CNMF-BASED ACOUSTIC FEATURES FOR NOISE-ROBUST ASR

Colin Vaz[1⋆], *Dimitrios Dimitriadis*[2], *Samuel Thomas*[2], *and Shrikanth Narayanan*[1]

[1]Signal Analysis and Interpretation Lab, University of Southern California, Los Angeles, CA 90089
[2]IBM, T. J. Watson Research Laboratory, Yorktown Heights, NY 10598

cvaz@usc.edu, <dbdimitr,sthomas>@us.ibm.com, shri@sipi.usc.edu

ABSTRACT

We present an algorithm using convolutive non-negative matrix factorization (CNMF) to create noise-robust features for automatic speech recognition (ASR). Typically in noise-robust ASR, CNMF is used to remove noise from noisy speech prior to feature extraction. However, we find that denoising introduces distortion and artifacts, which can degrade ASR performance. Instead, we propose using the time-activation matrices from CNMF as acoustic model features. In this paper, we describe how to create speech and noise dictionaries that generate noise-robust time-activation matrices from noisy speech. Using the time-activation matrices created by our proposed algorithm, we achieve a 11.8% relative improvement in the word error rate on the Aurora 4 corpus compared to using log-mel filterbank energies. Furthermore, we attain a 13.8% relative improvement over log-mel filterbank energies when we combine them with our proposed features, indicating that our features contain complementary information to log-mel features.

Index Terms— acoustic features, dictionary learning, feature extraction, non-negative matrix factorization, robust speech recognition

1. INTRODUCTION

Automatic speech recognition (ASR) is increasingly being used as the primary interface between humans and devices. Speech offers a natural and efficient way to communicate with devices. Furthermore, rich information contained in speech, such as emotion [1] and cognitive load [2], can help devices interact or respond appropriately to users. Unfortunately, ASR systems perform poorly in noisy environments. Generally, features extracted from noisy speech contain distortion and artifacts. Researchers have proposed several approaches to reduce the distortion and artifacts, including speech denoising [3], feature enhancement [4], feature transformation [5], and acoustic model adaptation [6, 7]. Multi-condition training has also been found to reduce word error rates on noisy speech [8]. The goal in all of these approaches is to reduce the mismatch between the features extracted from clean and noisy speech.

Speech denoising is a commonly-used pre-processing step. Popular methods for speech denoising include Wiener filtering and spectral subtraction methods [9]. These methods assume the power spectra of speech and noise are additive, and an estimate of the noise power spectra can be subtracted from the noisy power spectra at the frame level. Another denoising technique assuming additive components is non-negative matrix factorization (NMF) [10, 11]. In NMF, each frame of noisy speech is decomposed into components from a speech dictionary and a noise dictionary, and the underlying speech is recovered by keeping the components corresponding to the speech dictionary. Speech denoising, however, can introduce distortion and artifacts, such as musical noise, and has been shown to degrade ASR performance [8, 12]. Moreover, these algorithms operate at each frame independently, and so they can introduce discontinuities across frames. These discontinuities can manifest as noise in the features, thus contributing to feature mismatch between clean and noisy speech.

Another way to reduce feature mismatch is to extract features that are more robust to noise. Moreno et al. introduced Vector Taylor Series (VTS) features [13], which uses the Taylor series expansion of the noisy signal to model the effect of noise and channel characteristics on the speech statistics. Deng et al. proposed the Stereo-based Piecewise Linear Compensation for Environments (SPLICE) algorithm [14] for generating noise-robust features for datasets that have clean versions of the noisy data (stereo datasets). They assume that each cepstral vector from the noisy speech comes from a mixture of Gaussians, and that the clean speech cepstral vector has a piece-wise linear relationship to the noisy speech cepstral vector. Power-Normalized Cepstral Coefficients (PNCC), recently proposed by Kim and Stern [15], were shown to reduce word error rates on noisy speech compared to Mel-Frequency Cepstral Coefficients (MFCC) and Relative Spectral Perceptual Linear Prediction (RASTA-PLP) coefficients. Inspired by human auditory processing, the processing steps for creating PNCCs include a power-law nonlinearity, a denoising algorithm, and temporal masking.

