Bottleneck Features based on Gammatone Frequency Cepstral Coefficients

Jun Qi[1], *Dong Wang*[2,3,4], *Ji Xu*[1], *Javier Tejedor*[5]

[1] Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China
[2] Center for Speech and Language Technologies, Division of Technical Innovation
and Development, Tsinghua National for Information Science and Technology
[3] Center for Speech and Language Technologies, Research Institute of Information technology, Tsinghua
[4] Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
[5] Human Computer Technology Laboratory, Universidad Autónoma de Madrid, Spain

qij08@mails.tsinghua.edu.cn, wangdong99@mails.tsinghua.edu.cn
xuji010@gmail.com, javier.tejedor@uam.es

10.21437/Interspeech.2013-435

Abstract

Recent work demonstrates impressive success of the bottleneck (BN) feature in speech recognition, particularly with deep networks plus appropriate pre-training. A widely admitted advantage associated with the BN feature is that the network structure can learn multiple environmental conditions with abundant training data. For tasks with limited training data, however, this multi-condition training is unavailable, and so the networks tend to be over-fitted and sensitive to acoustic condition changes. A possible solution is to base the BN features on a channel-robust primary feature.

In this paper, we propose to derive the BN feature based on Gammatone frequency cepstral coefficients (GFCCs). The GFCC feature has shown nice robustness against acoustic change, due to its capability of simulating the auditory system of humans. The idea is to integrate the advantage of the GFCC feature in acoustic robustness and the advantage of the BN feature in signal representation, so that the BN feature can be improved in the condition of mismatched training/test channels. This is particularly useful for small-scale tasks for which the training data are often limited. The experiments are conducted on the WSJCAM0 database, where the test utterances are mixed with noises at various SNR levels to simulate the channel change. The results confirm that the GFCC-based BN feature is much more robust than the BN features based on the MFCC and the PLP. Furthermore, the primary GFCC feature and the GFCC-based BN feature can be concatenated, leading to a more robust combined feature which provides considerable performance gains in all the tested noise conditions.

Index Terms: Gammatone filters, bottleneck feature, robust speech recognition

1. Introduction

Acoustic features are essentially important for automatic speech recognition (ASR). The bottleneck (BN) feature has obtained great success, partially due to its capability of learning latent patterns of speech signals in a simple way. The early implementations use a standard multi-layer perceptron (MLP) where the input is a certain kind of primary feature such as Mel frequency cepstral coefficients (MFCCs) or perceptual linear predictives (PLPs), and the targets are phoneme classes. Once appropriately trained, the outputs of the MLP correspond to the posterior probabilities of the input feature over the phoneme classes. These posteriors have shown to be superior or at least complementary to the primary features, mainly attributed to their better discriminative power and stronger capability in capturing long-temporal dynamics [1, 2]. The BN feature is an extension to the MLP-based posterior feature, by involving two important changes: first, the 3-layer MLP is extended to a deep neural network (DNN) [3] by involving multiple hidden layers; second, the features are derived from a ‘bottleneck’ layer instead of the output layer, so the BN feature is based on some latent patterns rather than the phoneme classes. A recent study has demonstrated significant potential of the BN feature on ASR [4].

An advantage of the BN feature is that the deep network can learn multi-conditional acoustic environments with abundant training data. For small-scale tasks with limited training data, however, the network may be highly over-fitted and sensitive to acoustic condition changes in testing. For example, in a recognition task where the models are trained with clean speech but are tested with noise-corrupted speech, substantial performance degradation is often observed (see Section 4). This weakness on acoustic mismatch can be attributed to the large number of parameters and the complex no-linearity of the DNN structure. These complexities may cause great difficulties on model training if the training data are limited, as in the case of small-scale tasks. A possible solution is to derive the BN feature based on a primary feature that is robust against acoustic mismatch, such as Gammatone frequency cepstral coefficients (GFCCs).

