USING MUTUAL INFORMATION TO DESIGN FEATURE COMBINATIONS

Daniel P.W. Ellis[1] *and Jeff A. Bilmes*[2]

[1]International Computer Science Institute, Berkeley CA, USA
[2]Dept. of Electrical Engineering, University of Washington, Seattle WA, USA

ABSTRACT

Combination of different feature streams is a well-established method for improving speech recognition performance. This empirical success, however, poses theoretical problems when trying to design combination systems: is it possible to predict which feature streams will combine most advantageously, and which of the many possible combination strategies will be most successful for the particular feature streams in question? We approach these questions with the tool of conditional mutual information (CMI), estimating the amount of information that one feature stream contains about the other, given knowledge of the correct subword unit label. We argue that CMI of the raw feature streams should be useful in deciding whether to use independent or conjoint acoustic models for the streams; this is only weakly supported by our results. We also argue that CMI between the outputs of independent classifiers based on each stream should help predict which streams can be combined most beneficially. Our results confirm the usefulness of this measure.

1. INTRODUCTION

A perennially successful approach to improving speech recognition performance is to use several feature streams. Different feature extraction algorithms can reveal complementary aspects of the original acoustic signal, leading to more accurate classification in acoustic models. In our own work and that of others, systems combining multiple feature sources consistently outperform single-feature-source baselines, even when the combination schemes are naive and in spite of highly redundant information in the feature streams [1, 2, 3, 4]. This is not so surprising: as long as the separate feature streams contain some amount of complementary information, and provided our combination scheme can exploit it, more information should always be better.

Within the context of systems exploiting such feature-stream combination, there are several design choices whose optimal solutions are unknown. In this paper, we consider two of the most important: how to choose which feature streams to combine, and which combination strategies to employ. Choosing the streams to combine is complicated because it depends not only on the baseline utility of the feature stream as a basis for the desired speech-class discrimination, but also on the complementarity of the information in each stream. As an extreme example, combining the single best-performing feature stream with *itself* should not give much benefit, since the information in the two streams is entirely redundant and the combination is adding nothing. By contrast, we have seen cases in which adding a second stream that performs more than 10% worse than the original stream (in terms of relative Word Error Rate (WER)) affords a relative improvement of 5-10% in WER over the better stream (or 15-20% over the added stream), provided the two streams have significantly different properties [2].

We have also investigated a number of different combination schemes, both in terms of the point at which streams are combined (before or after the initial acoustic classifier, at the timescale of individual frames, or at some other synchronization point), and the precise rule used to combine them (for instance, combining probabilities by averaging, log-averaging, taking the max or taking the min) [5, 3, 6]. We have seen that the difference between best and worst combination schemes can easily exceed 20% relative, yet the best scheme depends on the particular streams to be combined and cannot easily be predicted.

Currently these design choices are made through a combination of intuition and empirical comparison. Since, however, it can take several weeks to train a speech recognition system for a large task such as Broadcast News, it would be preferable to find some simpler property of feature streams that could be used to decide when and how to combine them. In this paper, we investigate the usefulness of CMI in this role, making theoretical arguments about why the mutual information between different streams should relate to their properties in combination, then testing these arguments against some experiments with four different feature streams used on the Aurora noisy digits task [7].

The next section describes the task and our approach in more detail, including a description of our estimation of CMI. Section 3 presents the arguments that low CMI between classifier outputs should correlate with streams amenable to combination, and that low CMI between the feature streams should favor independent (rather than integrated) acoustic model spaces. Section 4 presents the experimental results for recognizers based on several different stream combinations, and compares them to the predictions based on CMI. Finally, we discuss the implications of the results and some future directions for this work.

