行政院國家科學委員會專題研究計畫成果報告

無線通訊環境下以語音擷取網路資訊相關技術之研究 (I) –

子計畫－：無線通訊環境下國語中文之聲學及語言處理基礎技術之研究

計畫編號：NSC 89-2213-E-002-176
執行期限：89 年 8 月 1 日至 90 年 7 月 31 日
主持人：李琳山　國立台灣大學資訊工程學系
E-mail: lsl@speech.ee.ntu.edu.tw

Abstract

The Linear Discriminant Analysis (LDA) has been widely used to derive the data-driven temporal filtering of speech feature vectors. In this report, we proposed that the Principal Component Analysis (PCA) can also be used in the optimization process just as LDA to obtain the temporal filters, and detailed comparative analysis between these two approaches are presented and discussed. It's found that the PCA-derived temporal filters significantly improve the recognition performance of the original MFCC features as LDA-derived filters do. Also, while PCA/LDA filters are combined with the conventional temporal filters, RASTA or CMS, the recognition performance will be further improved regardless the training and testing environments are matched or mismatched, compressed or noise corrupted.

Keyword: Speech Recognition, Robustness, Additive Noise, Temporal Filters, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA)

1. Introduction

One of the most challenging problems in automatic speech recognition (ASR) is to derive a robust speech feature representation for speech signals so that it is less sensitive to the various corrupting acoustic conditions, such as additive noise and channel distortion. The Cepstral Mean Subtraction (CMS) [1] and the Relative Spectral (RASTA) [2] techniques are typical examples in performing filtering on the time trajectories of speech features in order to alleviate harmful effects of various distortions and corruptions. Such processing approaches have been widely proved to be able to improve the performance of the ASR systems efficiently.

The RASTA approach tries to filter out relatively slow and relatively fast changes in the trajectories of a critical logarithmic short-time spectral component of speech [2,4]. The initial form of the RASTA filter was originally optimized on a relatively small series of ASR experiments with noisy telephone digits, and there was no guarantee that these solutions are also optimal to other ASR tasks and environments. It is therefore desirable to obtain optimal sets of time filtering coefficients for a specific given ASR task and environment, which have to be obtained data-driven according to some optimization criterion. Linear Discriminant Analysis (LDA) has been widely applied [3,4,5] in such approaches in the optimization process to yield the time trajectory filters. Since LDA is a stochastic technique that optimizes linear discriminability between classes, and therefore the speech features must be labeled into different classes before the LDA is performed. Such data-driven LDA-derived temporal filters were reported to yield better recognition performance than the conventional RASTA filters [3].

In this report, the Principal Component Analysis (PCA) instead of LDA is used in the optimization process in obtaining these temporal filters, and comparative analysis between PCA and LDA approaches is presented. It will be shown that the PCA-derived temporal filters have quite different frequency responses from those of either the CMS, original RASTA or the data-driven LDA-derived filters. Experimental results also showed that the data-driven approaches, including both the PCA and LDA methods, not only outperform the conventional CMS or RASTA approaches in most cases, but also can be properly combined with these conventional approaches to give a better recognition accuracy. On the other hand, the proposed data-driven PCA approach, though significantly easier to implement than LDA, gives a comparable performance as LDA does and sometimes better in recognition.

The remainder of the report is organized into 5 sections. In section 2, the approach to derive the data-driven temporal filters using the Principal Component Analysis (PCA) is proposed. Then section 3 introduces the experimental setup and shows the frequency responses of the resulted temporal filters, as well as the comparison with the LDA, RASTA and CMS filters. In section 4, experimental results are presented and discussed. Section 5 briefly compares the PCA and LDA temporal filters. Finally, a short conclusion is given in section 6.

2. Temporal Filter Design Using Principal Component Analysis (PCA)

Given an ordered sequence of K-dimensional feature vectors $\boldsymbol{x}(n)$, n=1, ……,N,

$$\boldsymbol{x}(n)=[x(n,1)\ x(n,2),\ \ldots,\ x(n,k),\ \ldots,\ x(n,K)]^T, \quad (1)$$

where $x(n,k)$ is the k-th component of the feature vector at time n, then the k-th time trajectory of $\boldsymbol{x}(n)$ is the sequence [$x(1,k)$ $x(2,k)$ …… x(N,k)],

denoted as $y_k(m)$; m=1,2, …. N, (2)

where $y_k(n)=x(n,k)$. This is illustrated in Figure 1.