We propose an algorithm for creating noise-robust acoustic features using convolutive NMF (CNMF) [16] without assuming any distribution on the noisy speech. CNMF creates a dictionary that contains spectro-temporal building blocks of a signal and generates a time-activation matrix that describes how to additively combine those building blocks to form the original signal. The time-activation matrix encodes the occurrence and magnitude of each spectro-temporal building block within the speech. Thus, the time-activation matrix can be discriminative of the different phonemes at the frame level when the dictionary remains fixed. In this paper, we will describe how to build dictionaries for speech and noise such that the time-activation matrices are robust to noise.

This paper is organized as follows. Section 2 describes the process we used to create acoustic features that are more invariant to acoustic noise. Section 3 discusses the ASR experiment and compares the word error rate with baseline log-mel features extracted from noisy and denoised speech. Section 4 gives insights into the results of our experiments and points out some of the limitations in our work. Finally, Section 5 offers our conclusions and directions for future work.

2. ALGORITHM FOR CREATING NOISE-ROBUST ACOUSTIC FEATURES

Log-mel filterbank energies are commonly used as a feature for acoustic modeling. First, the mel filterbank energies are calcu-

[⋆]The first author performed this work as an intern at IBM.

lated by taking a $D \times n$ spectrogram X and multiplying it by a $d \times D$ matrix A that contains the mel filterbank in its rows. The resulting $d \times n$ matrix $Y = AX$ contains a representation of the signal in the mel frequency domain, and $\log(Y)$ is given as input to the acoustic model. The mel filterbank smooths out adjacent frequency bins, so it can mitigate the influence of noise in the mel frequency if the noise occurs in isolated frequency bands. Nonetheless, if some additive noise E perturbs the input, then $Y_{\text{noisy}} = AX_{\text{noisy}} = A(X+E) = Y + AE$. This results in feature mismatch between clean and noisy speech.

We propose to use the time-activation matrices from CNMF as features for the acoustic model. Crucially, we describe an algorithm that reduces the effect of noise on the resulting time-activation matrices. The following sections describe the steps of the algorithm, and Figure 1 summarizes the algorithm in a flowchart.

2.1. Step 1: Learn a speech dictionary

Speech contains certain spectro-temporal properties that help distinguish it from background noise. CNMF is an algorithm that discovers the spectro-temporal building blocks of speech and stores the building blocks in a time-varying dictionary. CNMF decomposes a spectrogram $V \in \mathbb{R}_+^{m \times n}$ into a time-varying dictionary $W \in \mathbb{R}_+^{m \times K \times T}$ and time-activation matrix $H \in \mathbb{R}_+^{K \times n}$ by minimizing the divergence between V and $\hat{V} := \sum_{t=0}^{T-1} W(t) \overset{t\rightarrow}{H}$. $W(t)$ refers to the dictionary at time t (the third dimension of W) and $\overset{t\rightarrow}{H}$ means that the columns of H are shifted t columns to the right and t all-zero columns are filled in on the left. In this work, we minimize the generalized KL divergence between V and $\hat{V}$:

$$D\left(V \| \hat{V}\right) = \sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij} \ln\left(\frac{V_{ij}}{\hat{V}_{ij}}\right) - V_{ij} + \hat{V}_{ij} \tag{1}$$

