ON THE USE OF IDEAL BINARY MASKS FOR IMPROVING PHONETIC CLASSIFICATION

Arun Narayanan and DeLiang Wang

Department of Computer Science and Engineering
& Center for Cognitive Science
The Ohio State University
Columbus, OH 43210-1277, USA
{narayaar, dwang}@cse.ohio-state.edu

ABSTRACT

Ideal binary masks are binary patterns that encode the masking characteristics of speech in noise. Recent evidence in speech perception suggests that such binary patterns provide sufficient information for human speech recognition. Motivated by these findings, we propose to use ideal binary masks to improve phonetic modeling. We show that by combining the outputs of classifiers trained on the traditional MFCC features and this novel speech pattern, statistically significant improvements over the baseline MFCC based classifier can be achieved for the task of phonetic classification. Using the combined classifiers, we achieve an error rate of 19.5% on the TIMIT phonetic classification task using multilayer perceptrons as the underlying classifier.

Index Terms — Speech recognition, CASA, ideal binary mask, phone classification, TIMIT.

1. INTRODUCTION

Acoustic modeling forms a major component of speech recognition systems. For a typical speech recognition system continuous speech is labeled by an acoustic model to produce a phone sequence which can later be decoded using pronunciation and language models [1]. Phone classification is an instructive subtask, similar to phone sequence labeling, where the phone boundaries are assumed to be known before the underlying acoustic model performs classification. To deal with continuous speech for which phone boundaries are not known, a phonetic classifier must be coupled with a system that provides the phone boundaries through segmentation.

Considerable efforts have been put on different aspects of phonetic classification, such as features, kind and structure of the underlying classifier or the model, training strategies, etc. Gaussian mixture modeling (GMM) has been the most popular strategy for modeling the underlying classifier. The parameters are typically estimated via maximum likelihood (ML) estimation [1]. Several discriminative training strategies have also been suggested, including large margin training [2, 3] and maximum mutual information training [4]. Apart from GMMs, other strategies used for classification include support vector machines [5], nearest neighbor strategies [6], hidden conditional random fields [7], linear regularized least squares [8] and neural networks [9]. In [3], GMMs are used in a hierarchical structure to yield state-of-the-art results for this task.

This paper primarily focuses on the aspect of feature selection for acoustic modeling. Cepstral features have predominantly been used for the task of acoustic modeling. More specifically, Mel frequency cepstral features (MFCC), along with their delta and acceleration components, have been widely used [2, 7]. In [3, 10] a set of 8 types of cepstral features are extracted and the outputs of the individual classifiers trained on these features separately are aggregated to make the final classification. The use of multiple features significantly improves the classification performance. One of the main disadvantages of cepstral features is that their performance gets severely affected under noisy conditions. This is one of the reasons why MFCC based speech recognizers perform poorly when compared to humans under noisy conditions.

The noise robustness of human listeners is attributed to auditory scene analysis by Bregman [11]. Computational auditory scene analysis (CASA) tries to make use of perceptual cues to create noise robust systems [12]. Ideal binary mask (IBM) has been suggested as one of the important goals of CASA based systems [13]. A recent study in speech perception shows that the pattern of an IBM appears to provide sufficient information for human speech recognition [14]. In the study, IBMs are used to modulate speech shaped noise (SSN). Human subjects listen to IBM-gated noise and, despite a dramatic reduction of speech information, are able to recognize speech almost perfectly. The study suggests that the IBM encodes sufficient phonetic information for humans to perform speech recognition. Motivated by these findings, we explore the IBM for phonetic classification.

Previous work showed that IBMs can be used for isolated digit recognition in noise [15, 16]. In this paper we study IBMs to improve phone classification. Our goal is to use IBMs to augment traditional speech features to improve performance of acoustic models in clean and noisy conditions. In this initial study, we investigate clean conditions to understand how the use of IBMs can affect the performance of phonetic classifiers.

The rest of the paper is organized as follows. Section 2 provides the system description. Experimental results are shown in Section 3. We conclude with a discussion in Section 4.