Now we'd like to design an L-point FIR-filter which is performed on the time trajectory $y_k(m)$, m=1…N, with PCA

technique. First, an L-point rectangular window is shifted along the sequence $y_k(m)$, m=1…N, to obtain the sequences of L-dimensional vectors $\mathbf{z}_k(n)$, n=1 … N-L+1, where

$$\begin{matrix}\begin{bmatrix} x(1,1) \\ x(1,2) \\ \mathrm{M} \\ x(1,k) \\ \mathrm{M} \\ x(1,K) \end{bmatrix} & \begin{bmatrix} x(2,1) \\ x(2,2) \\ \mathrm{M} \\ x(2,k) \\ \mathrm{M} \\ x(2,K) \end{bmatrix} & \begin{bmatrix} x(3,1) \\ x(3,2) \\ \mathrm{M} \\ x(3,k) \\ \mathrm{M} \\ x(3,K) \end{bmatrix} & \Lambda & \begin{bmatrix} x(n,1) \\ x(n,2) \\ \mathrm{M} \\ x(n,k) \\ \mathrm{M} \\ x(n,K) \end{bmatrix} & \Lambda & \begin{bmatrix} x(N,1) \\ x(N,2) \\ \mathrm{M} \\ x(N,k) \\ \mathrm{M} \\ x(N,K) \end{bmatrix} \\ \mathbf{x}(1) & \mathbf{x}(2) & \mathbf{x}(3) & \Lambda & \mathbf{x}(n) & \Lambda & \mathbf{x}(N) \end{matrix} \begin{matrix} \rightarrow y_1(m) \\ \rightarrow y_2(m) \\ \mathrm{M} \\ \rightarrow y_k(m) \\ \mathrm{M} \\ \rightarrow y_K(m) \end{matrix}$$

Figure 1. The representation of the time trajectories of feature sequences

$$\mathbf{z}_k(n)=[\, y_k(n)\; y_k(n+1)\; y_k(n+2)\; \ldots\ldots\; y_k(n+L\text{-}1)]^{\mathrm{T}}. \qquad (3)$$

So $\mathbf{z}_k(n)$ is the windowed vector of $y_k(m)$ started at the time index n, on which the L-point FIR filter is applied, as depicted in Figure 2. Then, these L-dimensional vectors $\mathbf{z}_k(n)$ are viewed as the samples of the random vector $\mathbf{z}_k$ and hence the mean and the covariance of $\mathbf{z}_k$ may be calculated,

$$\mu_{z_k} = \frac{1}{N-L+1}\sum_{n=1}^{N-L+1}\mathbf{z}_k(n) \qquad (4)$$

$$\Sigma_{z_k} = \frac{1}{N-L+1}\sum_{n=1}^{N-L+1}(\mathbf{z}_k(n)-\mu_{z_k})(\mathbf{z}_k(n)-\mu_{z_k})^T \qquad (5)$$

Figure 2. The procedure to extract the L-dimensional vectors $\mathbf{z}_k(n)$ from the time trajectory $y_k(n)$

Next, with the procedure of PCA, we calculate the first eigenvector $\mathbf{w}_k$ corresponding to the largest eigenvalue of the covariance Σ_{z_k}. The components of this eigenvector are then taken as the coefficients of the L-point filter, which maps these L-dimensional vectors $\mathbf{z}_k(n)$, n=*1* … *N-L+1*, onto a one-dimensional output space. According to PCA, the filter $\mathbf{w}_k$ is optimal because it maximizes the variance of the output sequence among all possible L-point filters. Such a PCA process above is carried out for each time trajectory $y_k(m)$, k=*1,2, …K*, thus yielding a separate FIR filter for each time trajectory.

In practice, the above is first performed on the original acoustic feature vectors of the training speech database to obtain the desired FIR filters. These filters are then applied to the time trajectories of the feature vectors of both training and testing database to obtain the new feature vectors. These new feature vectors are finally used for model training and testing following normal ASR procedures.