To learn a speech dictionary, we concatenate the clean speech from a stereo dataset into one long utterance and create the spectrogram V_{clean} from this utterance. We use CNMF to decompose V_{clean} into a spectro-temporal speech dictionary W_{speech} and time-activation matrix H_{clean}. Researchers have shown that imposing sparsity on the time-activation matrix improves the quality of the dictionary [17, 18]. Thus, we augment the generalized KL divergence with an L_1 penalty on the time-activation matrix to encourage sparsity:

$$C_{\text{speech}} = D\left(V_{\text{clean}} \| \hat{V}_{\text{clean}}\right) + \lambda \sum_{k=1}^{K} \sum_{j=1}^{n} H_{kj}^{\text{clean}}, \tag{2}$$

where $\hat{V}_{\text{clean}} := \sum_{t=0}^{T-1} W_{\text{speech}}(t) \overset{t\rightarrow}{H}_{\text{clean}}$ and λ controls the level of sparsity of H. To minimize Equation 2, we iteratively update W_{speech} and H_{clean} with the following multiplicative updates:

$$W_{\text{speech}}(t) \leftarrow W_{\text{speech}}(t) \otimes \frac{\frac{V_{\text{clean}}}{\hat{V}_{\text{clean}}} \overset{t\rightarrow}{H^{\intercal}}_{\text{clean}}}{\mathbf{1}_{m \times n} \overset{t\rightarrow}{H^{\intercal}}_{\text{clean}}}, \forall t \in \{0, \ldots, T-1\} \tag{3a}$$

$$H_{\text{clean}} \leftarrow H_{\text{clean}} \otimes \frac{\sum_{t=0}^{T-1} W_{\text{speech}}^{\intercal}(t) \overset{\leftarrow t}{\left[\frac{V_{\text{clean}}}{\hat{V}_{\text{clean}}}\right]}}{\sum_{t=0}^{T-1} \left(W_{\text{speech}}^{\intercal}(t) \mathbf{1}_{m \times n}\right) + \lambda}, \tag{3b}$$

where $\otimes$ means element-wise multiplication and the division is element-wise.

2.2. Step 2: Learn a noise dictionary

We also use CNMF to learn the spectro-temporal properties of noise. Importantly, we want the noise dictionary to capture as much of the perturbations due to noise so that the time-activation matrix is unaffected by noise. That is, suppose we have clean speech V_{clean} that decomposes into W_{speech} and H_{clean}, and we have the corresponding speech corrupted by noise V_{noisy}. Then, we would like to find a noise dictionary W_{noise} such that the CNMF decomposition of V_{noisy} also yields the time-activation matrix H_{clean}.

To achieve this goal, we minimize the following cost function:

$$C_{\text{noisy}} = D\left(V_{\text{noisy}} \| \hat{V}_{\text{noisy}}\right) + \lambda \sum_{k=1}^{K} \sum_{j=1}^{n} H_{kj}^{\text{clean}}, \tag{4}$$

where $\hat{V}_{\text{noisy}} := \sum_{t=0}^{T-1} \left(W_{\text{speech}}(t) + W_{\text{noise}}(t)\right) \overset{t\rightarrow}{H}_{\text{clean}}$. The idea behind this cost function is to try to push the variability due to noise into W_{noise}. This formulation is similar to total variability modeling [19], where W_{speech} represents the universal background model (UBM) and W_{noise} represents the shift in the UBM due to some source of variability (in this case, noise).

To learn a noise dictionary, we pair the clean and noisy utterances in the stereo dataset. We concatenate the clean utterances and the noisy utterances and create spectrograms from these concatenated utterances V_{clean} and V_{noisy}. With V_{clean} and W_{speech} fixed, we run Equation 3b to get H_{clean}. Then, with V_{noisy}, W_{speech}, and H_{clean} fixed, we obtain the spectro-temporal noise dictionary W_{noise} by using the following update rule that minimizes Equation 4:

$$W_{\text{noise}}(t) \leftarrow W_{\text{noise}}(t) \otimes \frac{\frac{V_{\text{noisy}}}{\hat{V}_{\text{noisy}}} \overset{t\rightarrow}{H^{\intercal}}_{\text{clean}}}{\mathbf{1}_{m \times n} \overset{t\rightarrow}{H^{\intercal}}_{\text{clean}}}, \forall t \in \{0, \ldots, T-1\} \tag{5}$$

