Rapid Speaker Adaptation for Continuous Speech Recognition Using Merging Eigenvoices

Dong-jin Choi and Yung-Hwan Oh

Voice Interface Laboratory, CS Division, KAIST, Daejeon, Republic of Korea
{cdjin,yhoh}@speech.kaist.ac.kr

Abstract

Speaker adaptation in eigenvoice space is a popular method for rapid speaker adaptation. To improve the performance of the method and to obtain stabilized results, the number of speaker-dependent models should be increased and a greater number of eigenvoices should be re-estimated. However, the huge computation time required to find eigenvoices makes these solutions difficult, especially in a continuous speech recognition system. This paper describes a method to reduce computation time by estimating eigenvoices only for supplementary speaker-dependent models and merging them with the used eigenvoices. Experimental results show that the computation time is reduced by 73.7% while the performance is almost the same when the numbers of speaker-dependent models in two sets to be merged are the same.

1. Introduction

Generally, a speaker-dependent (SD) system outperforms a speaker-independent (SI) system when tested on the same speaker. Speaker adaptation aims to approach the performance of a SD system with as little data from the target speaker as possible. Maximum a posteriori (MAP) [1] and maximum likelihood linear regression (MLLR) [2] are well known speaker adaptation methods. However, these methods have a disadvantage in that the performance increases only slightly or even decreases with a very small amount of adaptation data [3]. Nowadays, people are concerned about rapid speaker adaptation, which is the technique of speaker adaptation using a small amount of data, around 30 sec. or less, since the range of applications which cannot request a long time speech for adaptation data are enlarged.

The eigenvoice technique [3],[4] is a popular rapid speaker adaptation method. This technique constrains the adapted model to be a linear combination of a small number of basis vectors (eigenvoices) obtained from a set of reference speakers, thereby reducing the number of free parameters to be estimated. One drawback of this technique is numerical problems. Principal component analysis (PCA) is too high a price to pay for rapid adaptation, especially in cases with large HMM systems like a continuous speech recognizer [5].

To improve performance of eigenvoice adaptation, a greater number of eigenvoices should be calculated. However, it is necessary to add sufficient SD models before estimating the eigenvoices because eigenvoices from insufficient SD models can yield unstable results [6]. A general eigenvoice adaptation method should combine used and supplementary SD models and calculate eigenvoices from the combined models to improve the performance. The computation time to calculate new eigenvoices using singular value decomposition (SVD) is increased in proportion to the number of combined SD models since the computation time of SVD is proportional to the amount of data. Time can be reduced if we calculate the eigenvoice for only complementary SD models and can use the used eigenvoices to calculate new eigenvoices.

In this paper, we describe a method to merge eigenvoices so that the computation time to add eigenvoices for performance improvement is decreased while word error rates of the systems that use and do not use merging eigenvoices are almost the same.

This paper is organised as follows. Speaker adaptation in eigenvoice space is reviewed in Section 2. Section 3 presents the method to merge eigenvoices. Section 4 evaluates the proposed method using the *Resource Management (RM)* corpus. Finally, we conclude this paper in Section 5.

2. Eigenvoice Technique

Suppose that we have R well-trained SD models and one SI model. The “supervector” is defined as $X_r = [(\mu_r^1)^T (\mu_r^2)^T \cdots (\mu_r^n)^T]^T$, in which μ_r^m is the mean vector of *m*-th Gaussian mixture of *r*-th SD model; *n* is the number of all Gaussian mixtures in *r*-th SD model. PCA is applied to the *R* supervectors, and *R* eigenvectors, $e(1), e(2), \cdots, e(R)$, are yielded. The mean supervector, $e(0)$, and the dominant *p* eigenvectors are called the “eigenvoices.” The supervector for the new speaker can be obtained by a linear combination of the eigenvoices, such that

$$X_t = e(0) + w(1)e(1) + \cdots + w(p)e(p) \qquad (1)$$

where $w(1), w(2), \cdots, w(p)$ can be calculated using maximum likelihood eigen-decomposition (MLED) [3].

The dimensions of supervectors for continuous speech recognition (CSR) are typically very large. For example, a general CSR system using 39 order feature parameters, 3 states, 6 mixtures, and triphone has about 300,000 order supervectors. Computing the covariance or correlation matrix of the eigenvoice in CSR is very difficult because of time and memory problems. Therefore, SVD is used generally in eigenvoice adaptation [6]. The SVD algorithm decomposes a matrix into two orthogonal matrices and a diagonal matrix:

$$X_{nN} - \mu(X)1 = U(X)_{nn}\Sigma(X)_{nN}V(X)_{NN}{}^{T} \quad (2)$$

where n is the dimension of the supervector and N is the number of SD models. $\mu(X)$ is the mean, 1 is a row N 1's, $U(\cdot)$ is an $(n \times n)$ matrix of eigenvectors, $\Sigma(\cdot)$ is an $(n \times N)$ matrix of spread values, and $V(\cdot)$ is an $(N \times N)$ matrix which contains information about the data projected into eigenspace.