The GFCC feature is an auditory feature based on a set of Gammatone filters which simulate the frequency response of human ears. For ASR, the GFCC was firstly proposed in [5] to handle noisy speech. Our previous work [6] enhances the implementation by a full time-domain design and some numerical smoothing techniques. A comprehensive study has shown that the GFCC feature with our implementation leads to significant performance improvement in various noise conditions when compared to the MFCC and the PLP. We thus believe it can help alleviate the weakness of the BN feature on mismatched acoustic conditions.

In this work, we combine the advantage of the GFCC in acoustic robustness and the advantage of the BN feature in signal representation, so that the BN feature can be enhanced in the condition of mismatched training/test channels. This is particularly important for small-scale tasks where the training data are usually limited. Specifically, we treat the GFCC as the primary feature, and derive the BN feature from a DNN with the GFCC as its input. This GFCC-based BN feature therefore can inherit the merit of the GFCC in acoustic robustness. We conduct our experiments on the WSJCAM0 database, and mix the test utterances with noises at different signal to noise ratios (SNR) to simulate the acoustic mismatch.

The rest of the paper is organized as follows: the time-domain implementation of the GFCC is reviewed in Section 2, and the GFCC-based BN feature is presented in Section 3. The experiments are reported in Section 4, followed by some conclusions in Section 5.

2. GFCC implementation

2.1. Gammatone filters

The GFCC feature is based on Gammatone filters, which are designed to simulate the auditory process of human ears [5]. A Gammatone filter with the center frequency at f_c is formulated as follows:

$$g(t) = at^{n-1}e^{-2\pi bt}cos(2\pi f_c t + \phi) \qquad (1)$$

where f_c is the central frequency, and ϕ is the phase which is usually set to be 0. The constant a controls the gain, and n is the order of the filter which is usually set to be equal or less than 4. Finally, b is a decay factor which is related to f_c and is given by:

$$b = 1.019 * 24.7 * (4.37 * f_c/1000 + 1). \qquad (2)$$

A set of Gammatone filters with different f_c forms a Gammatone filterbank, which can be applied to obtain the signal characteristics at various frequencies, resulting in a temporal-frequency representation similar to the FFT-based spectrogram. Each Gammatone filter in the filterbank is referred to as a channel. In order to simulate the human auditory behavior, the central frequencies of the channels are often equally distributed on the Bark scale [7].

An impulse invariant transformation can simplify the design. As illustrated in Figure 1, for the channel whose central frequency is f_c, the input signal $x(t)$ is first compensated by a frequency-dependent component $e^{-j2\pi f_c t}$, and then passes a frequency-independent base filter $\hat{G}(z)$. The output of the channel, denoted by $y(t; f_c)$, is finally obtained from the output of $\hat{G}(z)$ followed by a reverse compensation $e^{j2\pi f_c t}$.

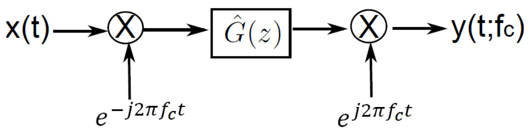


Figure 1: *Time domain Gammatone filtering.*

Considering the special case of $n = 4$, the base filter $\hat{G}(z)$ takes the following form:

$$\hat{G}(z) = \frac{3a}{1 - 4mz^{-1} + 6m^2z^{-2} - 4m^3z^{-3} + m^4z^{-4}} \qquad (3)$$

where f_s is the sampling frequency and $m = e^{-2\pi b/f_s}$. Note that (3) indicates that the channel $y(t; f_c)$ in Figure 1 can be derived from the input signal $x(t)$ by employing a series of time-domain filters which only involve multiplications and summations. This is fundamentally different from the commonly adopted implementation which realizes the filters in the frequency domain and involves complex computation such as FFT. Our previous work has shown that this time-domain implementation is highly efficient [6].