2.3. Step 3: Learn a time-varying projection

Once we have the speech and noise dictionaries in hand, we can generate time-activation matrices for the entire dataset. However, note that the CNMF cost function minimizes the signal reconstruction error; that is, it will find the time-activation matrix H_{utt} for each utterance V_{utt} that minimizes the KL divergence between V_{utt} and $\sum_{t=0}^{T-1} \left(W_{\text{speech}}(t) + W_{\text{noise}}(t)\right) \overset{t\rightarrow}{H}_{\text{utt}}$. This cost function is appropriate when you want the reconstructed signal (eg. denoised speech). What is important when using the time-activation matrices as features is the reduction in mismatch between the matrices from clean and noisy speech, which is not guaranteed by the CNMF cost function.

To reduce feature mismatch, we find a time-varying projection matrix $P \in \mathbb{R}_+^{K \times m \times T}$ that denoises the time-activation matrices from noisy speech by projecting them onto the space containing the time-activation matrices from clean speech. The cost function that achieves this is

$$C_{\text{proj}} = D\left(H_{\text{clean}} \| \hat{H}_{\text{denoised}}\right) + D\left(\hat{H}_{\text{clean}} \| \hat{H}_{\text{denoised}}\right), \tag{6}$$

where $\hat{H}_{\text{clean}} := \sum_{t=0}^{T-1} P(t) \overset{t\rightarrow}{\hat{V}}_{\text{clean}}$, $\hat{H}_{\text{denoised}} := \sum_{t=0}^{T-1} P(t) \overset{t\rightarrow}{\hat{V}}_{\text{denoised}}$, and $\hat{V}_{\text{denoised}} := \sum_{t=0}^{T-1} W_{\text{speech}}(t) \overset{t\rightarrow}{H}_{\text{noisy}}$. The first part of the cost function minimizes the divergence between the denoised and target clean time-activation matrices. The second part of the cost function ensures that P projects time-activation matrices from clean and noisy speech in the same way. The second part is useful during feature extraction (Step 4) where it is unknown whether the utterance

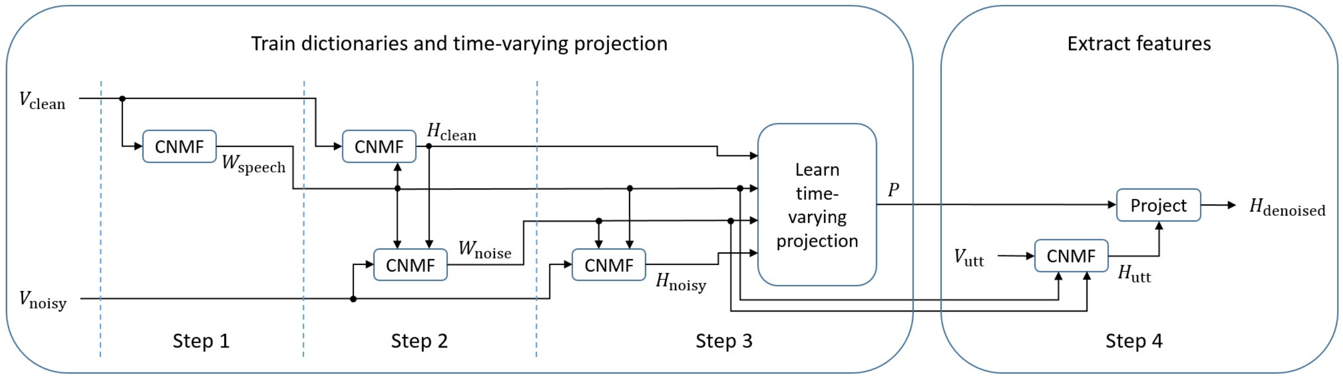


Fig. 1: Flowchart illustrating the algorithm for generating noise-robust time-activation matrices.