3. Experimental Setup

The speech database for the initial experiments included 8000 Mandarin digit strings produced by 50 male and 50 female speakers, taken from the database NUM-100A provided by the Association for Computational Linguistics and Chinese Language Processing at Taipei. The speech signal was recorded in normal laboratory environment at 8 kHz sampling rate and encoded with 16-bit linear PCM. The 8000 digit strings included 1000 each for 2, 3, 4, 5, 6 and 7-digit strings respectively plus 2000 single digit utterances. Among the 8000 Mandarin digital strings, 7520 were used in training, while the other 480 in testing. A 32ms Hamming window shifted with 16ms steps and a pre-emphasis factor of 0.95 were used to evaluate 15 mel-frequency cepstral coefficients (MFCCs). The time trajectories for the MFCC vectors in the training database were then processed by the PCA-derived FIR filters as described previously. The Length L of the FIR filter was preliminarily set to be 10.

Figure 3 shows the frequency responses of the 15 PCA-derived FIR filters. The 15 filters are very close, and the differences among them are almost unobservable in the figure, although they were derived from different trajectories of the original MFCC vectors. On the other hand, the 15 LDA-derived temporal filters were also constructed as described below for comparison. In the training database, the 7520 Mandarin digital strings were first segmented into 11 classes, i.e., the digits, 0-9, plus the silence. Then for each time trajectory of the feature vectors, the between-class matrix and the within-class matrix were calculated and used to obtain the LDA temporal filter coefficients [3]. The frequency responses of the resulted 15 LDA-derived FIR filters are shown in Figure 4. From Figures 3 and 4 we see that both the PCA-derived filters and LDA-derived filters don't attenuate the low modulation frequency components while having many sidelobes in their higher modulation frequency responses, which are significantly different from the frequency responses of the RASTA filter and CMS filter shown in Figure 5. Furthermore, the 15 LDA temporal filters, although similar in shape, are not very close to one another, and have higher sidelobes than the PCA temporal filters.

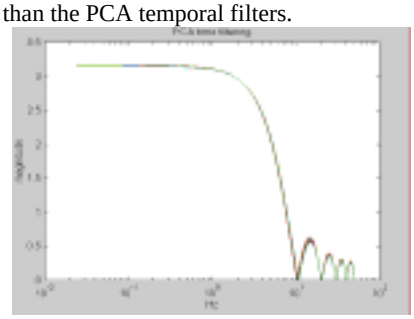

Figure 3. The frequency responses of the 15 PCA-derived temporal filters

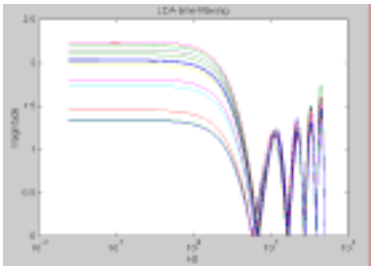

Figure 4. The frequency responses of the 15 LDA-derived temporal filters

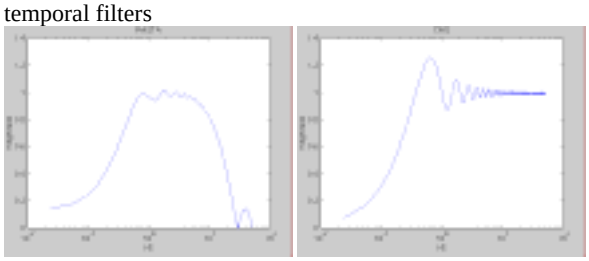

Figure 5. The frequency responses of the RASTA filter (left) and the CMS filter (right)

4. Experimental Results

The PCA-derived FIR filters were first individually performed on the time trajectories of the MFCC vectors for the 7520-string training database. The resulted 15-dimensional new features plus their time-derivatives are the components in the finally used 30-dimensional feature vectors. With these new feature vectors two versions of HMM's for each digit with 5 states were trained, with 2 and 4 mixtures per state respectively. Similarly, the LDA-, CMS- and RASTA-derived features from MFCC's along with their derivatives were used to train their individual HMM's for comparison. On the other hand, the 480 clean speech testing digit strings were processed by RealAudio compression algorithm and/or manually added with white noise at different levels to produce the compressed and/or noise corrupted speech data. The settings for the RealAudio compression data were "single rate" in file type, 6500 in bit rate, "6.5 Kbps Voice" for the codec and "56K Modem" for the target audience. These clean and compressed and/or noise corrupted speech data were first converted to MFCC's, then individually processed by the above four time filtering approaches to form four sets of feature vectors for testing.