2.2. GFCC implementation

With the Gammatone filterbank designed as described above, a frequency-time representation of the original signal, which is often referred to as a Cochleagram, can be obtained from the outputs of the filterbank. It is then straightforward to compute GFCC features from the Cochleagram [8]. We find that some revisions of the implementation lead to better numerical properties. The remaining of this section presents the details of our GFCC implementation. [1]

2.2.1. *Pre-emphasis*

It is well known that a pre-emphasis is helpful for reducing the dynamic range of the spectrum and intensifying the frequency components that involve most of the information of speech signals. Following the same idea, we implement the pre-emphasis in the GFCC as a 2-order filter given by:

$$H(z) = 1 + 4e^{-2\pi b/f_s}z^{-1} + e^{-2\pi b/f_s}z^{-2}$$

where b is defined in (2) and f_s is the sampling frequency.

2.2.2. *Average-based framing*

To derive the Cochleagram, we employ a window covering K points and shifting every L points to frame $y(t; f_c(m))$, where $f_c(m)$ is the central frequency of the m-th filter. The Cochleagram representation of the n-th frame of the signal is then computed as follows: first of all, average $y(t; f_c(m))$ within the window $t \in [nL, nL + K)$:

$$\bar{y}(n; m) = \frac{1}{K}\sum_{i=0}^{K-1}\gamma(f_c(m))|y(nL + i; f_c(m))|$$

where $\gamma(f_c(m))$ is a center frequency-dependent factor, and $|\cdot|$ represents the magnitude of a complex number; then, aggregate $\bar{y}(n; m)$ of all channels:

$$\bar{y}(n) = [\bar{y}(n; 0), \bar{y}(n; 1), ..., \bar{y}(n; M - 1)]^T$$

where M is the number of channels in the filterbank. In our experiment, we choose $K = 400$, $L = 160$, and $M = 32$ for 16 kHz speech signals, which result in 100 frames per second. The resulting matrix $\bar{y}(n; m)$ forms a Cochleagram.

2.2.3. *Log-based cosine transform*

Based on the Cochleagrams, the discrete cosine transform (DCT) is employed to obtain component-uncorrelated cepstral coefficients. While the DCT can be applied on the Cochleagram directly (as in [8]), we place a logarithm beforehand. The following equation presents the final cepstral coefficients:

$$g(n; u) = (\frac{2}{M})^{0.5}\sum_{i=0}^{M-1}\{\frac{1}{3}log(\bar{y}(n; i))cos[\frac{\pi u}{2M}(2i - 1)]\}$$

where u ranges from 0 to 31. Based on the observation that most values of $g(\cdot; u)$ are close to zero for $u \geq 13$, we choose the first 12 components of $g(n; u)$, which results in the 12-dimensional static GFCC feature:

[1] The code is publicly available at http://homepages.inf.ed.ac.uk/v1dwang2/public/tools/index.html

$$g(n) = [g(n;0), g(n;1), ..., g(n;11)]^T.$$

Considering that dynamic features generally help capture temporal information, we augment the static GFCCs with the first and second order derivatives, resulting in the 36-dimensional GFCC feature.

3. GFCC-based bottleneck features

We follow the architecture illustrated in Figure 2 to derive the BN feature. In this architecture, a DNN consisting of an input layer, three hidden layers and an output layer is constructed to represent the mapping from the primary feature to the phone-states. A particular design of this DNN structure is that the second hidden layer (i.e., the bottleneck layer) involves only 36 units, which are much less than the first and the third hidden layers (each involves 1024 units). By this configuration, the bottleneck layer can learn some latent patterns, which on the one hand represent the input speech signal and on the other hand discriminate the phone states. In prediction, the activations of the units at the bottleneck layer provide a full representation of the input frame based on the latent patterns, and therefore can be used as an efficient feature for ASR. This is the well-known BN feature. Note that the BN feature can be derived based on any primary feature, including the GFCC.

