ROBUST BLOCK-BASED CLUSTERING AND IDENTIFICATION OF AUTOREGRESSIVE SPEECH PARAMETERS BASED ON DYNAMIC STATE TRACKING

Ruofei Chen and Cheung-Fat Chan

Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong
ruofechen2@student.cityu.edu.hk, itcfchan@cityu.edu.hk

ABSTRACT

In this paper, we propose two block-based clustering and identification algorithms that contribute to robust estimation of autoregressive (AR) speech parameters in noisy environments. Motivated by the fact that the evolution pattern of speech dynamics could be an observable feature that are retained in a series of noisy observations, a dynamic state tracking scheme based on Kalman filter is incorporated to utilize this additional trajectory information in block-based AR codebook design. The proposed algorithm is devised in a sense that AR blocks with similar clean line spectrum frequency trajectories as well as noisy-to-clean mappings are clustered offline and identified online. It is compared with conventional vector quantization based approaches that directly minimize a distortion between AR parameters. Through objective assessments based on mean square error and log-spectral distance, it is demonstrated that the proposed algorithm achieves significant improvement over conventional methods in various conditions.

Index Terms— autoregressive model, vector quantization, clustering, Kalman filter

1. INTRODUCTION

Autoregressive (AR) model of speech has been well explored in various speech applications for its compact representation and its frequency-domain interpretation. In speech enhancement, techniques such as Wiener filtering and Kalman filtering attempt to estimate clean AR parameters of speech from noisy observations in different perspectives. In iterative Wiener filtering (IWF) such as [5][6] , clean AR parameters are sequentially estimated from noisy speech using maximum *a posteriori* (MAP) techniques. While in Kalman filtering applications [2][4], speech and/or noise are modeled as stochastic AR process, and AR parameters are represented in state-space form to model the state transition between time samples. Parameter estimation and iterative update can be achieved by expectation-maximization (EM) type algorithms [2]. One advantage is that, as the order of AR parameters is much lower than that of its corresponding time and frequency samples in each analysis frame, the estimation is performed with reduced dimensions. However, they generally suffer poor performance in adverse (e.g. low signal-to-noise ratio (SNR) and/or non-stationary) environments, albeit various constraints can be imposed on these estimators. It is because in such cases, without any prior information available in these methods, it is extremely difficult to retrieve clean AR parameters in noise-dominated and/or fast-varying observations. To address this fundamental limitation, a codebook driven approach is employed in [7] to estimate the clean AR parameters based on noisy observations in a maximum likelihood (ML) sense. In this approach, clean speech and noise AR parameters are offline trained and online identified by searching and combining the entries in codebooks. Given that the correct codebook entry is accessed, parameter estimation can be substantially improved as prior information of clean speech is incorporated. However, the identification process also becomes fairly difficult in adverse environments as very limited speech information can be observed in noisy AR parameters and the noisy-to-clean mapping would be a nearly one-to-many mapping. In addition, without inter-frame constraint, the identification is performed independently on a frame basis. Hence fluctuation is observed in envelope spectra constructed by codebook-derived AR parameters. Therefore, to improve the frame-based codebook approach, it is desirable to incorporate additional past information in codebook clustering and identification.

In [1], a block concept is introduced, and series of line spectrum frequencies (LSFs) are adopted as a variant of AR codebook. Subsequently, clean AR parameters are estimated through tracking its temporal trajectories using Kalman filtering. This block codebook approach is attractive for two reasons. First, temporal correlation between adjacent clean frames is taken into account in the codebook design. In doing so, various patterns of speech evolution are clustered and stored as additional prior information. Second, smoothed estimate of clean AR parameters of current frame is determined based on the entire block of observations. As a result, more smooth trajectories and less fluctuation is observed in the enhanced envelope spectrogram, as compared to that of frame-based codebook estimation. In this paper, several problems of the clustering and identification strategies used in [1] are discussed, and two robust clustering and identification algorithms that contribute to better estimation of clean AR parameters within the block codebook framework are proposed accordingly. They are compared and evaluated in objective measures, and experimental results show the improvement of the proposed algorithms over previous approaches.

