THE CMU-MIT REVERB CHALLENGE 2014 SYSTEM: DESCRIPTION AND RESULTS

Xue Feng[1]*, Kenichi Kumatani*[2]*, John McDonough*[2]

[1]Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139, USA

[2]Carnegie Mellon University
Language Technologies Institute
Gates Hillman Complex
5000 Forbes Avenue, Pittsburgh, PA 15213, USA

ABSTRACT

To evaluate state-of-the-art algorithms and draw new insights regarding potential future research directions in distant speech recognition, Kinoshita *et al.* [1] launched the *REverberant Voice Enhancement and Recognition Benchmark Challenge*, commonly known as the REVERB Challenge, intended to provide a test bed for researchers to evaluate their methods based on common corpora and evaluation metrics. In this work, we describe our system and present our results on the 2014 REVERB Challenge (RC). Our system is comprised of four primary components: an acoustic *speaker tracking system* to determine the speaker's position; this position is used for *beamforming* to focus on the desired speech while suppressing noise and reverberation; *speaker clustering* to determine sets of utterances spoken by the same speaker; and a *speech recognition engine* with *speaker adaptation* to extract word hypotheses from the enhanced waveforms produced by the beamformer. On the REAL RC evaluation data, our system obtained a word error rate of 39.9% with a single channel of the array, and 16.9% with the best beamformed signal.

Index Terms— Robust Speech Recognition, Microphone arrays

1. INTRODUCTION

Distant speech recognition (DSR) has recently gained a great deal of interest in the research community [2, 3, 4, 5, 6, 7, 8]. The REVERB Challenge (RC) addresses a certain level of fundamental issues in DSR. The RC data was comprised of two subcorpora: A *simulated corpus* was obtained by linearly convolving data captured with a close-talking microphone and adding noise; such a corpus could have been created at any time in the past 20 years. The *real corpus* was captured in a real meeting room with two circular, eight-channel microphone arrays; that portion of the challenge data was recorded at the University of Edinburgh by Lincoln *et al.* [9]. Results on portions of the corpus have long since been reported in the literature [10, 11, 12]. Indeed, the sole novel aspect of the REVERB Challenge is its requirement that speaker clustering be performed automatically prior to any speaker adaptation for the primary condition. Nonetheless, the REVERB Challenge seems to be the first such competition to have captured broad interest within the community, which is certainly a laudable accomplishment.

In this work, we describe our system and present our results on the REVERB Challenge 2014. Figure 1 presents a schematic diagram of our overall system. In Section 2, we discuss our system for speaker tracking. Our beamforming algorithms are presented in Section 3. We take up speaker clustering in Section 4. Section 5 presents our system for speaker adaptation and speech recognition. In Section 6, we provide evidence of the effectiveness of our system. In the last section of work, we present our conclusions as well as a prognosis for the future of the field.

2. SPEAKER TRACKING

In this section, we present our speaker tracking system, which, briefly, has two components. First, time delays of arrival are estimated between pairs of microphones with a known geometry. Subsequently, a Kalman filter is used to combine these measurements and infer the position of the speaker from them.

2.1. Time Delay of Arrival Estimation

Our speaker tracking system was based on estimation of *time delay of arrival* (TDOA) of the speech signal on the direct path from the speaker's mouth to unique pairs of microphones in the eight-element of array. TDOA estimation was performed with the well-known *phase transform* (PHAT) [13]

$$\rho_{mn}(\tau) \triangleq \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{Y_m(e^{j\omega\tau})Y_n^*(e^{j\omega\tau})}{|Y_m(e^{j\omega\tau})Y_n^*(e^{j\omega\tau})|}\, e^{j\omega\tau}\, d\omega, \qquad (1)$$

where $Y_n(e^{j\omega\tau})$ denotes the short-time Fourier transform of the signal arriving at the nth sensor in the array [14]. The definition of the PHAT in (1) follows directly from the frequency domain calculation of the cross-correlation of two sequences. The normalization term $\left|Y_m(e^{j\omega\tau})Y_n^*(e^{j\omega\tau})\right|$ in the denominator of the integrand is intended to weight all frequencies equally. It has been shown that such a weighting leads to more robust TDOA estimates in noisy and reverberant environments [15]. Once $\rho_{mn}(\tau)$ has been calculated, the TDOA estimate is obtained from