Table 1 and Table 2 list the recognition results for 4 mixtures. Table 1 is for mismatched condition, i.e., the speech models were trained with clean speech, while the testing speech were compressed and/or noise corrupted. In this table, the left half shows the results for noise corrupted speech, and the right half shows the results for RealAudio compressed and noise corrupted speech. The upper half compares the results of the four filtering approaches: CMS, RASTA, LDA and PCA with those of plain MFCC (Here, MFCC denotes the 15-dimensional MFCC feature vector plus its 15-dimensional time-derivative vector). It can be found that for clean uncompressed speech, CMS, RASTA and LDA are slightly worse than MFCC, while PCA is better, as in the first column. When the speech is noise corrupted, PCA performs close to CMS, RASTA and LDA, sometimes better and sometimes slightly worse. However, when the speech is RealAudio compressed in the right half of Table 1, PCA performs the best for clean speech and better than CMS and RASTA for noisy speech, probably because the PCA processing somehow complements the RealAudio compression. LDA performs not very well for clean speech but obviously better for noisy speech. An interesting phenomenon is that, while comparing the results for the uncompressed and RealAudio compressed data at the same noise level, we note that the latter very often outperform the former when PCA, LDA or plain MFCC are used. Probably the RealAudio compression somehow alleviates the noise effect in the speech signal and thus the mismatch between the model and test data is reduced. Plain MFCC, PCA and LDA happened to take this advantage, but CMS and RASTA didn't. The lower half of the table shows the additivity for the four filtering approaches. It can be found that PCA plus CMS is always significantly better than CMS alone, and very often better than PCA alone as well. Similar situation occurs for PCA plus RASTA. But CMS plus RASTA performs always between CMS alone and RASTA alone. So PCA is additive to either CMS or RASTA, but CMS and RASTA are not additive to each other. Similarly, it can be also found that LDA is additive to CMS or RASTA under noisy conditions. To briefly sum up, PCA plus CMS almost gives the best performance while the SNR is medium (20dB) or high (clean, 30dB), while LDA plus RASTA performs the best while the SNR is low (10dB). Similar trends can be observed for 2 mixtures, but left out here for lack of space.

SNR / model	clean	30dB	20dB	10dB	RealAudio Compressed			
					clean	30dB	20dB	10dB
MFCC	92.63	78.99	53.25	22.22	87.45	74.55	56.94	25.16
CMS	92.00	77.72	58.72	30.11	88.20	74.09	53.83	20.43
RASTA	88.95	77.20	61.60	35.23	81.12	69.89	57.97	33.85
LDA	91.54	75.65	58.43	31.32	86.53	77.09	62.06	38.80
PCA	**94.19**	77.61	60.51	29.82	**92.69**	76.91	62.35	35.18
LDA+CMS	90.56	81.35	70.58	41.74	88.14	78.07	68.85	44.33
PCA+CMS	93.61	**82.67**	**71.96**	41.80	90.85	**81.17**	69.49	39.72
LDA+rasta	90.33	80.20	70.41	**45.54**	86.01	78.18	**69.89**	**47.55**
PCA+rasta	92.11	80.25	65.52	37.77	88.95	79.68	66.90	33.62
CMS+rasta	90.85	76.97	60.16	32.64	85.90	71.50	57.40	27.58

Table 1. The digit recognition rates for different versions of HMM's with 5 states and 4 mixtures per state under mismatched conditions

SNR / model	clean	30dB	20dB	10dB	RealAudio Compressed			
					clean	30dB	20dB	10dB
MFCC	92.86	**90.73**	85.90	81.52	87.45	82.15	75.13	64.88
CMS	92.00	87.05	83.42	79.80	88.20	81.23	74.96	61.72
RASTA	88.95	86.30	83.42	76.11	81.12	71.79	67.47	56.53
LDA	91.54	89.58	85.55	80.25	86.53	82.56	80.31	71.51
PCA	**94.19**	89.69	87.16	**82.38**	**92.69**	82.79	79.56	70.52
LDA+CMS	90.56	87.80	85.32	77.95	88.14	82.96	79.56	73.11
PCA+CMS	91.36	89.46	**87.33**	81.75	90.85	**85.03**	**81.23**	**75.65**
LDA+rasta	90.33	85.84	81.87	75.19	86.01	81.92	77.43	72.19
PCA+rasta	92.11	87.45	84.80	78.41	88.95	83.13	79.97	72.37
CMS+rasta	90.85	87.22	83.19	76.63	85.90	76.51	69.31	57.51