The remainder of this paper is organized as follows. In Section 2, the original algorithm in [1] is briefly described and two algorithms are proposed to improve the AR parameter estimation. In Section 3, the performance of the proposed algorithms is evaluated. Finally, conclusion is drawn in Section 4.

2. ALGORITHM DEVELOPMENT

It is shown in [1] that, owing to the long-term tracking scheme, improved AR parameter estimation can be achieved with correctly identified Kalman system parameters. Therefore, in the offline training stage, it is desirable to find affordable sets of Kalman system parameters that best characterizes all the mappings between noisy and clean LSFs and all the state transitions between consecutive clean LSFs. While in online enhancement stage, the objective is to identify the optimal set (by means of certain distortion measures) of

pre-trained Kalman system parameters with only noisy observations available.

In [1], during clustering, the averaged noisy LSF vector $\bar{\mathbf{y}}$ (calculated from blocks of normalized autocorrelation coefficients) is defined as the block feature representation. Split vector quantization (VQ) are performed on $\bar{\mathbf{y}}$, and the distortion measure is defined as the log-spectral distance (LSD) as

$$d(\bar{\mathbf{w}}, \hat{\mathbf{w}}_j) = (\bar{\mathbf{w}} - \hat{\mathbf{w}}_j)^T(\bar{\mathbf{w}} - \hat{\mathbf{w}}_j) \quad (1)$$

where $\bar{\mathbf{w}}$ and $\hat{\mathbf{w}}_j$ are log magnitude spectra constructed by $\bar{\mathbf{y}}$ and current j^{th} codeword LSF vector $\hat{\mathbf{y}}_j$, respectively. During each iteration, associated blocks of parallel (both noisy and clean) LSFs (denoted as $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_K]$ and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_K]$, where P is the linear prediction order, and K is the number of frames per block) are regrouped based on the minimum distance rule as

$$j_{\min} = \arg\min_j \{d(\bar{\mathbf{w}}, \hat{\mathbf{w}}_j)\} \quad (2)$$

Assume that a total of L LSF blocks are grouped into the j^{th} cluster. The centroid $\hat{\mathbf{w}}_j$ is calculated by simply averaging out all $\bar{\mathbf{y}}_i$ within the cluster as

$$\hat{\mathbf{y}}_j = \sum_{i=1}^{L} \bar{\mathbf{y}}_i / L; \quad (3)$$

One iteratively update (1)-(3) until convergence criteria meets. A total of J clusters are formed and their associated parallel (both noisy and clean) LSF blocks are applied in ML learning. Consequently, J centroids of averaged LSF vectors and their corresponding sets of Kalman system parameters $\mathbf{\Theta} = \{\mathbf{F}, \mathbf{H}, \mathbf{Q}, \mathbf{R}, \hat{\mathbf{x}}_1, \mathbf{\Sigma}_1\}$ (where $\mathbf{F}$ is the state transition matrix, $\mathbf{H}$ is the linear mapping matrix for state and observation, $\mathbf{Q}$ and $\mathbf{R}$ are their prediction error covariances, $\hat{\mathbf{x}}_1$ and $\mathbf{\Sigma}_1$ are the initial state and its error covariance, respectively) are stored as $a\ priori$ information in the codebook.

However, there are several limitations of the above algorithm. First, the implementation of LSD is computationally expensive. Second, intuitively, the centroid should be derived by minimizing the distortion measure defined in (1), but it is difficult to compute the LSF centroid from the LSD measure. More importantly, the trajectory information within noisy blocks might already be smoothed out during clustering using this averaged feature representation. For instance, two blocks of LSFs with rising and falling formant frequency trajectories might be misclassified in a single cluster as the averaged LSFs are close, despite that their true state transition patterns are entirely different. As a consequence, it will deteriorate the offline ML learning, and subsequently the online estimation process.