$$\hat{\tau}_{mn} = \max_{\tau}\ \rho_{mn}(\tau). \qquad (2)$$

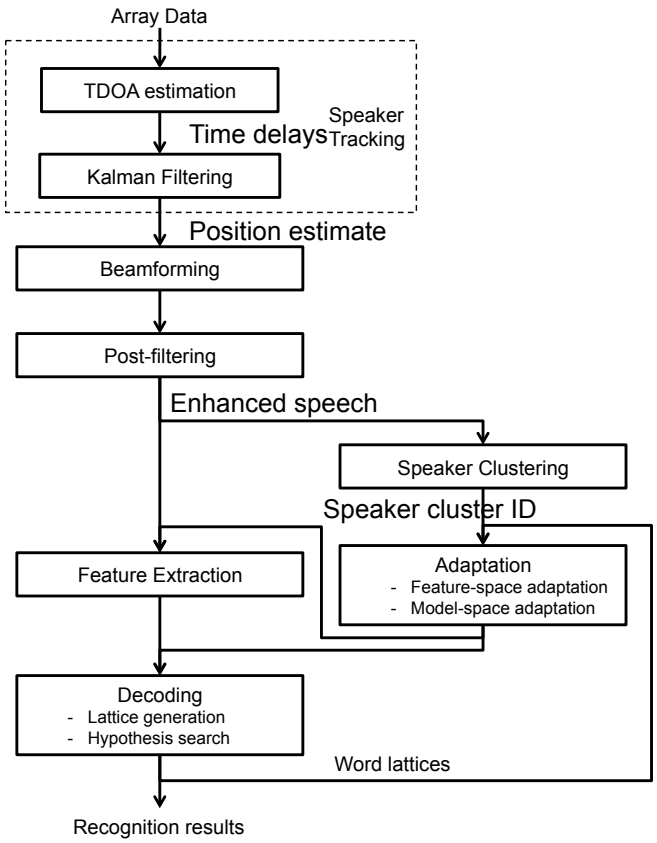

Fig. 1. Block diagram of the distant speech recognition system.

2.2. Kalman Filtering

Speaker tracking based on the maximum likelihood criterion [16] seeks to determine the speaker's position $\mathbf{x}$ by minimizing the error function

$$\epsilon(\mathbf{x}) = \sum_{s=0}^{S_2-1} \frac{[\hat{\tau}_s - T_s(\mathbf{x})]^2}{\sigma_s^2}, \tag{3}$$

where σ_s^2 denotes the error covariance associated with this observation, $\hat{\tau}_s$ is the observed TDOA as in (1) and (2), and $T_s(\mathbf{x})$ denotes the TDOAs predicted based on geometric considerations.

Although (3) implies that we should find $\mathbf{x}$ minimizing the instantaneous error criterion, we would be better advised to minimize such an error criterion over a series of time instants. In so doing, we exploit the fact that the speaker's position cannot change instantaneously; thus, both the present and past TDOA estimates are potentially useful in estimating a speaker's current position. Klee *et al.* [17] proposed to recursively minimize the least square error position estimation criterion (3) with a variant of the *extended Kalman filter* (EKF). This was achieved by first associating the *state* $\mathbf{x}_k$ of the EKF with the speaker's position at time k, and the kth observation with a vector of TDOAs. In keeping with the formalism of the EKF, Klee *et al.* [17] then postulated a *state* and *observation equation*,

$$\mathbf{x}_k = \mathbf{F}_{k|k-1}\mathbf{x}_{k-1} + \mathbf{u}_{k-1}, \text{ and} \tag{4}$$

$$\mathbf{y}_k = \mathbf{H}_{k|k-1}(\mathbf{x}_k) + \mathbf{v}_k, \tag{5}$$

respectively, where $\mathbf{F}_{k|k-1}$ denotes the *transition matrix*, $\mathbf{u}_{k-1}$ denotes the *process noise*, $\mathbf{H}_{k|k-1}(\mathbf{x})$ denotes the vector-valued *observation function*, and $\mathbf{v}_k$ denotes the *observation noise*. The process $\mathbf{u}_k$ and observation $\mathbf{v}_k$ noises are unknown, but both have zero-mean Gaussian pdfs and known covariance matrices, $\mathbf{U}_k$ and $\mathbf{V}_k$, respectively. Associating $\mathbf{H}_{k|k-1}(\mathbf{x})$ with the TDOA function $T_s(\mathbf{x})$ with one component per microphone pair, it is straightforward to calculate the appropriate linearization about the current state estimate required by the EKF [2, §10.2],

$$\bar{\mathbf{H}}_k(\mathbf{x}) \triangleq \nabla_{\mathbf{x}}\mathbf{H}_{k|k-1}(\mathbf{x}). \tag{6}$$

By assumption $\mathbf{F}_{k|k-1}$ is known, and the *predicted state estimate* is given by $\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k|k-1}\hat{\mathbf{x}}_{k-1|k-1}$, where $\hat{\mathbf{x}}_{k-1|k-1}$ is the state estimate from the prior time step. The innovation is defined as

$$\mathbf{s}_k \triangleq \mathbf{y}_k - \mathbf{H}_{k|k-1}\left(\hat{\mathbf{x}}_{k|k-1}\right).$$

The new filtered state estimate is obtained from

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{G}_k\,\mathbf{s}_k, \tag{7}$$

where $\mathbf{G}_k$ denotes the *Kalman gain* [2, §4.3]. A block diagram illustrating the prediction and correction steps in the state estimate update of a conventional Kalman filter is shown in Figure 2.