2. SYSTEM DESCRIPTION

2.1. Ideal binary masks

IBM is a time-frequency (T-F) mask, which is a 2D matrix of binary values that encodes the masking information of speech in

Fig. 1. IBMs of phones /aa/, /ae/, /eh/, /h#/ (sil), /iy/, /ow/ and /s/, ordered from left to right. In the figure, a white pixel indicates 1 and a black pixel 0. Note how the IBM captures the high energy regions of the phone (none for silence).

noise. An entry in the matrix assumes the value 1 if the corresponding T-F unit has a signal to noise ratio (SNR) that exceeds a threshold, also known as the local SNR criterion (LC). Mathematically, an IBM can be defined as

$$IBM(t,f) = \begin{cases} 1 & if\ SNR(t,f) > LC \\ 0 & otherwise \end{cases} \quad (1)$$

where *SNR(t,f)* denotes the SNR within the unit of time *t* and frequency *f*, measured in decibels.

As can be observed from Equation (1), we need the clean speech signal and the noise signal to create an IBM. But our experiments are conducted in clean conditions and hence, we need a 'hypothetical' noise to create an IBM. We create a speech shaped noise (SSN) from our training set for this purpose. SSN is a stationary noise with a long-term spectrum matching that of natural speech. SSN was also used in [14] to test speech intelligibility of IBM-gated noise.

To create the IBM, the clean speech signal and the noise, scaled to the desired SNR level, are first passed though a 64-channel gammatone filterbank with center frequencies spaced according to the ERB (Equivalent Rectangular Bandwidth) scale. Each filter response is then windowed into time frames using a 20 ms rectangular window and a frame shift of 10 ms, to produce a cochleagram [12]. The IBM is then created by calculating the local SNR within each T-F (Time-Frequency) unit and comparing it with the LC. Figure 1 shows IBMs of some of the phones from the TIMIT corpus [19]. The IBMs in the figure were created for a 3 dB mixture of the phones and SSN with the LC set to 0 dB.

2.2. Features used

We use two segment level features to build our classifiers. The features are based on the IBM and the more traditional Mel frequency cepstral coefficients (MFCC).

To create the segment level IBM-based feature, the IBM is first created for the speech segment at the desired SNR level. It is then divided into 5 parts. The first frame and the last frame remain unchanged to form the first and last part, respectively. The remaining frames are then split into three parts, roughly in a 2:3:2 ratio, and averaged. This yields 5 frames with the dimensionality of 320 (64x5). Log duration information is added to this average feature to create a 321 dimensional input representation for each sample.

To create the segment level MFCC-based feature, 13 MFCC features (including the 0^{th} cepstral coefficient) are extracted from a speech sample along with their delta and acceleration coefficients. A window size of 20 ms and an overlap size of 10 ms were used for the cepstral analysis. A pre-emphasis coefficient of 0.97 was also used as is commonly done. This yields a 39 dimensional feature for each frame (calculated using the HTK toolkit [17]). The average MFCC feature is then created in the exact same way as with the IBMs. Log duration is also added as an additional feature. This creates a 196 (39x5 + 1) dimensional average MFCC feature.

The average segment level features are similar to the ones used in [3], [5], [6], [8] and [10]. It is a common practice to create a fixed-size segment-level feature for the purpose of phone classification.

2.3. Classification strategy

We use multilayer perceptrons (MLP) as the underlying classifier. Separate MLPs are trained using each of the two features described above. Our assumption is that since the two features are very different from each other, the kind of errors made by the two classifiers will also be very different. To get the best of both, we need to combine the outputs of the two classifiers in an appropriate way.

We borrow ideas from [18] to combine the two classifier outputs. A softmax function is applied to the MLP outputs so that they sum up to 1. This allows the MLP outputs to be interpreted as probability measures. If there are M classes, we denote the outputs of each MLP as:

$$P(x \in C_i \mid MLP_k), i = 1,\ldots,M \quad (2)$$

$$such\ that,\ \sum_{i=1}^{M} P(x \in C_i \mid MLP_k) = 1 \quad (3)$$

Here x denotes the sample under consideration; C_i denotes the i^{th} class; and MLP_k the k^{th} MLP.