2.1. Block-based Matrix Quantization (MQ) Clustering (Algorithm A)

The most intuitive way to tackle the above problems is to extend the VQ to a MQ so that all noisy vectors within a block contribute to the total distortion measure. In addition, as the computation of distortion has been raised by a factor of K, computational affordable distortion measure is desired to replace the original LSD measure. Theoretical analysis of linear predictive coding (LPC) parameters in [3] shows that LSD can be reformulated as an approximate quadratic measure between LSFs as

$$d(\mathbf{Y}, \hat{\mathbf{Y}}_j) = \sum_{k=1}^{K} (\mathbf{y}_k - \hat{\mathbf{y}}_{j,k})^T \mathbf{W}_k (\mathbf{y}_k - \hat{\mathbf{y}}_{j,k}) \quad (4)$$

where $\mathbf{W}_k = \mathbf{J}_k^T \mathbf{R}_k \mathbf{J}_k$ is the sensitivity matrix with $\mathbf{J}_k$ being the Jacobian matrix transforming LSFs to direct LPC coefficients and $\mathbf{R}_k$ being the autocorrelation matrix. There are two reasons to adopt the LSF form rather than the direct LPC form. First, diagonalized sensitivity matrix indicates scale quantization of LSFs does not affect each other, and hence results in less quantization error. Second, the weighted mean square error (WMSE) is easy to compute compared to general quadratic measure. As a result, LSF blocks are regrouped based on the minimum distance rule for the distortion measure in 4.

For a cluster with L blocks, the block centroid $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_K]$ is then obtained by sequentially minimizing

$$\sum_{i=1}^{L} (\mathbf{y}_{i,k} - \hat{\mathbf{y}}_k)^T \mathbf{W}_{i,k} (\mathbf{y}_{i,k} - \hat{\mathbf{y}}_k) \quad (5)$$

which results in

$$\hat{\mathbf{y}}_k = (\sum_{i=1}^{L} \mathbf{W}_{i,k})^{-1} (\sum_{i=1}^{L} \mathbf{W}_{i,k} \mathbf{y}_{i,k}) \quad (6)$$

In this case, one iteratively update (4)-(6) and subsequently store a total of J block centroids $\hat{\mathbf{Y}}$ and corresponding Kalman system parameters $\mathbf{\Theta}$. This algorithm can be regarded as a straightforward extension of the algorithm used in [1]. However, the physical meaning of this algorithm accounts for its distinct advantage over the original one. It sequentially compares all the frames (including the trajectory information) within the block in both clustering and identification, so noisy blocks will be grouped together as long as all their frames match well. As such, misclassifications caused by the averaging effect are avoided. It is also worth mentioning that the additional computational cost required for a online block codebook searching can be removed as the distortion for previous frames in this block has been calculated in previous blocks and hence can be reused with a memory system.

2.2. Block-based Clustering with Dynamic State Tracking (Algorithm B)

In the previous algorithm, noisy LSF blocks are clustered based on the weighted distance measured directly from the LSF matrices. The corresponding Kalman system parameters are derived based on the intuition that similar (in WMSE sense) noisy LSF matrices lead similar noisy-to-clean mappings and also similar underlying state transitions. However, due to the fast-varying nature of both speech and noise, this statement is not always true. In this subsection, a hybrid split MQ and EM-type algorithm is proposed to utilize the feedback of Kalman filter output in iterative clustering. In the first stage of the proposed algorithm, noisy block centroid of the entire training set is calculated using (6). A small offset is then added to split it into two centroids. Parallel blocks are grouped into either cluster based on the rule in (4). The corresponding Kalman system parameters are learned and attached to each cluster. In the next stage, an EM-type algorithm that minimizes the distortion between true clean LSFs and the output of Kalman filtering is employed. In the E-step, for each noisy observation block $\mathbf{Y}$ in the training set, the expected value of current clean LSF estimate $\tilde{\mathbf{x}}_j$ in the j^{th} codebook entry is obtained by running a Kalman smoother for each set of system parameters $\mathbf{\Theta}_j^i$ (at i^{th} iteration), which is given by