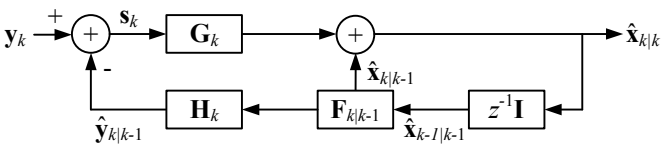

Fig. 2. Predictor-corrector structure of the Kalman filter.

The primary free parameters in our speaker tracking system are $\mathbf{U}_k$ and $\mathbf{V}_k$, the known covariances matrices of the process and observation noises, $\mathbf{u}_k$ and $\mathbf{v}_k$, respectively. In our system, we set $\mathbf{U}_k = \sigma_u^2\mathbf{I}$ and $\mathbf{V}_k = \sigma_v^2\mathbf{I}$, and then tuned σ_u^2 and σ_v^2 to provide the lowest tracking error, which required a multi-channel speech corpus with ground truth speaker positions; this requirement was admirably met by the corpus collected by Lathoud *et al.* [18].

Shown in Figure 3 is a plot of radial tracking error in radians as a function of σ_u^2 and σ_v^2. This study led us to choose the final parameters of $\sigma_u^2 = 0.1$ and $\sigma_v^2 = 1 \times 10^{-8}$ for our RC submission.

3. BEAMFORMING

The array processing component of our primary system was based on the super-directive maximum negentropy (SDMN) beamformer [19, 20], which incorporates the *super-Gaussianity* of speech into adaptive beamforming. It has been demonstrated through DSR experiments on the real array data in [12] that beamforming with the maximum negentropy (MN) criterion is more robust than conventional techniques against reverberation. This is due to the fact that MN beamforming strengthens the target signal by using reflected speech; hence MN beamforming is not susceptible to signal cancellation.

As shown in Figure 4, the SDMN beamformer has the generalized sidelobe canceller (GSC) architecture. The processing of SDMN beamforming can be divided into an upper branch and a lower branch. In the upper branch, the super-directive (SD) beamformer is used for the *quiescent vector* $\mathbf{w}_{\text{SD}}$. The process in the lower branch involves multiplication of the *block matrix* $\mathbf{B}$ and *active weight vector* $\mathbf{w}_{\text{a}}$. The beamformer's output for the array input vector $\mathbf{X}$ at frame k is obtained in the subband frequency domain as

$$Y(k,\omega) = \left(\mathbf{w}_{\text{SD}}(k,\omega) - \mathbf{B}(k,\omega)\mathbf{w}_{\text{a}}(k,\omega)\right)^H \mathbf{X}(k,\omega),$$

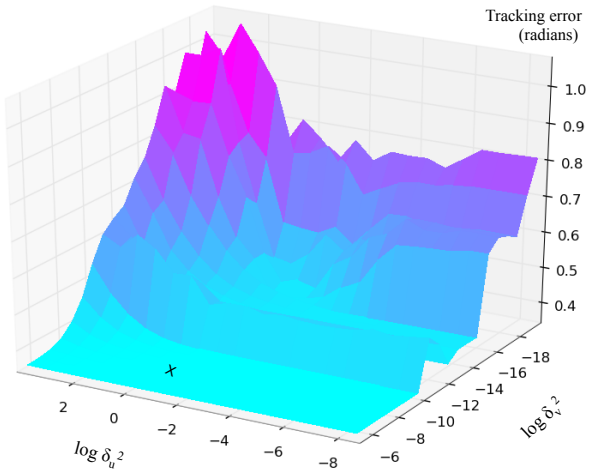

Fig. 3. Speaker tracking error vs. process and observation noise parameters. The 'x' mark denotes our resulting choice of the parameter values.

where ω is the angular frequency.