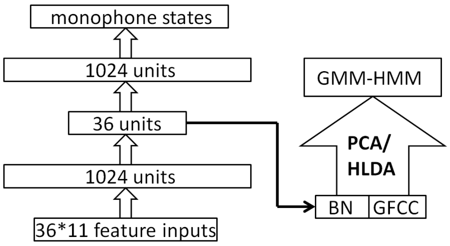


Figure 2: The BN feature and the combined feature based on the GFCC.

In our implementation, a window of 11 consecutive neighboring primary feature vectors centered at the investigated frame are concatenated and used as the DNN input. This concatenated feature is supposed to catch long temporal patterns. To ease the DNN training, Gaussian normalization is applied before the feature is fed into the DNN. The output layer is composed of 121 context-independent phone states.

Training the DNN requires appropriate initialization. It has been known that random initialization is usually unsatisfactory [9]. We follow the approach in [10] where an unsupervised pre-training based on the restricted Boltzmann machine (RBM) is conducted to initialize the DNN structure, and then a supervised fine-tuning based on the conventional back-propagation [11] is invoked to improve discriminative capability of the network. At the pre-training stage, the learning rate is set to 0.001 at the first iteration, and 0.01 for the rest of iterations. The data size is set to 256 and the momentum rate is set to 0.5. At the fine-tuning stage, the learning rate is set in an auto-regularized manner: it starts from 4.0 at the first several iterations, and then is decreased by a factor of 2 at each iteration. The maximum iteration is set to 20, and the convergence is tested on a subset of the training data.

Finally, the BN feature can be combined with the primary feature. This combination usually leads to better performance at least on clean speech [1, 2]. In our study, the principal component analysis (PCA) is employed to reduce the feature dimension and remove the inter-dimension correlation, which are as shown in Figure 2.

	WER%		
	MFCC	PLP	GFCC
Primary	11.48	11.39	10.03
BN	10.02	9.92	8.80
Primary + BN	9.79	9.60	8.36

Table 1: Baseline results on the clean speech data.

4. Experiments

4.1. Data profile and experimental setup

Our experiments are conducted on the WSJCAM0 database. The training data (dataset si_tr) consist of 17 hours of speech signals or 7880 utterances in total. The test data (dataset si_dt5a) are composed of 0.79 hours of speech signals or 368 utterances. All the data are in the reading style and are recorded in a noise-free environment. Performance is evaluated on a recognition task involving 5000 words.

To simulate the acoustic mismatch, a multitude of noise signals from the NOISE-92 database are used to corrupt the *test* data. These noise signals involve three types: white noise, babble noise, and f16 noise. The mixing is conducted at various SNR levels, including 30db, 20db and 15db.

For a fair comparison, all the primary features are derived from the same frequency range (80-5000 Hz) with the same frame rate (100 frames per second). Each feature involves 12 static components plus their first and second order derivatives, which amounts to 36-dimensional feature vectors. Cepstral mean subtraction (CMS) is applied in all the experiments.

The acoustic models are tied tri-phones modeled by 3-state left-to-right hidden Markov models (HMMs). The state emission is modeled by the Gaussian mixture model (GMM), where the number of Gaussian components is fixed to 16 and all the Gaussians are diagonal. The language model is a bigram model involving 5k words and is trained with the transcriptions of all the speech data in WSJCAM0.

The HTK toolkit from Cambridge is used to extract the MFCC and PLP features. The same toolkit is used to train the acoustic models and conduct the decoding. The toolkit TNet provided by BUT[2] is used to conduct the DNN training.

