

Frequency Offset Correction in Single Sideband(SSB) Speech by Deep Neural Network for Speaker Verification

Hua Xing, Gang Liu, John H.L. Hansen

Center for Robust Speech Systems (CRSS)
University of Texas at Dallas, Richardson, USA

hua.xing@utdallas.edu, gang.liu@utdallas.edu, john.hansen@utdallas.edu

Abstract

Communication system mismatch represents a major influence for loss in speaker recognition performance. This paper considers a type of nonlinear communication system mismatch- modulation/demodulation (Mod/DeMod) carrier drift in single sideband (SSB) speech signals. We focus on the problem of estimating frequency offset in SSB speech in order to improve speaker verification performance of the drifted speech. Based on a two-step framework from previous work, we propose using a multi-layered neural network architecture, stacked denoising autoencoder (SDA), to determine the unique interval of the offset value in the first step. Experimental results demonstrate that the SDA based system can produce up to a +16.1% relative improvement in frequency offset estimation accuracy. A speaker verification evaluation shows a +65.9% relative improvement in EER when SSB speech signal is compensated with the frequency offset value estimated by the proposed method.

Index Terms: frequency offset, single sideband, speaker verification, denoising autoencoder

1. Introduction

In radio communication, single side-band (SSB) communication is an important and commonly used contemporary communicative approach. The main reason for its popularity lies in the advantages of power saving and narrow bandwidth introduced by the techniques of suppressing or removing the carrier signal and one sideband, while only leaving a single sideband in the transmitted signal. These advantages are very appealing as the radio-frequency spectrum, once thought to be adequate for all needs, is becoming crowded due to increased data/voice traffic requirement in today's wirelessly connected society.

A disadvantage of SSB transmission is that the received signal is easily distorted by a frequency offset introduced by a mismatch between the carrier frequency of the received signal and the carrier frequency used in demodulation. For speech signals, the distortion of frequency shift makes the speech unpleasant, sounding strange and Donald Duck-like to the listener, and results in poor quality and intelligibility of the speech signal [1]. Moreover, frequency offset causes a problem in automatic speech and speaker recognition because it affects features based on spectral structure such as MFCCs and PLPs. Automatically estimating frequency offset in SSB speech, in order to help improve speaker recognition in radio communication data, has been a problem we investigated in current and previous studies [2].

A number of studies have been reported to detect and correct frequency offset in SSB speech [3],[4],[5],[6]. Most approaches are based on the relationship between the estimated

pitch f_0 and the observed peak locations corresponding to the harmonics of f_0 in voiced speech. In [3],[6], both f_0 and the positions of several spectral peaks, $p(n)$, are estimated, and Δf is deduced from the linear relationship $\Delta f = p(n) - n f_0$. In [4],[5], a comb filter with a spectral period equal to f_0 and a moving phase was fitted to the spectrum of voiced speech. Δf was then estimated as the phase of the best fitting comb filter. The ambiguity is obvious from this framework: if Δf is a possible frequency shift, $\Delta f \pm n f_0$ ($n = 1, 2, 3$) are also possible options. [4],[6] overcame this problem by accumulating the maximum value of correlation from frame to frame as a histogram. Δf was then estimated as the position that gives the maximum in this histogram. The method obtained effective results given sufficiently long speech signals. [7] used a probabilistic estimation method to overcome the above limitation.

In [2], we proposed a two-step method to estimate the frequency offset in SSB speech. In the first step, the value of Δf is scaled to a unique interval where board ambiguity can be mostly eliminated by a statistical method; then a fine-tuning is performed within the predetermined unique interval to estimate Δf without uncertainty. The first step which contributed most of performance improvement is critical to the proposed method. A statistical method and an innovative feature, SPSS-MFCC, were proposed to detect the unique interval which frequency offset should belong to; i-Vector and PLDA were used as back-end classifier of the proposed feature following the same strategies in speaker recognition [8],[9],[10].