$$\tilde{\mathbf{x}}_j = f_{\mathbf{\Theta}_j^i}(\mathbf{Y}) \quad (7)$$

where f denotes the Kalman smoother function with a set of recursion equations. The regrouping of parallel training data $\mathbf{X}$ and $\mathbf{Y}$

are achieved by applying the minimum distance rule on the distortion measure between true clean LSF vector $\mathbf{x}$ (last frame in current block) and the smoothed estimate $\tilde{\mathbf{x}}_j$, as defined by

$$d(\mathbf{x}, \tilde{\mathbf{x}}_j) = (\mathbf{x} - \tilde{\mathbf{x}}_j)^T (\mathbf{x} - \tilde{\mathbf{x}}_j) \tag{8}$$

The M-step is same as the ML estimation proposed in [1]. In doing so, new sets of Kalman system parameters $\mathbf{\Theta}_j^{i+1}$ are obtained by minimizing the total negative log-likelihood of parallel LSF blocks inside the j^{th} cluster as

$$\mathbf{\Theta}_j^{i+1} = g(\mathbf{X}_j, \mathbf{Y}_j) \tag{9}$$

where g denotes the ML estimator used in [1]. One performs (7)-(9) to iteratively estimate sets of Kalman system parameters $\mathbf{\Theta}$ that better characterizes the noisy-to-clean mapping as well as the state transition of training blocks. The EM algorithm terminates when the total distortion D^i for all J clusters in (8) at i^{th} iteration does not vary much during consecutive iterations. The convergence criterion is defined as

$$\frac{D^i - D^{i-1}}{D^i} < \xi \tag{10}$$

where ξ is the pre-defined tolerance and

$$D^i = \sum_{j=1}^{J} d(\mathbf{x}, \tilde{\mathbf{x}}_j)^i \tag{11}$$

The noisy block centroids are recomputed using regrouped data. Each block centroid is further split into two and the EM process is applied again to optimize $\mathbf{\Theta}$ for current level. The splitting stops when the desired number of entries is reached. Implementation procedure of the proposed algorithm is described in Table 1.

Table 1: The proposed Algorithm B for block-based clustering

Initial condition: $J = 1$, parallel training set $\mathbf{X}$ and $\mathbf{Y}$
While ($J <$ Max Codebook Size)
1. Find block centroids $\hat{\mathbf{Y}}_{\{1,\ldots,J\}}$ using (6)
2. Split the centroids by adding small offsets $J = 2J$
3. Learn initial Kalman system parameters $\mathbf{\Theta}^0_{\{1,\ldots,J\}}$ using (9)
While (Fractional change of $D < \xi$)
4. Obtain smoothed estimate $\tilde{\mathbf{x}}_{\{1,\ldots,J\}}$ with $\mathbf{\Theta}^i$ using (7)
5. Regroup parallel blocks using (8)
6. Learn new $\mathbf{\Theta}^{i+1}_{\{1,\ldots,J\}}$ using (9)
7. Compute the total distortion D using (11)
End
End

In the proposed clustering algorithm, optimal sets of Kalman system parameters are obtained in a sense that the MSE error between the desired clean LSF vector and the output of Kalman filter is minimized. The convergence behavior of this algorithm is shown in Fig.1. Fig.1(a) plots the averaged MSE distortion for each LSF coefficient in radians over EM iterations. While Fig.1(b) plots the percentage of membership jump during regroupings. It is evaluated in (SNR=10dB) white noise environments. J is the codebook size. The initial distortion error is estimated by Algorithm A. This algorithm converges as the total distortion and percentage of membership jumps monotonically decreases during each iteration. It is observed that, with sufficient number of codebook entries employed, significant improvement over MQ clustering (Algorithm A) in AR estimation is achieved. In addition, the large performance gain observed in the first iteration demonstrates the effectiveness of employment of (8) instead of (4) as the distortion measure.