Let us define the *cross-correlation coefficient* between the inputs of the mth and nth sensors as

$$\rho_{mn}(\omega) \triangleq \frac{\mathcal{E}\{X_m(\omega)X_n^*(\omega)\}}{\sqrt{\mathcal{E}\{|X_m(\omega)|^2\}\,\mathcal{E}\{|X_n(\omega)|^2\}}}, \tag{8}$$

where $\mathcal{E}\{\cdot\}$ indicates the expectation operator. The super-directive design is then obtained by replacing the spatial spectral matrix [2, §13.4] with the *coherence matrix* $\mathbf{\Gamma_N}$ corresponding to a diffuse noise field. The m,nth component of the latter can be expressed as

$$\Gamma_{\mathbf{N},m,n}(\omega) = \mathrm{sinc}\left(\frac{\omega\, d_{m,n}}{c}\right) = \rho_{mn}(\omega), \tag{9}$$

where $d_{m,n}$ is the distance between the mth and nth elements of the array. Given the array manifold vector $\mathbf{d}$ computed with the position estimate, the weight of the SD beamformer can be expressed as

$$\mathbf{w}_{\mathrm{SD}} = \frac{(\mathbf{\Gamma_N} + \sigma_{\mathrm{d}}\mathbf{I})^{-1}\,\mathbf{d}}{\mathbf{d}^H\,(\mathbf{\Gamma_N} + \sigma_{\mathrm{d}}\mathbf{I})^{-1}\,\mathbf{d}}, \tag{10}$$

where σ_{d} is an amount of diagonal loading and set to 0.01 for experiments. Notice that the frequency and time indicies ω and k are omitted here for the sake of simplicity. The SD beamformer has been proven to be more suitable than delay-and-sum (DS) and minimum

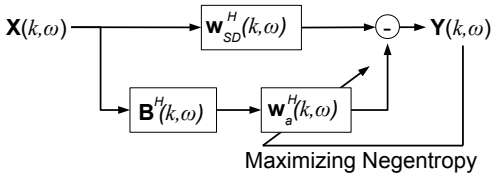

Fig. 4. Configuration of the super-directive maximum negentropy (SDMN) beamformer.

variance distortionless response (MVDR) beamformers in meeting room conditions [5, 9, 12].

Once the SD beamformer is fixed in the upper branch, the blocking matrix is constructed to satisfy the orthogonal condition $\mathbf{B}^H\mathbf{w}_{\mathrm{SD}} = \mathbf{0}$. Such a blocking matrix can be, for example, obtained with the modified Gram-Schmidt [21]. This orthogonality implies that the distortionless constraint for the direction of interest will be maintained for any choice of the active weight vector. In contrast to normal practice, the SD-MN beamformer seeks the active weight vector that maximizes the negentropy of the beamformer's output. Assuming that the speech subband samples can be modeled with the generalized Gaussian distribution (GGD) with shape parameter f, we can express the beamformer's negentropy as

$$\begin{aligned} J(Y) &= \log(\pi\sigma_Y^2) + 1 \\ &\quad - \left[\log\{2\pi\Gamma(2/f)B_f^2\hat{\sigma}_Y/f\} + 2/f\right], \end{aligned} \tag{11}$$

where

$$\sigma_Y^2 = \mathcal{E}\{|Y|^2\},$$

$$\hat{\sigma}_Y = \frac{1}{B_f}\left(\frac{f}{2}\right)^{1/f}\mathcal{E}\left\{|Y|^f\right\}^{1/f},$$

$$B_f = \sqrt{\Gamma(2/f)/\Gamma(4/f)},$$

and $\Gamma(\cdot)$ is the gamma function.

In this work, the shape parameter of the GGD is trained with the clean WSJCAM0 data of the clean training set based on the maximum likelihood criterion as described in [20].

In order to avoid large weights, we apply the regularization term to the optimization criterion. The modified optimization criterion can be written as

$$\mathcal{J}(Y) = J(Y) - \alpha|\mathbf{w}_{\mathrm{a}}|^2. \tag{12}$$

where α is set to 0.01 for the experiments.

Due to the absence of a closed-form solution with respect to $\mathbf{w}_{\mathrm{a}}$, we have to resort to the gradient-based numerical optimization algorithm. Upon taking the partial deviation of (12) with respect to $\mathbf{w}_{\mathrm{a}}$, we can obtain gradient information required for such a numerical optimization algorithm:

$$\frac{\partial\mathcal{J}(Y)}{\partial\mathbf{w}_{\mathrm{a}}{}^*} = \mathcal{E}\left[\left\{\frac{1}{\sigma_Y^2} - \frac{f|Y|^{f-2}}{2\,(B_f\hat{\sigma}_Y)^f}\right\}\mathbf{B}^H\mathbf{X}Y^* - \alpha\mathbf{w}_{\mathrm{a}}\right] \tag{13}$$

In this work, we use the Polak-Ribière conjugate gradient algorithm to find the solution.

3.1. Post-filtering

The post-filter used in our RC systems is a variant of the Wiener post-filter. One of the earliest and best-known proposals for estimating these quantities was by Zelinski [22]. A good survey of current techniques is given by Simmer *et al.* [23].

4. UNSUPERVISED SPEAKER CLUSTERING

In this section, we present our approach for grouping single-speaker speech utterances into speaker-specific clusters.