Deep neural networks (DNN) have become widely used for speech and speaker recognition problems in recent years to improve the recognition performance. Deep Belief Network (DBN) has been successfully used in speech recognition [11]. DBN and Auto Associative Neural Network (AANN) have been investigated as a front-end speaker specific feature extractor for speaker recognition [12],[13]. [14],[15] used DBN as a substitute for GMM UBM to extract statistics needed for Gaussian supervector system and i-Vector system in speaker recognition. Autoencoder, also known as Auto Associative Neural Network (AANN), has been proven to be able to extract robust speaker specific features by unsupervised training aimed at reconstructing the input feature before supervised fine tuning parameters. [16] used DNN to assist speaker recognition under lombard effect. In this study, we propose to implement a multi-layered Autoencoder in the first step of above frequency offset estimation process to detect the unique interval of the frequency offset in speech.

The paper is organized as follows: Section 2 and Section 3 briefly review the two-step framework and feature used in the first step, symmetric partial spectral smoothed MFCC (SPSS-MFCC) respectively. Section 4 discusses in detail the stacked

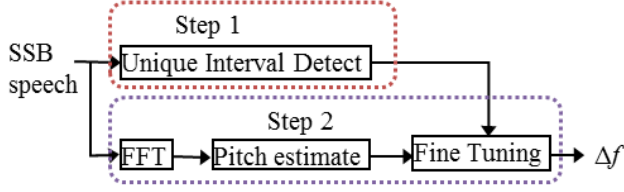


Figure 1: Block diagram of the overall frequency offset estimation

denoising autoencoder model, and the way we implement it in the first step of our framework. Section 5 is devoted to experimental settings and results. For comparison, the back end used in previous study, i-Vector and PLDA, is also briefly reviewed. Section 6 will draw conclusions.

2. Overview of Framework

An offset estimation method was proposed based on the two steps illustrated in Fig 1. In the first step, a unique interval that contains the possible frequency offset without ambiguity caused by the periodic property in frequency is detected, which means that only one value in this interval can be the possible candidate for more-refined estimation results in the second step. The length of interval is set to 50Hz in our study to satisfy above requirement, given that the pitch values of most speakers are between 60Hz and 200Hz. A statistical method was used to obtain this unique interval for each utterance. Once the unique interval is identified, a fine search for the frequency offset within the detected frequency interval is carried out in the second step.

2.1. Unique interval detection

A frequency offset range from -200Hz to 400Hz, where the majority of frequency offsets locate in the experiments, is divided into small bins of 50Hz without overlap, which are called unique intervals here. Each utterance is segmented into 40ms frames with a 20ms overlap. Voice activity detection (VAD) is used to choose the frames having voice information. A feature modified from MFCC has been proposed for each voice active frame representing different frequency offsets. Empirical observation shows that the proposed feature worked successfully in unique interval detection. The next section describes how we modify the MFCC feature to make it more efficient for this task.

After acoustic feature extraction, [2] used two back-end systems, GMM SVM back-end and i-Vector PLDA back-end to decide which unique interval should be labeled for the utterance.

Once the unique interval is detected, the value of the frequency offset can be finely searched within the determined unique interval using (1)-(3) based on Complex correlation:

$$C(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) \exp(j2\pi kn/N) \quad (1)$$

$$f_0 = f_s/T \quad (2)$$

$$\Delta f_r = \left(\frac{f_0}{2\pi} \arctan\left(\frac{\text{Im}C(T)}{\text{Re}C(T)}\right) \right) + rf_0 \quad (3)$$

Where $S(k)$ represents power spectrum on positive frequencies, $C(n)$ is mathematical definition of Complex correlation. T is

pitch period can be estimated as the index that provides the maximum value of the magnitude of complex correlation $C(n)$, after ruling out the first few indexes. Eq.(3) gives a set of possible values of frequency offset Δf for a given frame with different values of r , among which only the value that falls into the pre-determined unique interval is chosen as the correct one. More details of fine tuning process can be found in [4].

