has its own best feature in that the overall score generated from this feature is closest to the true score of that class. When we update the score for each class from multiple features, ideally we would like to assign greater weights to the more reliable features and smaller weights to the less reliable features. This paper concerns methods by which we attempt to determine the best weights for each feature.

4. TRAINING WEIGHTS FOR LIKELIHOOD COMBINATION FROM PARALLEL STREAMS

Since the weight of each feature stream in combination reflects the reliability of each feature in its prediction, our training algorithm first estimates the reliability of each feature via a *"loss"* computed from training data, and then assigns a weight to each feature according to this loss.

The algorithm can be described as follows. We define a *state vector* S_t^j for the j^{th} feature as an M -dimensional column vector of binary entries in each frame *t*, where M is the number of total HMM states. The entry in S_t^j corresponding to the true state of the HMM at time t is 1 and all other entries are 0. The *true* state of the system at time t is defined as determined by forced alignment to the correct transcription of the utterance. When S_t^j is so defined, it is easy to see that $E[S_t^j]$, the expected value of S_t^j given the whole observation sequence, is simply a vector of the *a posteriori* probabilities of the various states from the j^{th} feature at time *t.*

The *loss* incurred at time t of state i from the j^{th} feature is defined as

$$Loss(y^i_{j,t}) = \left|A_i(S_t^j - E[S_t^j])\right|^2 \tag{12}$$

where A_i is a row vector whose i^{th} element is 1 and all other elements are 0. The total loss over any set of observations for state i of the j^{th} feature is given by

$$L_{ij} = \sum_{t,obs} Loss(y^i_{j,t}) \tag{13}$$

where the summation is over all observations and over all time. We compute the total loss for the various states of the various features over the entire training set.

Once we obtain the total loss L_{ij} from training, we define the weight $W_{i,j}$ of each feature *j* in predicting the probability score of each state *i* as

$$W_{i,j} = \exp\left(\frac{-L_{ij}}{C}\right) / (NORM_i) \tag{14}$$

where

$$NORM_i = \sum_{j=1}^{N} \exp\left(\frac{-L_{ij}}{C}\right) \tag{15}$$

C is a parameter that controls how the difference of losses made by different features affect the reliability of those feature in their prediction. The smaller the value of *C*, the greater the relative emphasis that is applied to the more reliable features. The specific value of C is tuned in the validation set. Eq. (15) is somewhat arbitrary. Nevertheless, if a negative loss can be interpreted as the log likelihood of choosing feature stream *j* given state *i*, the weights generated from Eq. (14) are actually the posterior probabilities of choosing feature *j* given state *i* with a flat prior probability[7]. The determination of a more globally optimal transformation from $W_{i,j}$ to L_{ij} will be the subject of future research.

5. EXPERIMENTAL RESULTS

To compare the differences in effects produced by combinations of feature streams on the basis of weighted acoustic likelihood versus at the state level versus combinations of recognition hypotheses, we evaluated the performance of various combination functions using two speech corpora, 400 utterances from the DARPA Resource Management corpus (RM) and 700 utterances from the Telefónica Cellular Telephone corpus (TID), artificially corrupted by traffic noise at SNRs of 5 and 10 dB [6].

All experiments were conducted using the CMU SPHINX-III speech recognition system. The model structures used were a 3-state continuous HMM for RM, and 3-state semi-continuous model HMMs for the TID corpora. The feature sets used for the two TID corpora were standard MFCC and PLP. For the RM corpus, we designed three different feature sets to test how well the performance of various combination scheme depends on the feature set. The first feature set consisted of the standard MFCC and PLP features. The second set consisted of three different features derived using three different linear discriminant analyses (LDA). The first LDA was designed to maximize the difference between the first states in all HMMs, while the second and third LDAs were designed to maximize differences among all middle states and final states, respectively. The last feature set contains two LDA-based features, one designed to maximize differences among all the states, and the second designed to discriminate states belonging to the broad phone classes of vowels, fricatives, silence, stops and nasals.

For each testing corpus, we compare the performance of hypothe-

WER (%)	RM 1	RM 2	RM 3	TID 5 dB	TID 10 dB
Feature 1	10.29	10.72	8.36	25.54	12.53
Feature 2	11.49	9.53	9.58	26.60	13.17
Feature 3	N/A	9.69	N/A	N/A	N/A
Hyp (Max)	9.68	9.39	7.98	25.62	11.93
Lat (Max)	8.91	9.25	7.62	24.76	11.32
Sta (Max)	8.09	8.83	7.37	23.96	11.32
Sta (MaxW)	7.72	8.64	7.23	22.96	10.99
Sta (Sum)	8.30	8.83	7.15	23.16	11.24
Sta (SumW)	7.93	8.54	6.94	22.59	10.63
Sta (Prod)	10.12	31.51	7.42	22.46	10.49

Table 1. Recognition accuracy of various combination schemes on all testing corpora. Hyp, Lat, and Sta refer to hypothesis combination, lattice combination, and state-based combination, respectively. Max, Sum, Prod, MaxW, and SumW refer to the maximization, summation, product, weighted maximization, and weighted summation combination functions, respectively.

sis combination [5] and lattice combination [6] with the acoustic likelihood combination using all combination functions discussed in the paper. Table 1 shows (WERs) obtained from the combination experiments. Table 2 gives the matched-pair test [8] of statistical significance measurements between selected pairs of combination methods/functions.