Given a classifier, we analyze the performance of the classifier and encode the prior knowledge about the classifier by building a confusion matrix. This can be done either using the training set or a small held-out development set. An entry n_{ij} in the confusion matrix denotes the number of times the classifier predicted a sample belonging to C_i as C_j. Let $MLP_k(x)$ denote the final classification decision made by MLP_k, for a random input sample x. Using the confusion matrix, define:

$$P(x \in C_i \mid MLP_k(x) = j) = \frac{n_{ij}}{\sum_{i=1}^{M} n_{ij}} \quad (4)$$

which is the probability that a sample classified as belonging to C_j by the MLP has the correct class C_i. Using these probability estimates, we now define the belief of each classifier for each class as:

$$\begin{aligned} bel_k(x \in C_i) &= \sum_{j=1}^{M} P(x \in C_i, MLP_k(x) = j) \\ &= \sum_{j=1}^{M} P(x \in C_i \mid MLP_k(x) = j) P(x \in C_j \mid MLP_k) \end{aligned} \quad (5)$$

The first term can be obtained from the probability measures estimated from the confusion matrix. The second term is directly obtained from the MLP outputs. Finally, the belief for each class

can be estimated by adding up the beliefs of individual classifiers directly, or by adding up their log beliefs. Instead of weighing each classifier equally during this summation, we can also weigh the beliefs of each classifier based on our confidence on the individual classifiers. Therefore, the final belief can be defined as:

$$bel(x \in C_i) = \sum_k w_k bel_k(x \in C_i) \quad (6)$$

or

$$bel(x \in C_i) = \sum_k w_k \log(bel_k(x \in C_i)) \quad (7)$$

w_k can easily be determined using a held out development set since we have only 2 classifiers to combine. The classification decision is made by looking at the beliefs of each class and choosing the class with the greatest belief.

Note that the proposed combination technique is a special case of Bayesian evidence combination when the weighted belief assignments of individual classifiers are added to estimate the final belief in a class C_i. w_k can be thought of as the prior probability of the classifier k in this setting. On the other hand, when the weighted *log* beliefs are combined, it's a special case of Dempster-Shafer (DS) theory of evidence combination if we assume $(bel_k(x \in C_i))^{w_k}$ to be the belief of classifier k on C_i with every other subset of the set of hypothesis (phone classes in our case, also called the *frame of discernment*, Ω, in DS theory) being assigned a belief value 0 [18]. Log being a monotonic function, the DS theory of combining these beliefs would yield the same classification results as ours although the final assigned beliefs of each class will be different because of an additional normalization constant. Note that the normalization does not affect the classification results.

3. RESULTS

3.1. Experimental setup

We perform phone classification experiments using the well benchmarked TIMIT database [19]. As in standard practices, 61 phonetic labels were mapped to 48 phone classes. Glottal stops (/q/) were ignored. MLPs were then trained to perform a 48-class classification. The training set, the development set and the test set consisted of 3696, 400 and 192 utterances, respectively (see also [2, 3, 7, 8, 10]). This corresponds to 140225 tokens in the training set, 15057 tokens in the development set and 7215 tokens in the test set. Results were evaluated by mapping these 48 phone classes to 39 clusters as done in [20].

To generate the IBMs for creating the IBM-based feature, the SNR was set to 3 dB and the LC to 0 dB. This SNR was found to produce more discriminative IBMs among the three tested SNR conditions (0 dB, 3 dB and 6 dB). While generating the IBMs and the MFCC features, 30 ms of speech before and after the segment boundaries was also included [8].

The development set was used to tune all the hyper-parameters of the system, i.e., the SNR at which the IBMs were generated, the number of hidden units in the MLPs, and the optimal weights to combine the two classifier outputs (w_k in Equations (6) and (7)). The development set was also used for early stopping while training the MLPs. The features were standardized (zero mean, unit variance) using the means and variances calculated from the training set. All MLPs were trained using the ICSI Quicknet software package [21]. The final chosen classifier had 1750 hidden units for the MFCC-based classifier and 2500 hidden units for the IBM-based classifier. The optimal weights are shown in Table 2, along with the classification results.