3. Symmetric Partial Smoothed Spectral MFCC (SPSS-MFCC)

The overall process to calculate the SPSS-MFCC feature is similar to MFCC except for spectrum smoothing and the filter bank shape. Specifically, after pre-emphasis and segmentation, each speech frame is transformed into the frequency domain by a DFT. Next, the spectrum is smoothed by averaging the spectra of each frame with its adjacent frames. The length of the smoothing window can be varied with frequency. In our study, the window length of 3 is an empirical optimal. Following smoothing, a vector of spectral power representation is calculated by applying the symmetric partial mel-frequency filter to the smoothed spectrum where the width of the filters are symmetric to the middle of the entire frequency range and only the filters in low frequency and high frequency are used. The number of filters in the filter bank was determined empirically. The optimal number is 80 by experience. A discrete cosine transform (DCT) is applied to the logarithm of the power representation. The first 12 DCT coefficients and log of the energy constitute the 13-dimensional SPSS-MFCC coefficients, which is concatenated with its first and second differential $\Delta, \Delta\Delta$ to form a 39-dimensional feature vector for each signal frame. More details of SPSS-MFCC can be found in [2].

4. Deep Denoising Autoencoder

DNN is generally interpreted as a neural network with multiple linear or non-linear hidden layers which aims at representing the data in a form of encoding. Autoencoder is a class of such models that can represent the data in a series of nonlinear transformations [17]. The objective of learning is to minimize the data reconstruction error. Autoencoder is mainly used as a dimensionality reduction tool in the bottle-neck networks, but the flexibility in choosing their structural topology makes them an alternative strategy in the pre-training phase of constructing deep networks [18]. Denoising autoencoder is a strategy to avoid identity learning structure and obtain a more robust representation of the input. This section will first discuss principle of autoencoder and denoising autoencoder, and then a multi-layer structure is formed by stacking layers of the denoising autoencoder as a classifier.

4.1. Autoencoder

Autoencoder is a two-layered neural network with a single non-linear hidden layer. In greedy layer-wise structure as in Fig 2, the representation learned by one autoencoder serves as an input to the next level autoencoder. Specifically, an m dimensional input x is first mapped to a hidden representation y of dimension n :

$$y = f(Wx + b) \quad (4)$$

where f is a non-linear function which is in general a sigmoid function. This mapping process is also called an encoder, after which the latent representation is mapped back into a reconstruction z of the same length as x through a similar transfor-

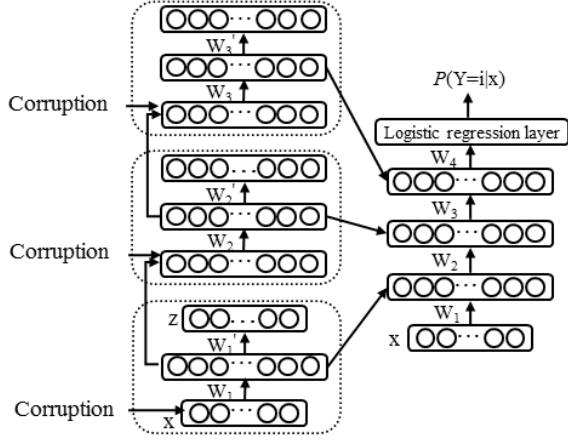


Figure 2: Structure of Stacked Denoising Autoencoder

mation:

$$z = f(W'y + b') \quad (5)$$

z can be viewed as a prediction of x given the code y . The mapping process from y to z is called a decoder. In Eq.5, W' could be any $n \times m$ dimensional matrix, here it is constrained by $W' = W^T$, where W^T indicates the transpose of W . The objective function used here for training the autoencoder network is the mean square error between the actual and reconstructed input:

$$L(x, z) = \frac{1}{|x|} \sum_{i=1}^{|x|} (x_i - z_i)^2 = \|x - z\|_2^2 \quad (6)$$

where $|x|$ indicates the length of x . The parameters of the model (W, b, b') are optimized so that the objective function is minimized iteratively. Stochastic gradient descent is used here to update the model parameters for each iteration.