is clean or noisy. Equation 6 can be minimized with the following multiplicative update:

$$P(t) \leftarrow P(t) \otimes \frac{\mathbf{1}\overset{t\rightarrow}{\hat{V}^{\intercal}}_{\text{clean}} + \frac{H_{\text{clean}}+\hat{H}_{\text{clean}}}{\hat{H}_{\text{denoised}}}\overset{t\rightarrow}{\hat{V}^{\intercal}}_{\text{denoised}}}{\left(\mathbf{1} + \ln\left(\frac{\hat{H}_{\text{clean}}}{\hat{H}_{\text{denoised}}}\right)\right)\overset{t\rightarrow}{\hat{V}^{\intercal}}_{\text{clean}} + \mathbf{2}\overset{t\rightarrow}{\hat{V}^{\intercal}}_{\text{denoised}}}, \quad \forall t \in \{0, \ldots, T-1\} \tag{7}$$

To learn the time-varying projection, we pair the clean and noisy utterances. For the clean utterances, we run CNMF with W_{speech} fixed to get H_{clean}. For the noisy utterances, we run CNMF with W_{speech} and W_{noise} fixed to get H_{noisy}. We then learn the time-varying projection with Equation 7.

2.4. Step 4: Extract acoustic features

Once we have learned the time-varying projection, we are ready to generate time-activation matrices for the entire dataset as features for the acoustic model. For each utterance V_{utt} in the corpus, we find the time-activation matrix H_{utt} with W_{speech} and W_{noise} fixed using the following update rule:

$$H_{\text{utt}} \leftarrow H_{\text{utt}} \otimes \frac{\sum_{t=0}^{T-1}\left(W_{\text{speech}}(t) + W_{\text{noise}}(t)\right)^{\intercal}\overset{\leftarrow t}{\left[\frac{V_{\text{utt}}}{\hat{V}_{\text{utt}}}\right]}}{\sum_{t=0}^{T-1}\left(\left(W_{\text{speech}}(t) + W_{\text{noise}}(t)\right)^{\intercal}\mathbf{1}_{m\times n}\right) + \lambda}, \tag{8}$$

where $\hat{V}_{\text{utt}} := \sum_{t=0}^{T-1}\left(W_{\text{speech}}(t) + W_{\text{noise}}(t)\right)\overset{t\rightarrow}{H}_{\text{utt}}$. Then, we use the time-varying projection P to calculate the denoised time-activation matrix $H_{\text{denoised}} = \sum_{t=0}^{T-1} P(t)\overset{t\rightarrow}{\hat{V}}_{\text{denoised}}$, where $\hat{V}_{\text{denoised}} := \sum_{t=0}^{T-1} W_{\text{speech}}(t)\overset{t\rightarrow}{H}_{\text{utt}}$. We input $\log(H_{\text{denoised}})$ as features into the acoustic model.

3. ASR EXPERIMENT

We investigated the performance of the proposed algorithm on the Aurora 4 corpus [20]. The training set consists of 7137 multi-condition sentences from the Wall Street Journal database. The noisy utterances are corrupted with one of six different noise types (airport, babble, car, restaurant, street traffic, and train station) at 10–20 dB SNR. The standard Aurora 4 test set consists of 330 base utterances from 8 speakers, with each of the utterances corrupted by the same six noises with SNRs ranging from 5–15 dB. The test set is divided into four categories:

- A: clean speech with near-field microphone.
- B: average of all noise conditions with near-field microphone.
- C: clean speech with far-field microphone.
- D: average of all noise conditions with far-field microphone.

The acoustic model for the ASR is a 7-layer fully-connected deep neural network (DNN) with 1024 neurons per hidden layer and 2000 neurons in the output layer. We use the rectified linear units (ReLU) activation function and a fixed dropout rate of 50% for layers 4 and 5. The training is based on the cross-entropy criterion, using stochastic gradient descent (SGD) and a mini-batch size of 256. We apply speaker-independent global mean and variance normalization to the features prior to augmenting them with delta and delta-delta, followed by splicing of 5 frames to the left and right for context. We used the task-standard WSJ0 bigram language model. The Aurora 4 test set is decoded using the IBM Attila dynamic decoder [21].