A core feature of our approach lies in the approximation of speaker-conditional statistics, and training the LDA parameters for finding the optimal discriminative subspace. Figure 5 shows the block diagram of the speaker clustering system.

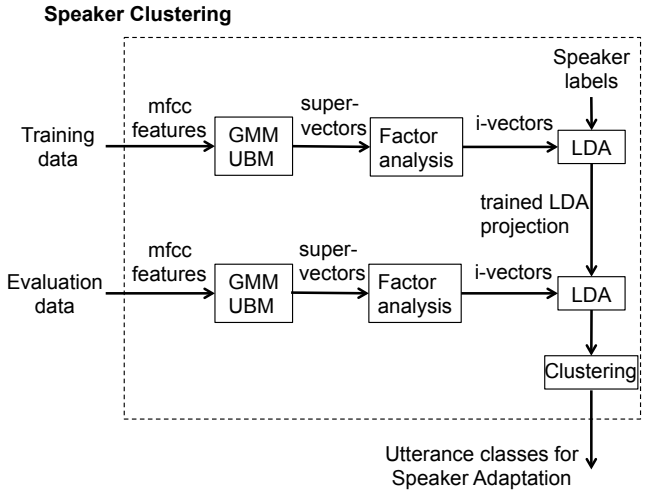

Fig. 5. Block diagram of the speaker clustering algorithm.

We start by computing supervectors. Next i-vectors are obtained by factor analysis. We then train an Linear Discriminant Analysis (LDA) matrix based projection from the i-vectors to a speaker-discriminant subspace. Speaker clusters are generated by recursively grouping the LDA feature vectors into the binary classes based on the Euclidean distance. Each cluster is recursively split until a Bayesian information criterion (BIC) converges to the predefined threshold. Thus, our binary tree clustering algorithmis performed in the fully automatic manner.

4.1. Supervectors for Speakers

For each utterance, a Gaussian Mixture Model (GMM) [24] with 512 mixtures is adapted, given appropriate front-end features (39-dimensional MFCC [25] features). We denote the GMM mean components, which are speaker-dependent, as supervectors $\mathbf{M}$. The Universal Background Model (UBM) [24] is a large GMM trained over all utterances to represent the speaker-independent distribution of features. We denote the UBM mean components, which are speaker-independent, as UBM vector $\mathbf{m}$.

4.2. Factor Analysis and i-Vectors

According to Total Variability Factor Analysis [26], given an utterance, the supervector $\mathbf{M}$ can be rewritten as follows:

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w} \tag{14}$$

The key assumption in factor analysis is that the GMM supervector of the speaker- and channel-dependent $\mathbf{M}$ for a given utterance can be broken down into the sum of two supervectors where supervector $\mathbf{m}$ is the speaker- and session-independent supervector taken from a UBM, $\mathbf{T}$ is a rectangular matrix of low rank that defines the variability space and $\mathbf{w}$ is a low-dimensional (90-dimensional in our system) random vector with a normally distributed prior $\mathcal{N}(0, 1)$. We refer to these new vectors $\mathbf{w}$ as identity vectors or i-vectors for short.

4.3. Linear Discriminant Analysis

The i-vectors $\mathbf{w}$ obtained from factor analysis contain both speaker and channel dependent information. To extract the speaker-discriminant subspace, LDA is applied to map the i-vectors to a 10-dimensional subspace.

The LDA criterion requires class labels to calculate class means as well as class covariance matrices, and must thus be supervised. We trained our LDA projection on the simulated training data and applied the projection matrix on the evaluation set to perform unsupervised dimensionality reduction.

4.4. Binary Tree Clustering Algorithm

After LDA, the binary tree clustering algorithm is performed on the subspace vectors in order to find speaker clusters. We first split the observations into two clusters based on the the Euclidean distance between the LDA feature vectors. Each cluster is further split into two clusters. Every time the binary class is generated, we check the BIC which indicates a degree of fitness of the model. Under the assumption that the model errors are independent and identically distributed according to a normal distribution, such a criterion can be expressed as

$$\text{BIC} = N \ln\left(\sigma_{\text{e}}^2\right) + K \ln\left(N\right) \tag{15}$$

where σ_{e}^2 is the error variance of the class, K is the number of the parameters and N is the number of utterances. Binary clustering is recursively performed until the difference of the BIC becomes below the threshold. In preliminary experiments on the development set, we chose 100.0 as the BIC threshold. Notice that our clustering algorithm does not require any prior information about a number of speakers and acoustic conditions.

5. SPEAKER ADAPTATION AND SPEECH RECOGNITION

The final component of our system is an engine for performing unsupervised speaker adaptation and speech recogntion. In this section, we describe the training and operation of these component.