6. DISCUSSION AND CONCLUSIONS

We first note that the WERs obtained with weighted maximization and weighted summation is always better than that obtained with flat maximization and summation, and the differences, particularly for the summation algorithms, are significant.

We also note that the best-performing combination functions at the state-combination level, weighted summation or maximization, always provide lower WERs than the best combination approach based on recognition results, the lattice combination method [6], and that the differences in scores are for the most part highly significant.

This difference is seen most directly in comparisons of results obtained using lattice combination and acoustic-likelihood-based state combination with the maximization combination rule. Even though both methods use the same combination function, the WERs obtained using acoustic likelihood are equal to or better than those obtained using lattice combination. We believe that this is a reflection of the fundamental importance of the state as the basic modeling unit in speech recognition and hence the best site for fusion of feature information.

The recognition results also reveal dramatic differences in the effects of the various combination functions. For example, the multiplication operation is very sensitive to the performance of the individual feature streams, and a single ineffective feature stream can severely degrade accuracy, even though most streams may have very good performance. We believe that this effect underlies the large WER observed with the “Prod” operation for the RM2 corpus. Similarly, the Max and Sum operations produce differing results for the various corpora, even though the Max operation is frequently used as a substitute for the Sum operation in HMM decoding. A more general approach to the selection of a best combination function will be the subject of future research.

P	RM 1	RM 2	RM 3	TID 5 dB	TID 10 dB
Sum:SumW	0.06	0.06	0.08	0.28	0.03
Max:MaxW	0.16	0.19	0.16	0.04	0.07
Sum:Max	0.23	1	0.3	0.01	0.8
SumW:MaxW	0.35	0.49	0.14	0.54	0.37
Prod:Sum	0	0	0.42	0.18	0.02
Prod:Max	0	0	0.85	0.01	0.01
Lat:Sum	0.12	0.19	0.17	0.01	0.81
Lat:Max	0.04	0.20	0.16	0.18	0.95
Lat:SumW	0.04	0.03	0.03	0.01	0.18
Lat:MaxW	0.01	0.05	0.11	0.01	0.40
Hyp:Sum	0.01	0.07	0.02	0.01	0.18
Hyp:Max	0.01	0.09	0.04	0.02	0.21

Table 2. Results of the matched-pair test [8] of statistical significance for selected pairs of combination methods and combination functions.

ACKNOWLEDGEMENTS

The authors thank Dr. Rita Singh for all many helpful discussions in the preparation of this paper. This research was sponsored by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

REFERENCES

[1] Neti, C., Potamianos, G., Luettin, J., Matthews, I., Glotin, H., and Vergyri, D. “Large-vocabulary audio-visual speech recognition: A summary of the Johns Hopkins Summer 2000 Workshop”, *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on,* pp. 619 -624, 2001

[2] Halberstadt, A. K. and Glass, J. R. “Heterogeneous measurements and multiple classifiers for speech recognition”, *Proc. ICSLP 1998,* pp. 995-998, 1998

[3] Bourlard, H. and Dupont, S. “A new ASR approach based on independent processing and recombination of partial frequency bands, *Proc. ICSLP 1996,* pp. 422-425, 1996

[4] Fiscus, J.G. “A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER),” *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 347-354, 1997

[5] Singh, R., Seltzer, M., Raj, B., and Stern, R.M. “Speech in noisy environments: robust automatic segmentation, feature extraction, and hypothesis combination”, *Proc. ICASSP 2001*, Vol 1, pp.273-276, 2001.

[6] Li, X., Singh, R., and Stern, R.M. “Combining search spaces of heterogeneous recognizers for improved speech recognition”, *Proc. ICSLP 2002,* Vol 1, pp. 405-408, 2002.

[7] Kivinen, J. and Warmuth, M. K. “Averaging Expert Predictions” *Proc. 4th European Conference on Computational Learning Theory, volume 1572 of LNAI,* pp. 153-167,1999.

[8] Gillick, L. and Cox, S.J. “Some statistical issues in the comparison of speech recognition algorithms”, *Proc. ICASSP 1989,* Vol 1, pp. 532-535, 1989.