We ran two baseline experiments: extracting 40-dimensional log-mel features from the unprocessed speech and extracting 40-dimensional log-mel features from speech denoised by CNMF. To obtain denoised speech, we calculated the denoised spectrogram

$$V_{\text{denoised}} = \frac{\hat{V}_{\text{speech}}}{\hat{V}_{\text{speech}} + \hat{V}_{\text{noise}}} \otimes V_{\text{utt}} \tag{9}$$

for each utterance V_{utt}, with $\hat{V}_{\text{speech}} = \sum_{t=0}^{T-1} W_{\text{speech}}(t)\overset{t\rightarrow}{H}_{\text{utt}}$ and $\hat{V}_{\text{noise}} = \sum_{t=0}^{T-1} W_{\text{noise}}(t)\overset{t\rightarrow}{H}_{\text{utt}}$. We converted the denoised spectrogram to the time-domain using the overlap-add method [22].

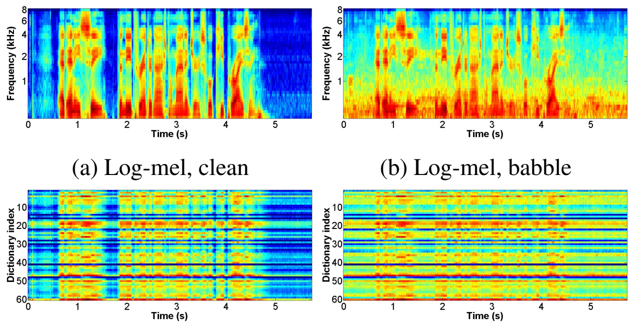
Next, we generated time-activation matrices in three different ways: using only a speech dictionary, using a speech and noise dictionary and keeping the rows of the activation matrix corresponding to the speech dictionary, and using the algorithm described in the previous section. We used $K = 60$, $T = 5$, and $\lambda = 2$ to generate these matrices. Furthermore, we appended the time-activation matrices generated using the proposed method to log-mel features. Table 1 shows the word error rates (WER) for all the experiments.

4. DISCUSSION

Table 1 shows that log-mel features extracted from denoised speech performed worse than log-mel features extracted from unprocessed speech. As mentioned previously, denoising is a common step taken by researchers when performing ASR on noisy speech. Our results indicate, in the context of multi-condition training, that it is better not

Table 1: Word error rates for different acoustic model features in different noise and channel conditions.

Feature	A	B	C	D	Average
Log-mel, unprocessed speech	4.82	8.32	8.03	16.35	12.34
Log-mel, denoised speech	5.29	9.67	9.19	19.36	14.52
Time-activations, speech dictionary	5.06	8.35	10.50	18.69	13.53
Time-activations, speech+noise dict.	5.01	8.29	10.42	18.27	13.29
Time-activations, proposed algorithm	4.43	7.34	7.86	16.29	11.82
Log-mel + time-activations	**4.22**	**7.17**	**7.70**	**15.56**	**11.37**



(a) Log-mel, clean (b) Log-mel, babble

(c) Time-activations, clean (d) Time-activations, babble

Fig. 2: Comparison of log-mel features and time-activation matrices for an Aurora 4 utterance.

to denoise the speech. Denoising most likely increases the WER because it introduces distortions and artifacts in the signal. Since most features, including log-mel features, are calculated directly from the signal, the features capture the artifacts, thus increasing the mismatch between features from clean and noisy speech. Moreover, the distortions and artifacts can vary by noise and SNR level. These introduce additional sources of variability in the log-mel features.