5.1. Feature Extraction

The feature extraction of our ASR system was based on cepstral features estimated with a warped *minimum variance distortionless response* [27] (MVDR) spectral envelope of model order 30. Due to the properties of the warped MVDR, neither the Mel-filterbank nor any other filterbank was needed. The warped MVDR provides an increased resolution in low–frequency regions relative to the conventional Mel-filterbank. The MVDR also models spectral peaks more accurately than spectral valleys, which leads to improved robustness in the presence of noise. Front-end analysis involved extracting 20 cepstral coefficients per frame of speech and performing global cepstral mean subtraction (CMS) with variance normalization. The final features were obtained by concatenating 15 consecutive frames of cepstral features together, then performing the LDA to obtain a feature of length 42.

5.2. System Training

Our best RC system was based on two acoustic models. The first model was trained on the clean WSJCAM0 [28] and WSJ0 corpora. Training consisted of conventional HMM training, with three passes of forward-backward training followed by Gaussian splitting and more training [29]; this was followed by speaker-adapted training (SAT) [2, §8.1.3].

To train the second acoustic model, we first took the WSJ0 and WSJCAM0 corpora and "dirtied" them up through convolution with

the multi-channel room impulse responses and addition of the multi-channel noise provided with the RC data. These dirty multi-channel streams were then used first for speaker tracking then for beamforming. Once we had produced the final processed single stream of data, they were once more used first for conventional HMM training and then for speaker-adapted training.

5.3. Recognition and Adaptation Passes

We performed four decoding passes on the waveforms obtained from the beamforming algorithm described in Section 3. Each pass of decoding used a different acoustic model or speaker adaptation scheme. For all passes save the first unadapted pass, speaker adaptation parameters were estimated using the word lattices generated during the prior pass, as in [30]. A description of the four decoding passes follows:

1. Decode with the unadapted, conventional ML acoustic model.
2. Estimate vocal tract length normalization (VTLN) [31] parameters and constrained maximum likelihood linear regression parameters (CMLLR) [32] for each speaker, then redecode with the conventional ML acoustic model.
3. Estimate VTLN, CMLLR, and maximum likelihood linear regression (MLLR) [33] parameters for each speaker, then redecode with the conventional model.
4. Estimate VTLN, CMLLR, MLLR parameters for each speaker, then redecode with the ML-SAT model.

All passes used the full trigram LM for the 5,000 word WSJ task, which was made possible through the fast-on-the-fly composition algorithm described in [34].

For the primary system, the true speaker identity for each utterance was replaced by the cluster index obtained through the clustering algorithm described in Section 4. The contrast system used the true speaker identities for speaker adaptation.

6. RESULTS

Table 1 shows the word error rates (WERs) obtained with our systems on the RC data. The results obtained with a single array channel (SAC) and close-talking microphone (CTM) are also presented in Table 1 as a contrast condition. All of our RC systems were based on full batch processing, although we anticipate that practical implementations could use frame-by-frame processing with little degradation in accuracy. All systems used the *Millennium* speech recognition engine, which is based on weighted finite-state transducers [35].

Primary System

In our primary system, the speaker tracking, speaker clustering, beamforming, feature extraction, speech recognition and speaker adaptation components were all developed as described in Sections 2 through 5. The array processing components of the system—speaker tracking and beamforming—both used eight channels of audio data from the circular arrays. Unsupervised speaker clustering was performed based on the i-vectors as described in Section 4.

For the first pass of the primary system, we trained the acoustic model with noisy speech processed with SD beamforming, described in Section 5.2. For the adapted passes, we used acoustic models trained based on clean WSJ0 and WSJCAM0 corpora as described in Section 5.2. Our final primary system employs the noisy acoustic model in the first pass and then switches to the clean acoustic model in the adapted passes.

Secondary System

We used the secondary system for our first result submission. A main difference between the primary and secondary systems is that the secondary system uses the K-mean clustering algorithm for speaker clustering. The number of clusters K is determined in preliminary experiments. Another difference is that the secondary system uses the clean acoustic model only. 40 and 20 clusters are used for SimData and RealData experiments. Although the K-mean clustering algorithm provides the better result, this could potentially violates one of the RV regulations.

Contrast System

The only difference between the primary and contrast systems was that the unsupervised speaker clustering used in the former was replaced by the true speaker labels in the latter, as determined by the names of the audio files, for the purpose of speaker adaptation. We built two contrast systems with SD-MN beamforming (Contrast A) and conventional SD beamforming (Contrast B). The results in Table 1 suggests that the beamforming method with the maximum negentropy criterion is more robust against reverberation. This is due to the facto that MN beamforming enhances the target signal by manipulating its weights so as to delay and add the reflections [12].