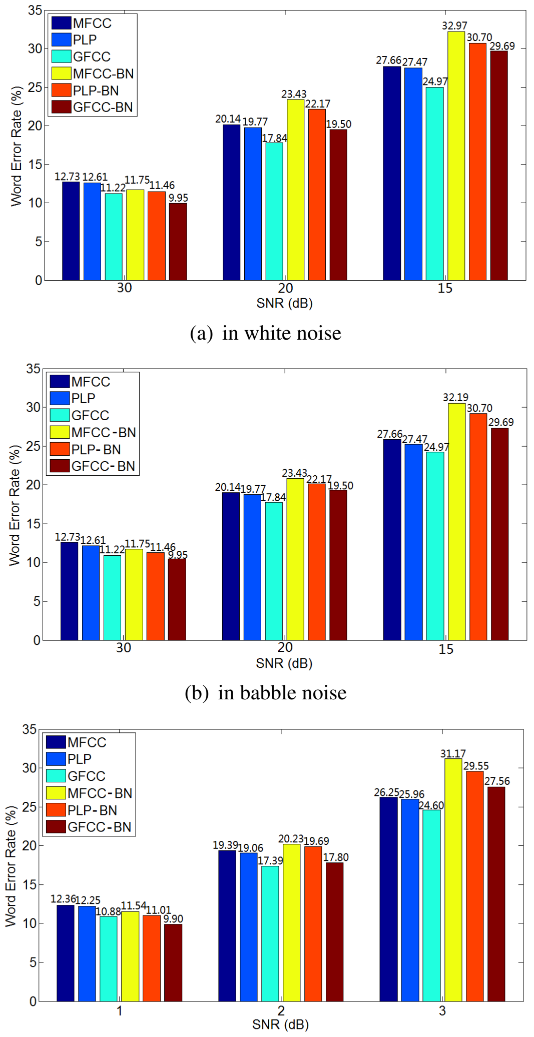
Table 1 presents the baseline results on the clean test data, in terms of the word error rate (WER). We observe that the BN features outperform the primary features in all the cases, and the combined features perform the best. When comparing the MFCC, PLP and GFCC, we see that the GFCC-based features, in spite of the primary feature or the BN feature or the combined feature, outperform their counterparts based on the MFCC and the PLP.

4.2. GFCC-based BN feature

We then investigate the behavior of the BN feature (without combination) in mismatched (noisy) conditions. The three types of noises are mixed with the test utterances at three SNR levels. The WER results are presented in Figure 3. We first observe that, under noisy conditions, the ASR performance is severely degraded in general, no matter which feature is used. Second, the degradation with the BN features is much more significant

[2] http://speech.fit.vutbr.cz/software

than with the primary features, confirming that DNNs are highly sensitive to channel mismatch in small-scale tasks. Comparing MFCC/PLP/GFCC, we see clear advantage of the GFCC feature in noisy conditions: both the GFCC primary feature and the GFCC-based BN feature lead to lower WERs than the M-FCC and PLP counterparts.