4.2. Denoising Autoencoder

An Autoencoder can reconstruct the input with arbitrary precision when the dimension of the hidden layer is equal to or greater than the input data dimension. To avoid the system learning an identity function, the input features are artificially corrupted by adding noise at a certain level and the network is trained to construct the clean input. The mismatch between the desired target and actual input compels the network to focus on the statistical structure of the input data in the hidden layer. Data corruption can be carried out in different ways depending upon the numerical structure of the input. Here we follow the process in [17] which randomly sets part of the input to zero. Consequently, x in Eq.4 is replaced by the corrupted version of input and cost function as follows:

$$L(\hat{x}, z) = \frac{1}{|\hat{x}|} \sum_{i=1}^{|\hat{x}|} (\hat{x}_i - z_i)^2 = \|\hat{x} - z\|_2^2 \quad (7)$$

where \hat{x} represents the corrupted input feature and z is the recovery of the clean inputs from the corrupted versions.

4.3. Stacked Denoising Autoencoder (SDA)

The trained denoising autoencoders can be stacked to form a deep network by feeding the latent representation of the denois-

ing auto-encoder calculated on the layer below as input to the current layer (see Fig 2). The layer-wise unsupervised pre-training of such an architecture as described in preceding section is accomplished one layer at a time. Each layer is trained as a denoising auto-encoder by minimizing the reconstruction of its input (which is the output code of the previous layer). After all layers have been pre-trained, the network goes through a second stage of training called fine-tuning. In this stage we carry out traditional supervised training aimed at minimizing the prediction error for the supervised task. First, a logistic regression layer is added on top of the network. Next, the entire network was trained as a multilayer perceptron with the parameters initiated by the pre-training results. At this point, we only consider the encoding parts of each auto-encoder as shown in Fig 2. Class probabilities are obtained by a *softmax* function applied to the output of the last hidden layer as follows:

$$P(Y = i|x) = \text{softmax}_i(Wz + b) = \frac{e^{W_i z + b_i}}{\sum_j e^{W_j z + b_j}} \quad (8)$$

where W and b are trained parameters of logistic regression layer, and z is the output of the last hidden layer.

4.4. SDA for unique interval detect

After a 39 dimensional SPSS-MFCC feature is extracted for each frame, an three hidden layered SDA is trained as preceding description using features derived from the training data. For test, each frame is scored after passing through the trained network. The max of the average scores over the entire utterance results in the overall class label.

5. Experiments and Results

5.1. Frequency Offset Estimation

The DARPA RATS corpus [19] contains voice communications in various languages transmitted over several adverse radio channels, one of which corresponds to SSB transmission demonstrating frequency offset distortion. The transmitted audio streams are recorded in parallel with the original clean source speech and time alignment is provided so that the oracle frequency shift value for each sentence can be traced by cross correlation the spectrogram of the source signal with that transmitted through the SSB channel.

Control experiments using our previous method based on GMM SVM and i-Vector PLDA techniques are also implemented on the same data set. Details of the systems can be found in [2]. We first assess the ability of uniqueness interval detection of SDA back-end systems compared with the other two for actual RATS data. The training data is constructed by adding noised extracted from RATS data by SNR of 20dB, 10dB and 0dB to the original clean resource data, passing them through a channel-like filter and then artificially shifting them by a value of central frequency of each uniqueness interval between -200Hz to +400Hz with a length of 50Hz and no overlapping. 2000 utterances for each shifting case under each SNR condition are used for training. The classification accuracies for three speech durations (2s, 5s and 10s) are summarized in Fig 3. A comparison between three systems shows superior performance for the SDA system over other two systems. Numerically, the averaged accuracies over various durations using three systems are 51.9% (GMM), 73.5%(i-Vector) and 85.6% (SDA).