6.1. Comparison of Different Speaker Clustering Strategies

K-means clustering [36] is perhaps one of the most straightforward speaker clustering methods for unsupervised adaptation. Namely, given a set of N observation samples in R^D and the number of clusters K, the objective of the K-means algorithm is to determine a set of K points in R^D and the means so as to minimize the mean squared distance from each data point to its nearest mean.

Table 2 shows WERs obtained with our binary tree clustering and K-means clustering algorithms under the same condition. Table 2 also shows the WERs obtained with true speaker identities as a reference. In the K-mean clustering algorithm, we used 40 and 20 clusters for SimData and RealData experiments respectively. It is clear from Table 2 that the K-mean clustering algorithm can provide the better speech recognition performance.

It is also clear from Table 2 that the use of the true speaker labels yielded a reduction in error rate of approximately 1.0% absolute for the simulated data; the reduction was larger, approximately 4.5% absolute for the real data. This difference in behavior is ascribed to the fact that the simulated WSJCAM0 training data, which was used to estimate the LDA transformation on the i-vectors prior to K-means clustering, matched the simulated evaluation set much better than the real evaluation set. Hence, the separation of speaker classes was better for the simulated data than for the real data.

However, speaker clustering based on the K-means algorithm typically requires a good estimation of K which is associated with the number of speakers. In contrast, binary tree clustering with the BIC does not require any knowledge about the number of speakers. The number of clusters is determined solely based on the BIC, a indicator of a degree of over-fitting for the given adaptation data. In the case that the number of clusters is close to the actual number of speakers or fewer than that, the BIC tends to converge.

7. CONCLUSIONS

The 2014 REVERB Challenge is the first single speaker challenge to address DSR with speech material captured from real human speak-

		Simulated Data							Real Data		
		Room 1		Room 2		Room 3			Room 1		
System	Speaker Clustering	Near	Far	Near	Far	Near	Far	Ave.	Near	Far	Ave.
Primary	Binary tree with BIC	12.89	14.71	14.09	19.38	16.62	31.45	18.19	17.18	20.16	18.73
Secondary	K-means	8.44	8.91	9.99	13.19	10.77	20.17	11.91	16.74	19.51	18.13
Contrast A (MN BF)	Ground truth	7.74	8.68	9 .33	12.81	9.54	19.74	11.31	13.41	15.06	14.50
Contrast B (SD BF)	Ground truth	8.17	9.23	10.10	15.00	15.00	29.04	14.42	16.7	17.93	17.31
SAC	Ground truth	8.4	10.27	14.1	30.54	17.11	44.65	20.85	38.38	41.41	39.9
CTM	Ground truth	6.81	6.81	7.59	7.59	7.08	7.08	7.16	7.98	7.36	7.67

Table 1. Word error rate results of REVERB Challenge 2014 for primary and contrast conditions.

	Simulated Data							Real Data		
	Room 1		Room 2		Room 3			Room 1		
Clustering algorithm	Near	Far	Near	Far	Near	Far	Ave.	Near	Far	Ave.
Binary tree clustering with BIC	12.89	14.71	14.09	19.38	16.62	31.45	18.19	17.18	20.16	18.73
K-means clustering	8.44	8.91	9.99	13.19	10.77	20.17	11.91	15.97	18.67	17.33
Ground truth	7.74	8.68	9 .33	12.81	9.54	19.74	11.31	13.41	15.06	14.50

Table 2. Comparison of word error rates for different clustering methods.

ers in real acoustic environments with actual microphone arrays.

In this work, we have described our system for the 2014 REVERB Challenge and presented our results. On the REAL RC evaluation data, our system obtained a word error rate of 39.9% with a single channel of the array, and 18.7% with the best beamformed signal. In a contrast system using the true speaker identities, we obtained an error rate of 14.5%. We look forward to 2015 and beyond.

Acknowledgment

The authors are grateful to James Glass of Massachusetts Institute of Technology for his support and help, to Bhiksha Raj and Rita Singh of Carnegie Mellon University for their support and encouragement in the course of this work.

8. REFERENCES

[1] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Häb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, S. Gannot, and B. Raj, "The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, October 2013.

[2] M. Wölfel and J. McDonough, *Distant Speech Recognition*. London: Wiley, 2009.

[3] I. Himawan, I. McCowan, and M. Lincoln, "Microphone array beamforming approach to blind speech separation," in *Proc. of MLMI*, 2007, pp. 295–305.

[4] E. Zwyssig, M. Lincoln, and S. Renals, "A digital microphone array for distant speech recognition," in *Proc. of ICASSP*, 2010, pp. 5106–5109.

[5] I. Himawan, I. McCowan, and S. Sridharan, "Clustered blind beamforming from ad-hoc microphone arrays," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, pp. 661–676, 2011.