However, note that the distortion measure in (8) is not accessible in online enhancement stage as true clean LSF vector is missing. Hence, in practice, an approximate distortion measure is defined and adopted online as

$$d(\mathbf{y}, \mathbf{H}_j \tilde{\mathbf{x}}_j) = (\mathbf{y} - \mathbf{H}_j \tilde{\mathbf{x}}_j)^T (\mathbf{y} - \mathbf{H}_j \tilde{\mathbf{x}}_j) \tag{12}$$

The effect of this mismatch in clustering and identification are evaluated with different SNR settings in the next section. As a consequence, for each analysis block in online adaptation, a full codebook search is performed by applying Kalman filtering on noisy observations with each set of optimized system parameters. The entry is identified by finding the index with minimum distance defined in (12).

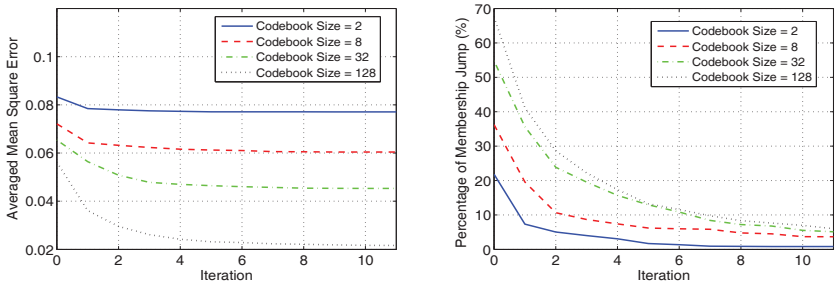
3. EXPERIMENTAL RESULTS

The proposed Algorithm A (MQ) and Algorithm B (MQ + EM) are compared with the original one (VQ) proposed in [1]. The distortion between noisy and true clean features is adopted as the benchmarks. The experiment settings are aligned with those in [1]. Clean speech and noise are taken from IEEE sentence database and NOISEX-92 database, respectively. Clean speech is manually corrupted by additive noise at SNR level of 0, 5, and 10 dB. The total length of training data is approximately 40 minutes. Separate testing data (different from training, approximately 5 minutes) are adopted in performance assessments. Two types of noise, namely, white Gaussian noise and car interior noise are adopted. Both block shift and frame shift are 8ms. The frame size and the length of Fourier transform are 256. The order of LPC analysis is 18. The number of frames per block is 13.

The MSE results of the proposed algorithms with various codebook size and SNR settings in white noise environments are illustrated in Fig.2. It is observed that the performance gain of Algorithm A over the original is relatively constant and small in various conditions. The gap is relatively large in low SNR conditions, which indicates that the trajectory information is more important for identification in adverse conditions. The performance gain of Algorithm B grows significantly as the codebook size increases. The theoretical upper bound of Algorithm B (assume (8) is available in identification) is plotted in dashed line while the achievable (as (12) is used in identification) is plotted in solid line. It is noticed that the gap between the two reduces as the SNR increases. It indicates that this identification mismatch is more severe in low SNR conditions. Overall speaking, realizable Algorithm B still achieves significant improvement over VQ/MQ-based algorithms in various conditions. To correlate the improvement in AR parameters with the improvement in spectral magnitude, the LSD measure is conducted as it compares the difference between log-scale magnitude spectra of noisy and enhanced speech. It is defined as