2. APPROACH

In recent experiments with the Aurora noisy digits task [6], we conducted an exhaustive investigation into combinations between four feature streams: PLP cepstral coefficients, their deltas, and modulation-filtered spectrogram (MSG) features for two different modulation bands, 0-8 Hz and 8-16 Hz [8]. Our speech recognizer was the hybrid connectionist-HMM framework [9], in which a neural network serves as the acoustic model, estimating discriminative posterior probabilities for each subword class. These probabilities, converted to likelihoods, are then used in a conventional HMM decoder to find the most likely word sequence.

The four streams are summarized in table 1, which shows their relative performance as bases for recognition. Since the Aurora task includes 28 test conditions spanning a wide range of SNRs,

Stream	Description	Elements	Bsln WER Ratio
PLPa	PLP cepstra	13	105.9%
PLPb	deltas of PLPa	13	125.6%
MSGa	modspec 0-8Hz	14	112.7%
MSGb	modspec 8-16Hz	14	141.6%

Table 1: The four basic feature streams and their individual performance, expressed as the average ratio of per-condition WER to the HTK baseline system.

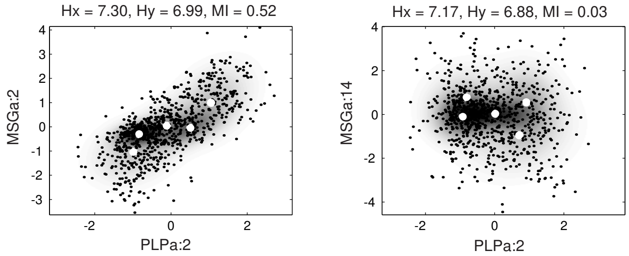

Figure 1: Examples of mutual information values calculated from Gaussian mixture models of joint distributions. Each plot shows data points overlaid on the resulting GM model; the white dots are the centers of the 5 Gaussians. The data on the left have a relatively high MI, those on the right very little MI.

it is most useful to look not at absolute WER but at the ratio of WERs compared to the Aurora baseline system (defined in HTK and using MFCC features plus deltas and double-deltas). This ratio is averaged across all 28 conditions to give the 'Baseline WER Ratio' figure.

We contrasted two combination schemes: feature combination (FC), where the streams are concatenated to make a single, larger feature space for which a single acoustic model is built, and posterior combination (PC), in which separate acoustic models for each stream calculate the posterior probability of each phone class, and these probabilities are combined and passed on to later processing. In our experiments, we combined posteriors by averaging in the log domain (i.e. taking the geometric mean). Modulo a factor of the prior probabilities, this is the theoretically correct approach if the two feature streams are conditionally independent. In practice, for speech recognition feature stream combination tasks, log-domain averaging has consistently out-performed other rules such as linear averaging, or taking the max or min [3].

Four streams offer six possible pairs, and each pair was combined both by FC (by training a new neural-net acoustic model on the concatenated feature vectors) and by PC (by combining the posterior probability outputs of neural-nets trained on each individual stream). The results are presented in section 4 and show a variation of 20% or more depending on the choice of streams and combination method.

We also investigated ways of combining all four streams. The best scheme, as reported below, was to use FC to combine the closely-related pairs of streams (i.e. cepstra and their deltas, and the two kinds of MSG features), then to use PC to combine the two resulting classifiers, further highlighting the difficulty in choosing the best combination scheme.

2.1. Estimating Conditional Mutual Information

The mutual information (MI) between two random variables is a measure of how much you discover about one variable given knowledge of the other. Formally, it is the difference between the joint and marginal entropies of the variables, that is, how much the entropy is reduced by evaluating it over the full joint distribution rather than summing the entropies of the distributions of each variable separately. Random variables that behave independently will have a mutual information of zero; the maximum mutual information exists between two observations that are deterministically related and is equal to their individual entropies (which must be the same).

Conditional mutual information (CMI) introduces a set of conditioning events; mutual information is calculated between the distributions of the two variables given a particular event, and the overall value is the expected value over all events. In our experiments, we condition upon the 'true' phone class (as determined by our forced-alignment training labels). The full MI between two streams will be dominated by the information relating to the current phone, present in all reasonable feature streams. In contrast, CMI reflects only the residual co-behavior of the two streams when the phone class is known, and is therefore a more sensitive measure of the relationship between streams.