[6] K. Kumatani, T. Arakawa, K. Yamamoto, J. McDonough, B. Raj, R. Singh, and I. Tashev, "Microphone array processing for distant speech recognition: Towards real-world deployment," in *Proc. APSIPA Conference*, Hollywood, CA, December 2012.

[7] J. McDonough, K. Kumatani, and B. Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Processing Magazine*, vol. 29, pp. 127–140, November 2012.

[8] T. Virtanen, R. Singh, and B. Raj, Eds., *Techniques for Noise Robustness in Automatic Speech Recognition*. New York, NY: Wiley, 2012.

[9] M. Lincoln, I. McCowan, I. Vepa, and H. K. Maganti, "The multi–channel Wall Street Journal audio visual corpus (MC–WSJ–AV): Specification and initial experiments," in *Proc. of ASRU*, 2005, pp. 357–362.

[10] J. McDonough, K. Kumatani, T. Gehrig, E. Stoimenov, U. Mayer, S. Schacht, M. Wölfel, and D. Klakow, "To separate speech!: A system for recognizing simultaneous speech," *Proc. of MLMI*, 2008.

[11] K. Kumatani, J. McDonough, D. Klakow, P. N. Garner, and W. Li, "Adaptive beamforming with a maximum negentropy criterion," in *Proc. HSCMA*, Trento, Italy, May 2008.

[12] ——, "Adaptive beamforming with a maximum negentropy criterion," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 17, pp. 994–1008, July 2009.

[13] G. C. Carter, "Time delay estimation for passive sonar signal processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 463–469, 1981.

[14] M. Omologo and P. Svaizer, "Acoustic event localization using a crosspower–spectrum phase based technique," in *Proc. of ICASSP*, vol. II, 1994, pp. 273–6.

[15] J. H. DiBiase, H. F. Silverman, and M. S. Brandstein, "Robust localization in reverberant rooms," in *Microphone Arrays*, M. Brandstein and D. Ward, Eds. Heidelberg, Germany: Springer Verlag, 2001, ch. 4.

[16] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice–Hall, 1993.

[17] U. Klee, T. Gehrig, and J. McDonough, "Kalman filters for time delay of arrival–based source localization," *Journal of Advanced Signal Processing, Special Issue on Multi–Channel Speech Processing*, August 2005.

[18] G. Lathoud, J.-M. Odobez, and D. Gatica-Perez, "AV16.3: an audio-visual corpus for speaker localization and tracking," in *Proceedings of the MLMI'04 Workshop*, 2004.

[19] K. Kumatani, L. Lu, J. McDonough, A. Ghoshal, and D. Klakow, "Maximum negentropy beamforming with superdirectivity," in *European Signal Processing Conference (EUSIPCO)*, Aalborg, Denmark, 2010.

[20] K. Kumatani, J. McDonough, B. Rauch, and D. Klakow, "Maximum negentropy beamforming using complex generalized gaussian distribution model," in *Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Pacific Grove, CA, USA, 2010.

[21] H. L. Van Trees, *Optimum Array Processing*. New York: Wiley, 2002.

[22] R. Zelinski, "A microphone array with adaptive post-filtering for noise reduction in reverberant rooms," in *Proc. ICASSP*, New York, NY, USA, April 1988.

[23] K. U. Simmer, J. Bitzer, and C. Marro, "Post-filtering techniques," in *Microphone Arrays*, M. Branstein and D. Ward, Eds. Heidelberg: Springer, 2001, pp. 39–60.

[24] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[25] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[26] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[27] M. Wölfel and J. McDonough, "Minimum variance distortionless response spectral estimation, review and refinements," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 117–126, Sept. 2005.

[28] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "WSJCAMO: a British English speech corpus for large vocabulary continuous speech recognition," in *Proc. ICASSP*, 1995.

[29] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. P. V. Valtchev, and P. C. Woodland, *The HTK Book*, 3rd ed. Cambridge University Engineering Department, 2006.

[30] L. Uebel and P. Woodland, "Improvements in linear transform based speaker adaptation," in *Proc. of ICASSP*, 2001.

[31] L. Welling, H. Ney, and S. Kanthak, "Speaker adaptive modeling by vocal tract normalization," *IEEE Trans. on SAP*, vol. 10, no. 6, pp. 415–426, Sep. 2002.

[32] M. J. F. Gales, "The generation and use of regression class trees for MLLR adaptation," Cambridge University, Tech. Rep. CUED/F–INFENG/TR263, 1996.

[33] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, April 1995.

[34] J. McDonough and E. Stoimenov, "An algorithm for fast composition with weighted finite–state transducers," in *Proc. of ASRU*, Kyoto, Japan, 2007.

[35] M. Mohri, F. Pereira, and M. Riley, "Weighted finite–state transducers in speech recognition," *Jour. on CSL*, vol. 16, pp. 69–88, 2002.

[36] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.