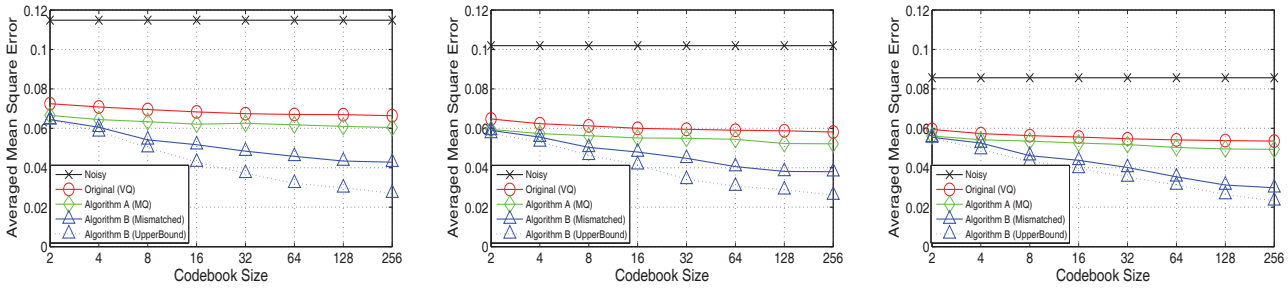
$$\mathrm{LSD}(S(\omega), \hat{S}(\omega)) = \sqrt{\frac{1}{2\pi} \int_{-\pi}^{\pi} \left[10 \log_{10} \frac{S(\omega)}{\hat{S}(\omega)} \right]^2 d\omega} \tag{13}$$

where $S(\omega)$ and $\hat{S}(\omega)$ represent the spectral shape (with unity gain) in this evaluation. The LSD improvement over the distortion between noisy and clean pairs are summarized in Table 2. It is observed that performance gain in spectral envelope is achieved using proposed algorithms in both noisy environments with various SNR settings. The LSD results are consistent with the MSE results in



(a) Total distortion versus iteration (b) Membership jump versus iteration

Fig. 1: Convergence analysis for Algorithm B



(a) SNR=0dB (b) SNR=5dB (c) SNR=10dB

Fig. 2: Comparison of various algorithms in white noise environments with different SNR settings

LSFs, and it can be concluded that Algorithm B performs best and Algorithm A performs a bit better than the original averaged VQ approach.

Table 2: Objective evaluation results of various block-based algorithms

Noise Type	Method	LSD Improvement(in dB) Input SNR 0dB	5dB	10dB
Gaussian White Noise	Original(VQ)	8.54	7.56	6.89
	Algorithm A(MQ)	8.87	7.81	7.02
	Algorithm B(MQ+EM)	10.85	9.57	8.64
Car Interior Noise	Original(VQ)	7.72	6.97	6.54
	Algorithm A(MQ)	8.08	7.65	6.93
	Algorithm B(MQ+EM)	9.21	8.54	8.01

4. CONCLUSION

Two block-based algorithms are proposed to improve the AR parameter estimation by incorporating additional trajectory information in memory-based AR codebook design. The enhanced estimates can be used in any speech applications that require clean AR parameter estimation. The effectiveness of the proposed algorithms is demonstrated through objective measures such as MSE and LSD.

5. REFERENCES

[1] R. Chen and C. F. Chan, "Analysis-synthesis based speech enhancement with improved spectrum envelope estimation by tracking speech dynamics," *Proceedings of the ICASSP2011*, pp. 4644–4647, May 2011.

[2] S. Gannot, D. Burshtein, and E. Weinstein, "Iterative and sequential kalman filter-based speech enhancement algorithms," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 4, pp. 373–385, Jul. 1998.

[3] W. Gardner and B. Rao, "Theoretical analysis of the high-rate vector quantization of LPC parameters," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, pp. 367–381, Sep. 1995.

[4] J. D. Gibson, B. Koo, and S. Gray, "Filtering of colored noise for speech enhancement and coding," *IEEE Transactions on Signal Processing*, vol. 39, no. 8, pp. 1732–1742, Aug. 1991.

[5] J. Hansen and M. Clements, "Constrained iterative speech enhancement with application to speech recognition," *IEEE Transactions on Signal Processing*, vol. 39, no. 4, pp. 795–805, Apr. 1991.

[6] J. Lim and A. Oppenheim, "All-pole modeling of degraded speech," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 3, pp. 197–210, Jun. 1978.

[7] S. Srinivasan, J. Samuelsson, and W. B. Kleijn, "Codebook driven short-term predictor parameter estimation for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 163–176, 2006.