Performance of the overall system is evaluated in terms of estimation accuracy calculated as the percentage of estimates

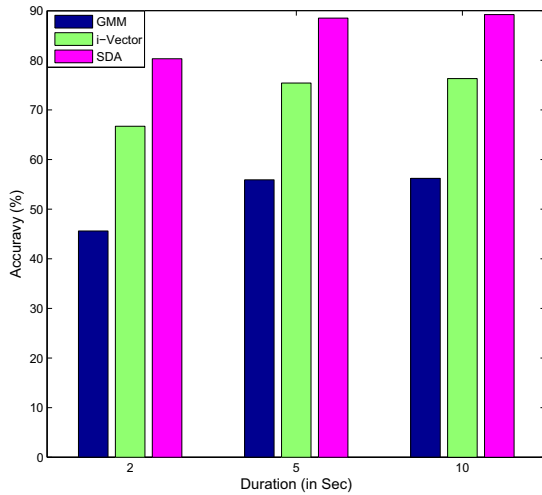


Figure 3: Uniqueness interval classification accuracy (%) using three back end strategies on various speech durations.

that fall within 10Hz of the correct value. Fig 4 demonstrates results of fine-tuning with/without proposed unique interval detection methods using various data lengths. The results show that gross scaling searching range into the uniqueness interval detected by any of the three proposed methods can increase estimation results dramatically. Since SDA performs the best in uniqueness interval detection, it is no surprise that the combined SDA system outperforms the other systems in overall estimation results. The averaged accuracies of overall estimation over 10 durations for systems with three uniqueness interval front-end techniques are 48.4%(GMM), 68.9%(i-Vector), and 80%(SDA), while it is only 27.9% for the system without any uniqueness interval detection.

5.2. Application to Speaker Verification

The proposed frequency shift estimating system is applied to the SSB channel of RATS corpora for speaker verification. The data was produced in an ideal condition that the frequency offset keeps unchanged through each utterance. However, in actual transmissions the frequency offset can drift over time. This is the reason for focusing on estimating frequency offset using a very short duration block of speech. In this study, for each utterance a 5s segment is used to estimate the frequency offset of the entire utterance with the proposed system. The offset is compensated by employing the shift with the estimated value in opposite manner. The compensated voice transmissions are used for speaker verification.

A 39 dimensional MFCC is used as acoustic feature for speaker verification system. A 512-mixture, gender-independent UBM was trained using the training data. The UBM means were used to train a 400-dimensional i-Vector using the development dataset. The resulting i-Vectors were then used to train a PLDA system, producing a 100-dimensional subspace for final scoring.

Speaker verification results are compared under two conditions: (1) training using clean source speech mismatched with distorted/compensated testing data, and (2) training using compensated speech matched with testing data.

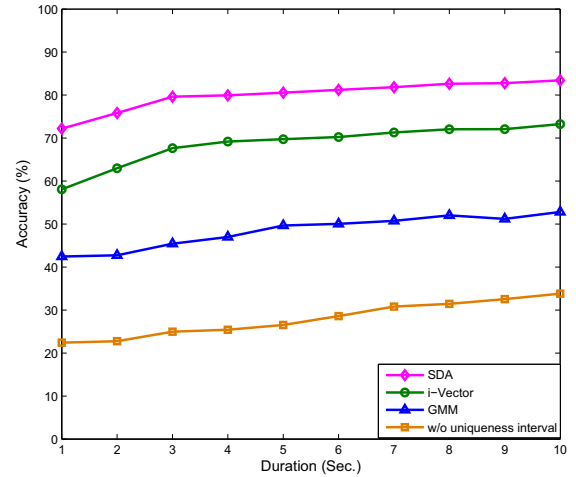


Figure 4: Results of fine-tuning with/without unique interval detection using three back-end systems.

Evaluation is carried out on SSB data (i) without compensation, (ii) compensated using system with SDA and (iii) i-Vector PLDA system respectively. Table 1 shows EER results on these data. Under either matched or mismatched training condition, compensation using values estimated by the proposed SDA system obtains the best speaker verification performance. Numerically, for match condition a +13.8% relative gain is obtained compared to the data without compensation and a +9.1% improvement compared to the data compensated using previous method; for mismatched condition, +32% and +65.9% relative gain is obtained.