The results show that using the time-activation matrices directly as features outperforms using them as a denoising pre-processing step. Unfortunately, calculating the time-activation matrices with only a speech dictionary performs below log-mel features on unprocessed speech. Since the speech dictionary is fixed when generating features, a poorer performance is expected because the variability due to noise had to be captured by the time-activation matrix, making it susceptible to noise. Adding the noise dictionary gave slight improvements because it was able to capture some of the variability due to noise. However, the noise dictionary did not adapt to different noises during feature extraction, reducing its efficacy in capturing the noise variability. On the other hand, generating time-activation matrices with the proposed algorithm outperformed all of the previous experiments. In different noise conditions with the near-field microphone (category B), we achieved a 11.8% relative improvement over log-mel features on unprocessed speech. This result suggests that designing noise-robust features can improve ASR performance on noisy speech compared to extracting standard features on unprocessed or denoised speech.

Finally, appending the time-activation matrices to the log-mel features gives the best-performing system. In category B, we achieved a 13.8% relative improvement over log-mel features on unprocessed speech. The improvement in performance over using just the time-activation matrices indicates that the time-activation matrices contain complementary information to log-mel features. The log-mel features are a low-dimensional projection of the spectrogram, and so they contain spectral information. On the other hand, the time-activation matrix is an encoding of the spectrogram relative to the speech dictionary. Thus, the time-activation matrix doesn't contain spectral information, but rather shows the magnitude of different spectro-temporal speech patterns at each frame. For visualization, Figure 2 compares the log-mel features and time-activation matrices extracted for an Aurora 4 utterance in clean and babble noise. Notice that the time-activation matrix for babble noise is more closely matched to the matrix for clean speech than the log-mel features for babble noise are to clean log-mel features.

A limitation of our algorithm is the need for clean versions of noisy speech in the corpus (stereo dataset). We used the clean speech when learning the dictionaries and time-varying projection. This limits our approach to datasets with clean speech. One approach around this constraint is to learn the dictionaries and projection on a different stereo dataset, and then apply the dictionaries and projection when extracting features on a non-stereo dataset. Another workaround is to use a voice activity detector (VAD) to learn the speech dictionary only from frames that have a high confidence of containing speech. Additionally, the frames marked as non-speech can be used to adapt the noise dictionary during the feature extraction step. Extending on the VAD idea, we can obtain a measure of speech confidence at the frame and frequency levels directly from CNMF using $\hat{V}_{\text{speech}} \Big/ \left(\hat{V}_{\text{speech}} + \hat{V}_{\text{noise}}\right)$. This matrix contains values between 0 and 1 that indicate the proportion of the signal energy belonging to speech. We can modify the cost function for learning the speech dictionary to place greater weight on regions with high speech proportion. Similarly, we can bias the noise dictionary learning to favor regions with low speech proportion.

5. CONCLUSION

We proposed an algorithm to generate noise-robust time-activation matrices using CNMF, and we used these as features for the acoustic model. The algorithm centered upon forcing the variability due to noise out of the time-activation matrices and into the dictionaries. ASR results on the Aurora 4 dataset indicate a 11.8% relative improvement of the WER over log-mel features. Furthermore, combining the time-activation matrices with log-mel features gives a 13.8% relative improvement of the WER over log-mel features. Our experiments show that our algorithm for creating time-activation matrices is more robust to noise and contains complementary information to log-mel features.

To build upon this work, we will explore ways to generate noise-robust time-activation matrices without access to clean speech, as mentioned in the previous section. We will explore ways to adapt the noise dictionary during feature extraction to increase its usefulness. We will also train the dictionaries discriminatively, instead of unsupervised as it is currently. We will investigate other approaches to generating noise-robust time-activation matrices, such as joint-adaptive training [23]. Finally, we will incorporate channel compensation into our algorithm.

6. REFERENCES

[1] C. M. Lee and S. Narayanan, "Towards detecting emotion in spoken dialogs," *IEEE Trans. Speech and Audio Process.*, vol. 13, no. 2, pp. 293–302, Mar. 2005.