3.2. Experimental results

Table 1 summarizes the baseline results, measured in terms of error rates. The MFCC based classifier performs significantly better than the IBM based classifier. We note that the performance obtained on the core test set by using just the MFCC based features (an error rate of 20.8%) is comparable to the results obtained for this task using other single-classifier strategies like hidden conditional random fields (20.8% and 21.3% in [7]), large margin training of Gaussian mixture models (21.1% in [2]) and linear regularized least squares method (20.9% in [8]). Although the classifier trained on the IBM based features does not perform that well, an error rate of 28.9% by merely using binary features, we believe, is an interesting result. As shown in Table 1, the error rates obtained on the development set are similar.

Table 1. Error rates of the baseline classifiers on the TIMIT development set and the core test set.

Feature used	Development Set	Core Test Set
IBM-based	27.2%	28.9%
MFCC-based	19.4%	20.8%

Table 2. Error rates when the baseline classifiers are combined by adding their beliefs or log-beliefs. The 2nd and 3rd columns specify the weights assigned to each of the baseline classifiers.

What to combine	w_{MFCC}	w_{IBM}	Classification Results	
			Development Set	Core Test Set
Beliefs	0.706	0.294	18.5%	19.8%
log-beliefs	0.709	0.291	**18.3%**	**19.5%**

To see whether the output produced by the IBM based feature can potentially contribute to classification, we calculated the error rate on the development set when at least one of the two classifiers correctly predicted the output. We observed that, at least one of two classifiers predicted correctly with an accuracy of 86.5%, which corresponds to an error rate of 13.5%. This is clearly a significant improvement over the baseline results on the development set, although not achievable unless we know the ground truth labels. More importantly, this observation confirms our assumption that the errors made by the two classifiers are significantly different and hence combining them is a sensible strategy.

Table 2 summarizes the results when the outputs of the two classifiers were combined as described in the previous section. If we directly use the beliefs of the two classifiers (Equation (6)), we obtain an absolute improvement of around 1% over the baseline on the core test set. On using the log beliefs (Equation (7)), an absolute improvement of 1.3% is obtained over the baseline results. This improvement is statistically significant at 5%. Upon running the test to see if at least one of the classifiers was able to correctly classify an input from the core test set, we obtained an error rate of 14.5%. This again shows that the errors made by the

two classifiers are different and useful for improving the overall accuracy for the task.

In [22], MFCC features were combined with features extracted using a convolutional deep belief network for phone classification. It is worth noting that the proposed IBM based feature and the combination strategy perform better than the results reported in [22]. The state-of-the-art result for the task of phone classification is an error rate of 16.7% obtained using a committee based classifier with 8 different features and a hierarchical classification model [3]. We believe the use of the IBM-based feature in such a framework could further reduce the error rates for this task.

4. DISCUSSIONS

We have demonstrated how features derived from ideal binary masks can be used to improve phone classification. The feature is very different from the traditionally used speech features. We obtained an error rate of 19.5% on the core test set which compares favorably to most results reported in recent phone classification literature.

This study shows that IBMs can be effectively used to improve phone classification in clean conditions. We believe that the true strength of the proposed approach lies in its potential applications in noisy conditions. IBM estimation is an active area and several strategies exist to estimate IBM in noisy conditions making it a more robust feature as compared to MFCC [12]. Therefore, when noise corrupts a speech signal, one could adjust the classifier weights based on the SNR of the mixture, to give greater weight to the IBM based classifier as noise corrupts the MFCC based feature. This will be examined in our future research. We would also like to study how the knowledge obtained through phone classification experiments can be extended to perform phone recognition when segment boundaries are not known in advance. This will be important to build an acoustic model with direct applications to speech recognition.

Noise robustness is an important challenge for ASR research. We believe that novel features like the IBM may offer promising new avenues for building noise robust automatic speech recognition systems.