It is often assumed that only p eigenvectors with large spread values are of interest. We can modify (2) as follows:

$$X_{nN} - \mu(X)1 \approx U(X)_{np}\Sigma(X)_{pp}V(X)_{Np}{}^{T} \quad (3)$$

The SVD is usually computed by a batch $O(4n^2N + 8nN^2 + 9N^3)$ time algorithm [7]. As mentioned above, because n is very large in CSR, the time required to calculate eigenvoice can be intolerable if N increases.

3. Merging Eigenvoices

There are many methods for merging eigenspace models, but they fail to handle a change in the mean. The mean should be updated in the merging process in CSR because the means of Gaussians are very important in speech recognition. Hall introduced a method for merging eigenspace models that explicitly and accurately keep track of the mean of the observations [8],[9].

Suppose that there are two sets of supervectors, X and Y. The SVD of these data are specified as

$$X_{nN} - \mu(X)1 \approx U(X)_{np}\Sigma(X)_{pp}V(X)_{Np}{}^{T} \quad (4)$$

$$Y_{nM} - \mu(Y)1 \approx U(Y)_{nq}\Sigma(Y)_{qq}V(Y)_{Mq}{}^{T} \quad (5)$$

in which N and M are the number of supervectors of X and Y and p and q are the number of eigenvectors to be used as eigenvoices with respect to X and Y.

Then we can specify the SVD eigenspace merged as

$$Z_{n(N+M)} - \mu(Z)1 \approx U(Z)_{ns}\Sigma(Z)_{ss}V(Z)_{(N+M)s}{}^{T} \quad (6)$$

in which s is the number of eigenvoices to use in the merged eigenspace.

3.1. Method of Merging Eigenvoices

Suppose that there is an orthonormal basis set, Γ_{ns}, that spans both eigenspace models of X and Y, and $\mu(X) - \mu(Y)$. We can then specify the required eigenvectors, $U(Z)_{ns}$, as a multiplication of Γ_{ns} and a rotation matrix, R_{ss} such that

$$U(Z)_{ns} = \Gamma_{ns}R_{ss} \quad (7)$$

To calculate Γ_{ns}, we can compute the residues, H_{nq}, of each of the eigenvoices in $U(Y)_{nq}$:

$$G_{pq} = U(X)_{np}{}^{T}U(Y)_{nq} \quad (8)$$

$$H_{nq} = U(Y)_{nq} - U(X)_{np}G_{pq} \quad (9)$$

The residue of $\mu(X) - \mu(Y)$, h_n, with respect to $U(X)_{np}$ is specified as

$$g_p = U(X)_{np}^{T}(\mu(X) - \mu(Y)) \quad (10)$$

$$h_n = (\mu(X) - \mu(Y)) - U(X)_{np}g_p \quad (11)$$

The H_{nq} is all orthogonal to each of eigenvoices in $U(X)_{np}$. However, some of the H_{nq} are zero vectors because such vectors represent the intersection of the two eigenspaces. These zero vectors should be removed in H_{nq}. The h_n also needs the removing process for the same reason.

$$v_{nt} = Orthobasis(\zeta[H_{nq}, h_n]) \quad (12)$$

where ζ is an operation that removes very small column vectors from the matrix and $Orthobasis(\cdot)$ is the function to compute a set of mutually orthogonal unit vectors that support its argument. t is the number of vectors from (12) and satisfies the following equation :

$$s = p + t \le p + q + 1 \le \min(n, M + N) \quad (13)$$

By (12), Γ_{ns} can be specified as

$$\Gamma_{ns} = [U(X)_{np}, v_{nt}] \quad (14)$$

Using (4), (5), (6), (7), and (14), we can specify the following equation:

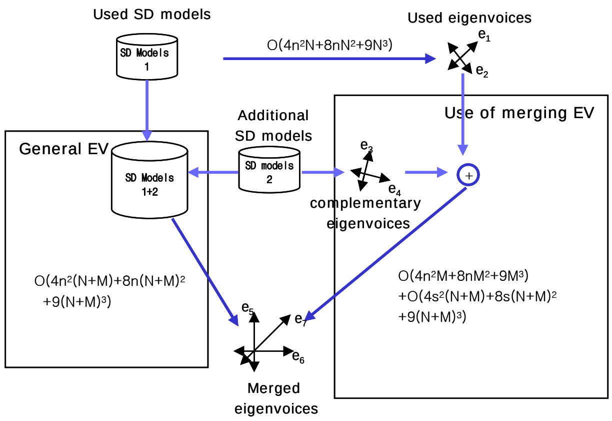

Figure 1: Comparison of general eigenvoice adaptation and the proposed method to use merging eigenvoices