The entropy of empirical data is typically measured by histogram methods, where the relative probabilities of different values are estimated by counting. MI requires this to be done for joint distributions – implying N^2 bins if N were adequate for the marginal variables – and CMI requires MI for each conditioning class, further subdividing the available training examples. To avoid problems arising from having too little data to make accurate estimates of joint distributions, we first fit a low-order Gaussian mixture model to the joint distribution, then use numerical methods to derive the MI of the model [10, 11]. We typically use 5 mixture components; increasing this to 20 made little change, so this value seems adequate. Figure 1 illustrates the modeling of some example data pairs, and figure 2 shows the complete matrix of element-to-element CMIs for the four base feature streams (before passing through the classifier).

We are interested here in the mutual information between multidimensional feature vectors, but our CMI estimation algorithm is practical only for pairs of scalar variables. To obtain an indication of the full stream-to-stream CMI, we use the average across all vector elements of the maximum CMI to any of the elements in the other stream. This is a kind of per-element CMI; calculating the full stream-to-stream CMI would need to incorporate the within-vector dependence as well.

3. CMI IN SYSTEM DESIGN

In this section we describe how we see CMI being useful in the design of combination schemes.

3.1. Choosing feature streams

As argued above, the streams that show the most benefit from combination will have the greatest amount of complementary, non-redundant information relevant to the classification task. One way to measure this is to look at the *outputs* of classifiers trained on

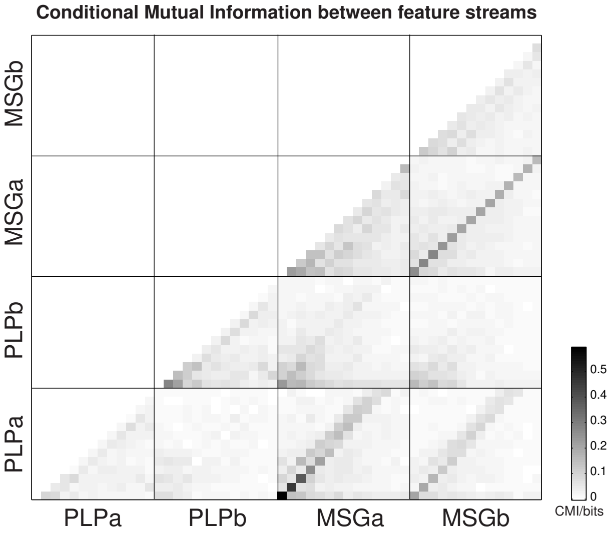

Figure 2: Full matrix of pairwise CMIs between the elements of the four feature streams. Only pairs beyond the leading diagonal are shown. To make the streams more comparable, the spectral MSG features were transformed by a DCT before CMI calculation.

each stream, which we assume have successfully extracted the relevant information from the feature streams. (In practice, we use log probabilities whose somewhat Gaussian distributions are easier to model than plain probabilities.) If the classifiers behave very similarly, then the useful information in the streams is effectively redundant. In particular, if the classifiers make the same errors, there is little hope of a gain from combining them.

Differences in classifier outputs suggest different information available in the streams, which are then promising candidates for combination. Since, for a well-modeled task, the majority of test cases are correctly classified by both classifiers, significant differences are enhanced by focusing on a subset of the test data known to give many errors. Pairs of feature streams whose classifier outputs exhibit little mutual information (i.e. uncorrelated behavior) on this 'difficult' subset are promising candidates for combination, since they appear to reflect the availability of different task-relevant information.

The design procedure this suggests is as follows: (a) Train single-stream classifiers based on each available feature stream; we assume that all streams have roughly equivalent performance in isolation. (b) Measure the CMI between each pair of classifier outputs over a set of 'difficult' cases; and (c) build multi-stream recognition systems based on the stream pairs with the lowest CMI, combined by some unspecified method.