(a) in white noise

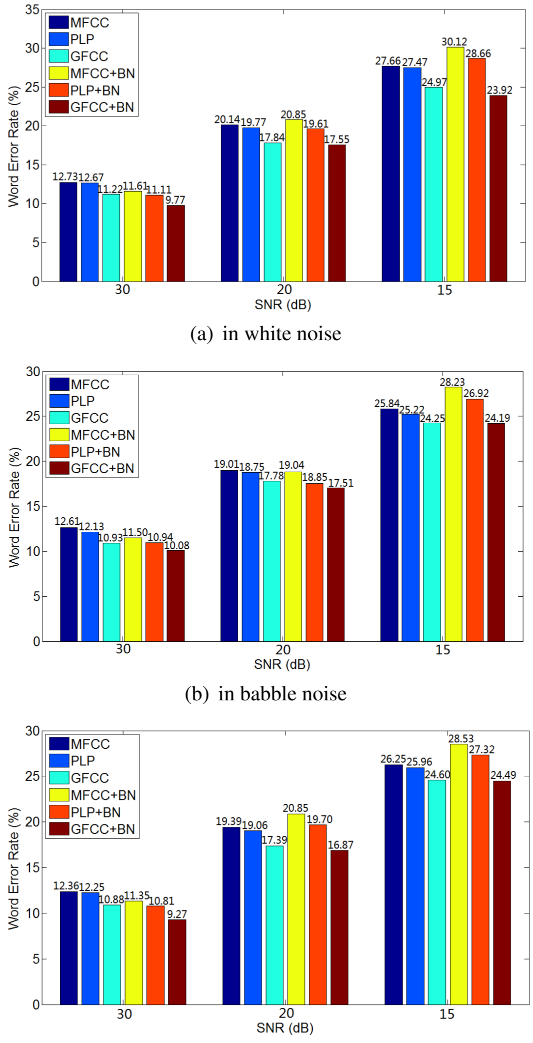
(b) in babble noise

(c) in f16 noise

Figure 3: Performances of the BN features in various noise conditions.

4.3. GFCC-based combined features

The second experiment examines the performance of the combined features (primary + BN). Figure 4 presents the results. We observe that the combination indeed provides some gains if the SNR is high. If the SNR is low (e.g., $\leq$ 20db), the combined features perform generally worse than the corresponding primary features. The only exception is the GFCC-based combined feature, which outperforms the primary feature in all the noise conditions, although relatively marginal when the SNR is low. This result confirms that the robustness of the GFCC can significantly mitigate the inherit weakness of DNNs in channel mismatch, and therefore can help extend the application of BN features to scenarios where the training data are limited and the test condition mismatches the training condition.



(a) in white noise

(b) in babble noise

(c) in f16 noise

Figure 4: Performances of the combined features in various noise conditions.

5. Conclusions

This paper proposed to use the GFCC feature to deal with the weakness of the BN feature in mismatched training/test conditions. Our experiments on the WSJCAM0 database confirmed that the GFCC-based BN feature is much less sensitive to noise interference. In fact, the BN features based on the MFCC and PLP features simply fail if the noise level is high, while the GFCC-based BN feature, if combined with the primary GFC-C, still works well, even though the SNR goes to 15db. Future work will extend the study to other scenarios of channel mismatch beyond noise.

6. Acknowledgements

This work was partially supported by Tencent Ltd. under the contract No. 20132000294. Special thanks to Jenny Chen, Li Lu and Eryu Wang from Tencent Ltd. for the initial discussion and close collaboration.

7. References

[1] H. Hermansky and S. Sharma, “Temporal patterns (TRAPS) in ASR of noisy speech,” in *Proc. ICASSP’99*, 1999, pp. 289–292.

[2] Q. Zhu, Y. Chen, and N. Morgan, “On using MLP features in LVCSR,” in *Proc. Interspeech’04*, 2004, pp. 921–924.

[3] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[4] D. Yu and M. L. Seltzer, “Improved bottleneck features using pretrained deep neural networks,” in *Proc. Interspeech’11*, 2011, pp. 237–240.

[5] Y. Shao, Z. Jin, D. Wang, and S. Srinivasan, “An auditory-based feature for robust speech recognition,” in *Proc. I-CASSP’09*, 2009, pp. 4625–4628.

[6] J. Qi, D. Wang, Y. Jiang, and R. Liu, “Auditory feature based on gammatone filters for robust speech recognition,” in *the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2013.

[7] E. Zwicker, “Subdivision of the audible frequency range into critical bands,” *The Journal of the Acoustical Society of America*, vol. 33, no. 2, 1961.

[8] X. Zhao, Y. Shao, and D. Wang, “Casa-based robust speaker identification,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 5, pp. 1608–1616, 2012.

[9] J. Frankel, D. Wang, and S. King, “Growing bottleneck features for tandem ASR,” in *Proc. Interspeech’08*, 2008, p. 1549.

[10] A. Mohamed, G. E. Dahl, and G. E. Hinton, “Acoustic modeling using deep belief networks,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 504–507, 2012.

[11] M. Christopher, *Pattern Recognition and Machine Learning*. New York, Inc. Secaucus, NJ, USA: Springer, 2007.