Table 2. The digit recognition rates for different versions of HMM's with 5 states and 4 mixtures per state under matched noisy conditions

Table 2 is for matched noisy condition, i.e., the speech models were trained with noisy speech at the same noise level as the testing speech, but the training speech was not RealAudio compressed. So all of the 7520 clean speech digit strings in the training database were first manually added with white noise at different levels to produce the noise corrupted speech data. Then they were converted to MFCC and individually processed by the above four time filtering approaches plus their combinations to form ten sets of training feature vectors. Therefore for each noisy environment, ten sets of HMM's were trained and used to recognize the ten sets of testing features under matched noisy condition. The arrangements of data in Table 2 are the same as those in Table 1. The matched noisy conditions provide the opportunities to observe the real discriminating capabilities, instead of the robustness with respect to noise, of the various features. The upper left half of the table shows PCA almost always provide feature parameters with significantly higher discriminating capabilities than MFCC, CMS, RASTA and LDA, i.e., PCA very often performs the best. CMS, RASTA and LDA, on the other hand, may slightly degrade the discriminating functions of MFCC, since they often perform slightly worse than MFCC. Also, by comparing the results listed in the left half and the right half of Table 2, we see that RealAudio compressed data always give worse results than the uncompressed data for all filtering approaches. However, such performance degradation

is smaller for PCA and LDA than for CMS and RASTA. The lower half of the table shows that PCA and LDA are not additive to either CMS or RASTA in terms of discriminating capabilities in general. Such additivity only happens when the testing data is RealAudio compressed. Similar trends can be observed for 2 mixtures as well.

In conclusion, the above results indicated that CMS, RASTA and LDA may slightly reduce the discriminating capabilities of MFCC features, but they can efficiently reduce the mismatch between training and testing environments at noisy conditions. However, the proposed PCA approach not only keeps or enhances the discriminating capabilities of MFCC, but can reduce the influence caused by mismatched conditions. It performs specially well with RealAudio compressed speech too. LDA performs almost as well as PCA, and sometimes better than PCA under noisy conditions, but worse for clean speech. Also, either PCA or LDA approach can be used together with the RASTA or CMS to further reduce the mismatch caused by compression or additive noise. This is probably because the frequency responses of PCA and LDA filters are quite different from those of CMS and RASTA, and thus PCA, LDA and CMS, RASTA more or less emphasizes different components of the time trajectories. However, since the frequency responses of RASTA and CMS filters are more similar especially in low frequency region, thus combining them won't be very helpful.

5. Further Comparison Between PCA and LDA Temporal Filtering

Here, we'd like to further compare the PCA/LDA-derived temporal filters. For the k-th time trajectory $y_k(m)$ of the features $\boldsymbol{x}(n)$, the LDA process maximizes the ratio

$$\boldsymbol{w}_k^T \boldsymbol{S}_B \boldsymbol{w}_k / \boldsymbol{w}_k^T \boldsymbol{S}_W \boldsymbol{w}_k \quad (6)$$

with respect to the filter coefficients $\boldsymbol{w}_k$, while the PCA process maximizes the value

$$\boldsymbol{w}_k^T (\boldsymbol{S}_B + \boldsymbol{S}_W) \boldsymbol{w}_k \quad (7)$$