$$
\begin{aligned}
Z_{n(N+M)} - \mu(Z)1 &\approx U(Z)_{ns}\Sigma(Z)_{ss}V(Z)_{(N+M)s}{}^T \\
&= [U(X)_{np}, v_{nt}]R_{ss}\Sigma(Z)_{ss}V(Z)_{(N+M)s}{}^T \\
&= [U(X)_{np}\Sigma(X)_{pp}V(X)_{Np}{}^T - \mu(Z)1 \\
&\quad , U(Y)_{nq}\Sigma(Y)_{qq}V(Y)_{Mq}{}^T - \mu(Z)1]
\end{aligned} \tag{15}
$$

Multiplying both sides by $[U(X)_{np}, v_{nt}]^T$, we obtain

$$
\begin{aligned}
&[U(X)_{np}, v_{nt}]^T[U(X)_{np}\Sigma(X)_{pp}V(X)_{Np}{}^T - \mu(Z)1 \\
&\qquad , U(Y)_{nq}\Sigma(Y)_{qq}V(Y)_{Mq}{}^T - \mu(Z)1] \\
&= R_{ss}\Sigma(Z)_{ss}V(Z)_{(N+M)s}{}^T
\end{aligned} \tag{16}
$$

Using SVD of the left side of (16), we can calculate R_{ss}. The time to calculate it is much shorter than in (6) since this SVD is an $s \times (N+M)$ problem and $s << n$.

Lastly, the merged eigenvoices, $U(Z)_{ns}$, can be obtained using (7).

3.2. Time Complexity

Fig. 1 shows the block diagram and the time complexity for the general eigenvoice adaptation method and the proposed method to use merging eigenvoices. Suppose that we used eigenvoices obtained from SD models 1 and SD models 2 is added to improve the performance. In the general eigenvoice adaptation method, we should join SD models 1 and 2 to make SD models 1+2 and use new eigenvoices to be calculated from SD models 1+2. In this process, the time complexity of SVD, which takes the most of the time is

$$O(4n^2(N+M)+8n(N+M)^2+9(N+M)^3) \tag{17}$$

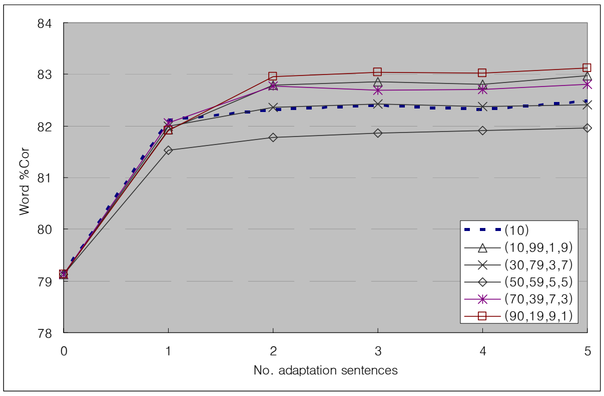

Figure 2: Word correction rate when using and not using merging eigenvoices and when *s*=10

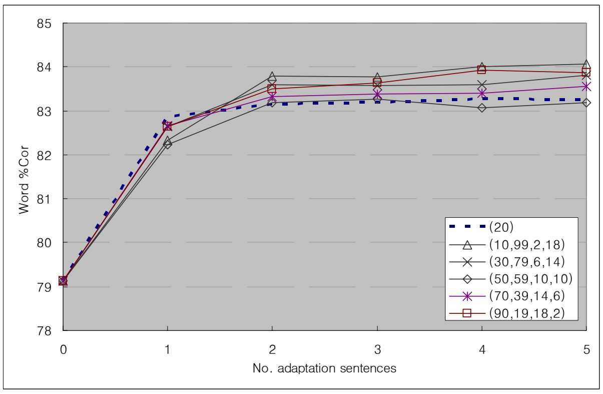

Figure 3: Word correction rate when using and not using merging eigenvoices and when s=20

In the proposed method, we should calculate eigenvoices from only SD model 2 and merge it with the used eigenvoices to obtain the new eigenvoices. The time complexity in this method is

$$
\begin{aligned}
&O(4n^2M + 8nM^2 + 9M^3 \\
&+ 4s^2(N+M) + 8s(N+M)^2 + 9(N+M)^3)
\end{aligned} \tag{18}
$$

Usually, *N*+*M* is much smaller than *n*, and *s* is smaller than *N*+*M* in the eigenvoice adaptation; therefore, we can ignore the time to merge eigenvoices. This means that it takes time to calculate eigenvoices from only SD models 2.