[2] B. Yin, F. Chen, N. Ruiz, and E. Ambikairajah, "Speech-based cognitive load monitoring system," in *Proc. Int. Conf. Acoustics, Speech, and Signal Process.*, 2008, pp. 2041–2044.

[3] D. Macho, L Mauury, B. Noé, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a noise-robust DSR front-end on AURORA databases," in *Proc. Int. Conf. Spoken Lang. Process.*, 2002, pp. 17–20.

[4] T. Yoshioka and T. Nakatani, "Noise model transfer: novel approach to robustness against nonstationary noise," *IEEE Trans. Acoustics, Speech, and Lang. Process.*, vol. 21, no. 10, pp. 2182–2192, Oct. 2013.

[5] J. Droppo, A. Acero, and L. Deng, "Evaluation of the SPLICE algorithm on the Aurora2 database," in *Proc. Eurospeech*, 2001, pp. 217–220.

[6] O. Kalinli, M. L. Seltzer, J. Droppo, and A. Acero, "Noise adaptive training for robust automatic speech recognition," *IEEE Trans. Acoustics, Speech, and Lang. Process.*, vol. 18, no. 8, pp. 1889–1901, Nov. 2010.

[7] Y. Wang and M. J. F. Gales, "Speaker and noise factorization for robust speech recognition," *IEEE Trans. Acoustics, Speech, and Lang. Process.*, vol. 20, no. 7, pp. 2149–2158, Sep. 2012.

[8] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. Int. Conf. Acoustics, Speech, and Signal Process.* IEEE, 2013, pp. 7398–7402.

[9] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoustics, Speech, and Signal Process.*, vol. 20, no. 2, pp. 113–120, Apr. 1979.

[10] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[11] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Adv. in Neu. Info. Proc. Sys. 13*, 2001, pp. 556–562.

[12] A. Narayanan and D. L. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 22, no. 4, pp. 826–835, 2014.

[13] P. J. Moreno, B. Raj, and R. M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Proc. Int. Conf. Acoustics, Speech, and Signal Process.* IEEE, 1996, pp. 733–736 vol. 2.

[14] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, "High-performance robust speech recognition using stereo training data," in *Proc. Int. Conf. Acoustics, Speech, and Signal Process.* IEEE, 2001, pp. 301–304.

[15] C. Kim and R. M. Stern, "Power-Normalized Cepstral Coefficients (PNCC) for robust speech recognition," in *Proc. Int. Conf. Acoustics, Speech, and Signal Process.* IEEE, 2012, pp. 4101–4104.

[16] P. Smaragdis, "Convolutive Speech Bases and Their Application to Supervised Speech Separation," *IEEE Trans. Acoustics, Speech, and Lang. Process.*, vol. 15, no. 1, pp. 1–12, Jan. 2007.

[17] P. O. Hoyer, "Non-negative Matrix Factorization with Sparseness Constraints," *J. Machine Learning Research*, vol. 5, pp. 1457–1469, Dec. 2004.

[18] P. D. O'Grady and B. A. Pearlmutter, "Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint," *Neurocomputing*, vol. 72, no. 1-3, pp. 88–101, Dec. 2008.

[19] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE/ACM Trans. Acoustics, Speech, and Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[20] N. Parihar and J. Picone, "Analysis of the Aurora large vocabulary evaluations," in *Proc. Eurospeech*, 2003, pp. 337–340.

[21] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila Speech Recognition Toolkit," in *Proc. IEEE Workshop Spoken Lang. Technology*, Dec. 2010, pp. 97–102.

[22] D. Griffin and J.S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoustics, Speech, and Signal Process.*, vol. 32, no. 2, pp. 236–243, Apr. 1984.

[23] A. Narayanan and D. L. Wang, "Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training," *IEEE/ACM Trans. Acoustics, Speech, and Lang. Process.*, vol. 23, no. 1, pp. 92–101, Jan. 2015.