with respect to the filter coefficients $\boldsymbol{w}_k$, where $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ are the between-class covariance and the within-class covariance of the the k-th time trajectory of the features $\boldsymbol{x}(n)$. These two optimization criteria seem inconsistent and quite different. For example, maximizing equation (7) may reduce the ratio in equation (6) since the resulted within-class variance $\boldsymbol{w}_k^T \boldsymbol{S}_W \boldsymbol{w}_k$ is also amplified. On the other hand, the goal of maximizing equation (6) is to reduce the within-class variance $\boldsymbol{w}_k^T \boldsymbol{S}_W \boldsymbol{w}_k$ and increase the between-class variance $\boldsymbol{w}_k^T \boldsymbol{S}_B \boldsymbol{w}_k$ simultaneously, thus making the models more discriminant. However, when we use the PCA-derived filter $\boldsymbol{w}_{k\text{-PCA}}$ to evaluate the desired ratio $(\boldsymbol{w}_k^T \boldsymbol{S}_B \boldsymbol{w}_k / \boldsymbol{w}_k^T \boldsymbol{S}_W \boldsymbol{w}_k)$ for LDA in equation (6), we found that such ratios are very close to those obtained by the optimal LDA-derived filter $\boldsymbol{w}_{k\text{-LDA}}$, slightly lower though, as depicted in Figure 6 for the 15 time trajectories. Thus the PCA-derived temporal filters actually offer very similar linear discriminability as LDA even under LDA's criterion, i.e., the ratio $(\boldsymbol{w}_k^T \boldsymbol{S}_B \boldsymbol{w}_k / \boldsymbol{w}_k^T \boldsymbol{S}_W \boldsymbol{w}_k)$, although the PCA filters are derived from a different criterion. On the other hand, if we assume all classes are Gaussian and evaluate the model discriminability of PCA and LDA approaches using a different criterion, the average KL2-distance between each pair of the 11 classes here for all time trajectories, the result is **13.86** for PCA and **13.60** for LDA, i.e., PCA is even slightly better than LDA under this criterion. Furthermore, for PCA process the training database doesn't need to be labeled into different classes in order to obtain the between-class/within-class covariance matrices, which is necessary for LDA. Therefore implementing PCA is usually much easier than LDA, which is a major advantage of PCA- derived filters.

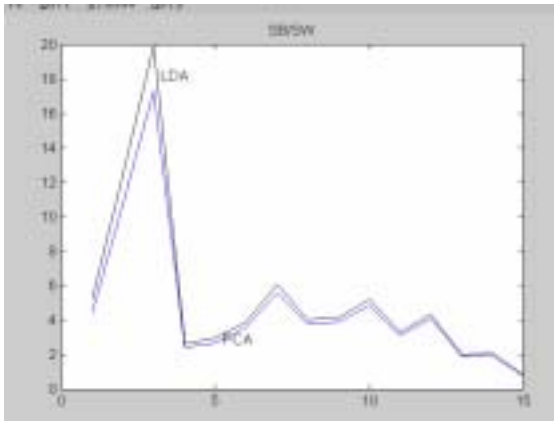

Figure 6 The ratio $(\boldsymbol{w}_k^T \boldsymbol{S}_B \boldsymbol{w}_k / \boldsymbol{w}_k^T \boldsymbol{S}_W \boldsymbol{w}_k)$ obtained by PCA-derived $\boldsymbol{w}_{k\text{-PCA}}$ filter LDA-derived filter $\boldsymbol{w}_{k\text{-LDA}}$ respectively for the 15 time trajectories of speech features

6. Conclusion and Future Work

In this report, we proposed a new temporal filtering approach using the principal component analysis (PCA), and comparative analysis with LDA-derived filtering approach is presented. Significant improvements in recognition accuracy under different conditions show the effectiveness of the PCA filtering approach. The PCA temporal filtering may efficiently alleviate the mismatch caused by noise corruption and/or RealAudio compression. It can also be easily combined with other temporal filtering methods to provide further improvements. When compared with LDA-derived filters, PCA is better under some conditions while LDA is better under other conditions. The two approaches actually complements each other.

7. References

[1] S. Furui, "Cepstral analysis technique for automatic speaker verification". IEEE Trans. Acoust. Speech Signal Process. 29:254-272, 1981

[2] H. Hermansky and N. Morgan, "RASTA processing of speech". IEEE Trans. Speech Audio Process. 2, 1994

[3] C. Avendano, S. van Vuuren and H. Hermansky, "Data Based Filter Design for RASTA-like Channel Normalization in ASR" ICSLP 96

[4] S. van Vuuren and H. Hermansky, "Data-driven Design of RASTA-like Filters", Eurospeech 97

[5] M. L. Shire, "Data-driven Modulation Filter Design under Adverse Acoustic Conditions and Using Phonetic And Syllabic Units", Eurospeech 99