Acknowledgements. The research described in this paper was supported in part by AFRL as a subcontractor to RADC Inc. (FA8750-09-C-0067) and an AFOSR grant (FA9550-08-1-0155).

5. REFERENCES

[1] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A guide to theory, algorithms and system development*, Prentice Hall PTR, New Jersey, 2001.

[2] F. Sha, and L.K. Saul, "Large margin Gaussian mixture modeling for phonetic classification and recognition," *Proc. ICASSP*, pp. 265-268, 2006.

[3] H.A Chang and J. Glass, "Hierarchical large margin Gaussian mixture models for phonetic classification," *Proc. IEEE Workshop on ASRU*, pp. 272-277, 2007.

[4] S. Kapadia, V. Valtchev, and S.J. Young, "MMI training for continuous phoneme recognition on the TIMIT database," *Proc. ICASSP*, pp. 491-494, 1993.

[5] P. Clarkson, and P.J. Moreno, "On the use of support vector machines for phonetic classification," *Proc. ICASSP*, pp. 585-588, 1999.

[6] L. Golipour, *Context-independent monophone recognition using phoneme segmentation and a nonparametric classification approach*, Ph.D. thesis, University of Quebec, 2010.

[7] D. Yu, L. Deng, and A. Acero, "Hidden conditional random field with distribution constraints for phone classification," *Proc. INTERSPEECH*, pp. 676-679, 2009.

[8] R. Rifkin, K. Schutte, M. Saad, J Bouvrie, and J. Glass, "Noise robust phonetic classification with linear regularized least squares and second-order features," *Proc. ICASSP*, pp. 881-884, 2007.

[9] S. Zahorian, P. Silsbee, and X. Wang, "Phone classification with segmental features and binary-pair partitioned neural network classifier," *Proc. ICASSP*, pp. 1011-1014, 1997.

[10] A Hilberstadt, and J. Glass, "Heterogeneous measurements and multiple classifiers for speech recognition," *Proc. ICSLP*, pp. 995-998, 1998.

[11] A.S. Bregman, *Auditory Scene Analysis,* MIT Press, Cambridge, MA, 1990.

[12] D.L. Wang, G.J. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms and Applications,* Wiley/IEEE Press, Hoboken, NJ, 2006.

[13] D.L. Wang, "On ideal binary masks as the computational goal of auditory scene analysis," In P. Divenyi (Ed.), *Speech Separation by Humans and Machines,* Kluwer Academic, Boston, MA, 2005.

[14] D.L. Wang, U. Kjems, M.S. Pedersen, J.B. Boldt, and T. Lunner, "Speech perception of noise with binary gains," *J. Acoust. Soc. Amer.*, vol. 124, pp. 2303–2307, 2008.

[15] A. Narayanan and D.L. Wang, "Robust speech recognition from binary masks," *J. Acoust. Soc. Amer.*, vol. 128, pp. EL217-EL222, 2010.

[16] S.G. Karadogan, J. Larsen, M.S. Pedersen, and J.B. Boldt, "Robust isolated speech recognition using binary masks," *Proc. EUSIPCO*, pp. 1988-1992, 2010.

[17] S. Young, D. Kershaw, J. Odell, V. Valtchev, and P. Woodland*, The HTK Book (for HTK Version 3.4)*, Microsoft Corp., Redmond, WA, 2009.

[18] M. Cheriet, N. Kharma, C.-L. Liu, and C. Suen, *Character Recognition Systems: A guide for students and practictioners*, Wiley-Interscience, Hoboken, NJ, 2007.

[19] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, and N.L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus cdrom," *NIST*, 1993, CD-ROM.

[20] K.F. Lee, and H.W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE T-ASSP*, vol. 37, pp. 1641-1648, 1988.

[21] D. Johnson, "ICSI Quicknet Software Package," 2004 [Online]. Available: http://www.icsi.berkeley.edu/Speech/qn.html.

[22] H. Lee, Y. Largman, P. Pham, and A.Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in NIPS* 22, pp. 1096-1104, 2009.