4. Experimental Results

We used the Resource Management (RM) corpus to evaluate the proposed method. The speech was parameterized into the 12 mel-frequency cepstral coefficients along with normalized log-energy and their first and second-order time derivatives. This yielded a 39-dimensional feature vector. Acoustic models are trained based on monophones, and each HMM state has one mixture components for the output distribution. The standard augmented 109-speaker SI training

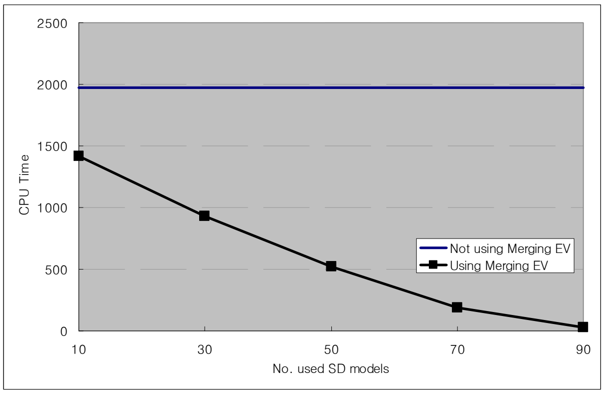

Figure 4: Computation time when using and not using merging eigenvoices

set of RM was used for building the SI models. A set of 109 SD models were generated by performing MLLR adaptation of these SI models with full regression matrices, followed by MAP adaptation. We used a 12-speaker SD training set for the adaptation phase and 1200 sentences from SD evaluation-test set for the recognition phase.

Word correction rates, both using merging eigenvoices and not using them, are compared in Figs. 2 and 3. We use 10 and 20 eigenvoices, respectively. Solid and dotted lines indicate the cases which use and do not use merging eigenvoices, respectively. In the key, the number next to the dotted line in parentheses is the number of merged eigenvoices. The numbers next to the solid lines are the number of the used SD models, the number of the supplementary SD models, the number of the used eigenvoices, and the number of the eigenvoices to be calculated from the supplementary SD models, in that order. The adaptation performance is almost the same regardless of the number of eigenvoices or SD models.

Fig. 4 shows the computation time in cases that use and do not use merging eigenvoices when a total of 109 combined SD models are used. We ran the same code several times and chose the smallest value to minimize the effect of other concurrently running processes. In the case that merging eigenvoices are not used, CPU time is constant since we should calculate eigenvoices from combined SD models regardless of the number of used SD models. However, in the case that merging eigenvoices are used, CPU time decreases along with the number of used SD models since we should calculate eigenvoices from only complementary SD models and merge them with the used eigenvoices

5. Conclusion

In this paper, a method for merging eigenvoices has been presented. Instead of calculating new eigenvoices from combined SD models to improve the performance of eigenvoice adaptation systems, we calculate the eigenvoices from only complementary SD models and merge them with used eigenvoices.

Experimental results show that there is only slight performance reduction due to computation error in the merging phase, while the computation time is remarkably reduced depending on the number of added SD models.

Future research should include experiments for more complex HMM systems. Further tests are needed with systems based on triphones or multiple Gaussian mixtures.

6. Acknowledgements

This work was supported (in part) by the Ministry of information & Communications, Korea, under the Information Technology Research Center (ITRC) Support Program.

7. References

[1] C. H. Lee, C. H. Lin, and B. H. Juang, "A study on speaker adaptation of the parameters of continuous density hidden Markov models", *IEEE Trans. Signal Processing*, vol. 39, pp. 806-814, 1991.

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

[3] R. Kuhn, J. Junqua, P. Nguyen, and N. Niedzielski, "Rapid Speaker Adaptation in Eigenvoice Space", *IEEE Tran. Speech and Audio Processing*, vol. 8, no. 6, pp. 695-707, 2000.

[4] R. Kuhn, P. Nguyen, J. Junqua, L. GoldWasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini, "Eigenvoices for speaker adaptation", *Proc. Int. Conf. Speech Language Processing*, vol. 5, pp. 1771-1774, 1998.

[5] P. C. Woodland, "Speaker Adaptation: Techniques and Challenges", *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 85-90, 2000.

[6] R. Westwood, "Speaker Adaptation Using Eigenvoices", *MS thesis*, Cambridge University, 1999.

[7] Gene H. Golub and Charles F. Van Loan, *Matrix Computations.* (3rd ed.), Johns Hopkins, 1996.

[8] P. Hall, D. Marshall, and R. Martin, "Merging and Splitting Eigenspace Models", *IEEE Tran. Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 1042-1049, 2000.

[9] P. Hall, D. Marshall, and R. Martin, "Adding and Subtracting Eigenspaces with Eigenvalue Decomposition and Singular Value Decomposition", *Image and Vision Computing*, vol. 20, pp. 1009-1016, 2002.