3.2. Choosing combination methods

Looking at CMI between single-stream classifier outputs should indicate which streams can be combined, but does not tell us *how* to combine them. For this question, the CMI of the base features (i.e. before feeding into the classifier) may be helpful, at least for deciding between Feature Combination (FC, where a single model space is formed from both streams) and Posterior Combination (PC, where single-stream classifier outputs are combined, in our case by log-domain averaging).

Since FC models a single, composite feature space, it has the ability to exploit interdependence between the feature streams that may only be evident when the full joint distribution of the streams is available for modeling. In PC, by contrast, the streams do not meet until *after* the features have been converted to class probabilities, so no feature-space interactions can be learned. By the same token, each PC classifier operates in a lower-dimensional feature space and thus is better able to model a training set of fixed size.

From this perspective, the conditional mutual information between the feature streams should help predict if FC is worthwhile or unnecessary. If, given the correct class, one feature stream is largely independent of the other (i.e. their CMI is close to zero), then there is little structure to be learned by the larger FC model. A large CMI suggests interrelated streams that might benefit from FC.

4. RESULTS

Table 2 includes all six pairwise combinations of the feature streams from table 1, along with the best-performing 4-stream system, which combined FC combinations of PLPa+b and MSGa+b. The table shows overall word-error rate performance (again expressed as the average of per-condition ratios to the standard Aurora baseline) for both Feature Combination and Posterior Combination. Also shown are the conditional mutual information estimates, in bits, for both the base feature streams and the posterior probabilities from classifiers trained on those streams. The CMIs are measured over a special 'difficult' subset of the test data, chosen from utterances in which the baseline system made word errors.

As predicted, a small probability CMI correlates well with stream pairs, such as PLPa and MSGb, that show particular gains from combination. The four stream system actually has a larger probability CMI, but these numbers are not really comparable given the much better baseline performance of the stream pairs being combined in the 4-way system.

Although feature CMI varies over a larger range, it is only weakly correlated with the advantage of FC over PC; our argument that PC should be favored when feature CMI is small is not conclusively supported. Looking at just the first two lines, we do see that PC is much worse than FC for the MSG streams, which have a large feature CMI. However, in the next four lines, the PC system that performs least badly compared to FC occurs for PLPa with MSGa, which also has the largest feature CMI of that set. The 4-stream system of the final line, the only example in which PC outperforms FC, nonetheless has a relatively large feature CMI.

5. DISCUSSION AND CONCLUSIONS

It is disappointing to find little support for our prediction that base feature stream CMI should be low when PC is a relatively successful combination method. One explanation could be that the stream interdependence being measured is dominated by behavior that is irrelevant to the classification task.

A second problem could lie in the estimation of stream CMI from pairwise element CMI as described in section 2.1. Whereas the probability CMIs were measured between streams with very similar characteristics and internal correlation (since they were all approximations of the same posterior probability sequences), the feature CMIs are measured between very different stream types which might make comparisons between them irrelevant. As

Stream 1	Stream 2	FC WER Ratio	PC WER Ratio	Ftr CMI	Prob CMI
PLPa	PLPb	89.6%	97.6%	0.04	0.26
MSGa	MSGb	85.8%	101.1%	0.21	0.25
PLPa	MSGa	86.4%	88.3%	0.23	0.26
PLPa	MSGb	78.1%	86.3%	0.11	0.15
PLPb	MSGa	87.5%	89.7%	0.09	0.24
PLPb	MSGb	82.6%	89.9%	0.05	0.19
PLPa+b	MSGa+b	74.1%	63.0%	0.16	0.24

Table 2: Comparison of various feature stream combinations: Recognition performance (average per-condition ratio to baseline WER) for both Feature Combination (FC WER Ratio) and Posterior Combination (PC WER Ratio), along with CMI based on both the basic features (Ftr CMI) and on the posterior probability outputs of classifiers trained on individual streams (Prob CMI).

noted, our streamwise CMI estimates can be influenced by within-stream dependence, and this is varies among the different feature streams, as seen in figure 2.