	Mismatch	Match
w/o compensation	26.1	5.8
Compensate using i-Vector	13.1	5.5
Compensate using SDA	8.9	5.0

Table 1: EER(%) of speaker verification experiment on SSB speech w/o frequency offset compensation using different methods under matched/mismatched training conditions

6. Conclusion

In this paper, we proposed using a three-layer stacked denoising autoencoder to estimate frequency offsets in a SSB Mod/DeMod communication channel. Working with the two-step framework and SPSS-MFCC feature proposed previously, the new SDA based algorithm improves relative offset estimation performance by +16.1%. When applied to speaker verification, the compensation according to the offset estimated by the SDA system is shown to improve the speaker verification performance in both matched (up to +9.1% relative improvement) and mismatched training conditions (+32% relative improvement) compared to the previous i-Vector system solution.

7. References

- [1] P. F. Assmann, S. Dembling, and T. M. Nearey, "Effects of frequency shifts on perceived naturalness and gender

- information in speech.” in *INTERSPEECH*, 2006.
- [2] H. Xing, P. C. Loizou, and J. H. Hansen, “Frequency offset correction in single sideband speech for speaker verification,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4022–4026.
 - [3] J. Suzuki, T. Shimamura, and H. Yashima, “Estimation of mistuned frequency from received voice signal in suppressed carrier ssb,” in *Global Telecommunications Conference, 1994. GLOBECOM’94. Communications: The Global Bridge., IEEE*, vol. 2. IEEE, 1994, pp. 1045–1049.
 - [4] R. J. Dick, “Co-channel interference separation,” DTIC Document, Tech. Rep., 1980.
 - [5] D. Cole, S. Sridharan, and M. Moody, “Frequency offset correction for hf radio speech reception,” *Industrial Electronics, IEEE Transactions on*, vol. 47, no. 2, pp. 438–443, 2000.
 - [6] S. Ganapathy and J. Pelecanos, “Enhancing frequency shifted speech signals in single side-band communication,” *Signal Processing Letters, IEEE*, vol. 20, no. 12, pp. 1231–1234, 2013.
 - [7] T. Güllow, U. Heute, and H. J. Kolb, “Ssb-carrier mismatch detection from speech characteristics: Extension beyond the range of uniqueness,” in *Proc. EUSIPCO*, 2002.
 - [8] G. Liu and J. H. Hansen, “An investigation into back-end advancements for speaker recognition in multi-session and noisy enrollment scenarios,” *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 1978–1992, 2014.
 - [9] G. Liu, T. Hasan, H. Boril, and J. H. Hansen, “An investigation on back-end for speaker recognition in multi-session enrollment,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7755–7759.
 - [10] G. Liu and J. H. Hansen, “Supra-segmental feature based speaker trait detection,” in *Proc. Odyssey*, 2014.
 - [11] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.
 - [12] V. Vasilakakis, S. Cumani, and P. Laface, “Speaker recognition by means of deep belief networks,” *Proc. Biometric Technologies in Forensic Science*, 2013.
 - [13] Y. S. P. J. and S. R., “Bottleneck features for speaker recognition,” in *Odyssey*, 2012, pp. 105–108.
 - [14] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, “A novel scheme for speaker recognition using a phonetically-aware deep neural network,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.
 - [15] W. Campbell, “Using deep belief networks for vector-based speaker recognition,” in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
 - [16] M. M. Saleem, G. Liu, and J. H. Hansen, “Weighted training for speech under lombard effect for speaker recognition,” *ICASSP 2015*, 2015.
 - [17] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.
 - [18] K. Chen and A. Salman, “Learning speaker-specific characteristics with a deep neural architecture,” *Neural Networks, IEEE Transactions on*, vol. 22, no. 11, pp. 1744–1756, 2011.
 - [19] K. Walker and S. Strassel, “The rats radio traffic collection system,” in *Proc. Odyssey*, 2012.