Note that the only kind of Posterior Combination that we investigated was log-domain averaging. As mentioned above, this particular combination rule is related to the theoretically optimal method for conditionally independent feature streams i.e. streams with a feature CMI of zero (consistent with our argument that PC should be better when the feature CMI is small). Other combination rules correspond to different assumptions about the component streams, and might turn out to be more consistent with the CMIs we measured.

One factor not discussed so far is the influence of the baseline performance: judgments about stream combinations must respect the baseline utility of the individual streams, as well as indications of the complementarity of stream information as investigated here.

In summary, we argued that low CMI between classifier outputs indicated streams that would benefit from combination because they contained complementary information. This was borne out by our experimental results, and thus classifier output CMI could serve as a useful basis for system design, avoiding the slow process of evaluating complete systems based on all possible feature configurations.

We also investigated whether CMI could indicate which stream combination approach would be most beneficial for a particular pair of streams. We argued that low CMI between base features (i.e. *before* passing through the classifier) should indicate streams with relatively little structure in their joint distribution, obviating the need for the monolithic model space of Feature Combination, and permitting the use of the simpler models of Posterior Combination, which might be expected to train more successfully from limited data. This was not supported by our experimental results; perhaps a more sophisticated estimation of the CMI between feature streams would have greater success as a basis for system design.

6. ACKNOWLEDGMENTS

This work was supported by the European Union under the ESPRIT LTR project Respite (28149).

REFERENCES

[1] S. Sharma, D. Ellis, S. Kajarekar, P. Jain and H. Hermansky, "Feature extraction using non-linear transformation for robust speech recognition on the Aurora database," Proc. ICASSP, Istanbul, II-1117-1120, June 2000.

[2] A. Janin, D. Ellis and N. Morgan, "Multi-stream speech recognition: Ready for prime time?" Proc. Eurospeech-99, II-591-594, Budapest, September 1999.

[3] K. Kirchoff and J. Bilmes, "Dynamic classifier combination in hybrid speech recognition systems using utterance-level confidence values," Proc. ICASSP, Phoenix, II-693-696, May 1999.

[4] J. Billa, T. Colhurst, A. El-Jaroudi, R. Iyer, K. Ma, S. Matsoukas, C. Quillen, F. Richardson, M. Siu, G. Zavaliagkos, H. Gish, "Recent Experiments in Large Vocabulary Conversational Speech Recognition," Proc. ICASSP, Phoenix, May 1999.

[5] S. Wu, B. Kingsbury, N. Morgan and S. Greenberg, "Incorporating Information from Syllable-length Time Scales into Automatic Speech Recognition," Proc. ICASSP, Seattle, 721-724, 1998.

[6] D. Ellis, "Stream combination before and/or after the acoustic model," ICSI Technical Report TR-00-007 http://www.icsi.berkeley.edu/techreports .

[7] D. Pearce, *Aurora Project: Experimental framework for the performance Evaluation of distributed speech recognition front-ends*, ETSI working paper, September 1998.

[8] B. Kingsbury, *Perceptually-inspired Signal Processing Strategies for Robust Speech Recognition in Reverberant Environments,* Ph.D. dissertation, Dept. of EECS, University of California, Berkeley, 1998.

[9] N. Morgan and H. Bourlard, "Continuous Speech Recognition: An Introduction to the Hybrid HMM/Connectionist Approach." *Signal Processing Magazine*, 25-42, May 1995.

[10] J. Bilmes, "Maximum mutual information based reduction strategies for cross-correlation based joint distributional modeling," Proc. ICASSP, Seattle, 469-472, April 1998.

[11] J. Bilmes, *Natural Statistical Models for Automatic Speech Recognition,* Ph.D. Dissertation, Dept. of EECS, University of California, Berkeley, May 1999.