Eigenvoice 병합을 이용한 연속 음성 인식 시스템의 고속 화자 적응*

최동진(KAIST), 오영환(KAIST)

<차 례>

<Abstract>

Rapid Speaker Adaptation for Continuous Speech Recognition Using Merging Eigenvoices

Dong-jin Choi, Yung-Hwan Oh

Speaker adaptation in eigenvoice space is a popular method for rapid speaker adaptation. To improve the performance of the method, the number of speaker dependent models should be increased and eigenvoices should be re-estimated. However, principal component analysis takes much time to find eigenvoices, especially in a continuous speech recognition system. This paper describes a method to reduce computation time to estimate eigenvoices only for supplementary speaker dependent models and to merge them with the used eigenvoices. Experiment results show that the computation time is reduced by 73.7% while the performance is almost the same in case that the number of speaker dependent models is the same as used ones.

* Keywords: Speaker adaptation, Eigenvoice, Principal component analysis, Model merging.

* 본 연구는 정보통신부 대학 IT연구센터 육성, 지원사업의 연구결과로 수행되었습니다.

1. 서 론

화자 독립(speaker independent, SI) 음성 인식 시스템은 여러 화자로부터 수집된 음성 자료를 이용하여 훈련된 시스템으로 훈련 자료에 포함되지 않은 어떠한 사용자가 사용하더라도 고른 성능을 나타낸다. 화자 종속(speaker dependent, SD) 음성 인식 시스템은 한 사람의 음성 자료만 이용하여 훈련된 시스템으로 훈련 자료를 제공한 사람에 대해서는 화자 독립 시스템보다 높은 성능을 나타내지만, 다른 사용자에 대해서는 성능이 현저히 떨어지는 단점을 가지고 있다.

일반적으로 음성 인식 시스템은 불특정 다수의 화자를 대상으로 만들어지고, 화자 종속 시스템을 구성하기 위해서는 한 사람으로부터 매우 많은 음성 자료를 녹음해야 하는 어려움이 있어 화자 독립 음성 인식 시스템을 만들어 사용하는 것이 보통이다. 화자 적응(speaker adaptation)은 사용자가 제공한 소량의 음성을 이용하여 화자 독립 음성 인식 시스템을 그 화자에 특화되도록 보정하여 화자 종속 음성 인식 시스템에 가까운 성능을 얻어 낼 수 있는 기술이다.

화자 적응의 대표적인 방법으로는 maximum a posteriori(MAP) 화자 적응[1]과 maximum likelihood linear regression(MLLR)[2]로 대표되는 변환 기반 화자 적응이 있다. 하지만 이 방법들은 적응 자료가 매우 적은 양으로 제한적인 경우에는 각각 성능 향상이 거의 없거나 오히려 성능 저하가 일어나는 문제점이 있다[3].

최근에는 사용자에게 많은 양의 음성을 요구할 수 없는 어플리케이션의 종류가 늘어남에 따라 10에서 30초사이의 매우 적은 양의 자료를 이용하여 화자 적응을 수행하는 고속 화자 적응에 대한 관심이 높아지고 있다. 고속 화자 적응은 사용자에게 많은 양의 적응 자료를 이용하지 않고 음성 인식 시스템의 성능을 높일 수 있다는 장점이 있다. Eigenvoice 화자 적응[3][4]은 여러 개의 화자 종속 모델로부터 얻은 소수의 기저벡터(basis vector)들의 가중합을 이용하여 화자 독립 모델을 적응시킨다. 이 방법은 추정해야 하는 파라미터의 수를 최소화함으로써 적은 양의 적응 자료에도 강인한 결과를 얻을 수 있다는 장점을 가진다.

그러나 eigenvoice 화자 적응은 적응 자료의 양이 늘어나도 성능 향상이 제한적이고, 적응 과정 이전에 여러 개의 화자 종속 모델을 미리 구성해야 하는 등 여러 가지 문제점을 가지고 있다. 특히 화자 적응의 성능을 높이기 위하여 eigenvoice의 수를 늘리는 경우 화자 종속 모델의 수를 늘리고 다시 eigenvoice들을 재추정하는 과정이 필요한데, 이를 위해서는 매우 많은 계산량을 필요로 한다. 또한 연속 음성 인식 시스템에 eigenvoice 화자 적응을 이용하는 경우에는 eigenvoice의 차원 수가 급격히 증가하게 되므로 계산량 문제가 단어 인식 시스템보다 훨씬 큰 문제가 될 수 있다.

본 논문에서는 추가되는 화자 종속 모델을 포함하는 전체 화자 종속 모델들에 대해 eigenvoice들을 재추정하는 대신에, 추가된 화자 종속 모델들에 대해서만

eigenvoice들를 추정하고 이를 기존의 eigenvoice들과 병합함으로써 계산량을 감소시키는 방법을 제안한다.

먼저 2장에서는 eigenvoice 화자 적응에 대해 기술하고, 3장에서 각각 따로 구성된 eigenvoice들을 병합하는 방법을 설명한다. 4장에서 제안한 방법을 사용한 경우의 인식 성능 및 계산 시간 변화에 대한 실험 및 결과를 보이고, 5장에서 결론을 맺겠다.

2. Eigenvoice 화자 적응

2.1. 개요

Eigenvoice 화자 적응 방법은 여러 개의 화자 종속 모델로부터 principal component analysis (PCA)를 이용하여 기저벡터를 추출한 후 화자 적응에 사용한다. 자세한 알고리즘은 다음과 같은 순서를 따른다.

첫째, 화자 독립 모델과 화자 종속 모델을 구성한다. 화자 종속 모델은 화자 독립 모델을 구성하기 위해 사용하였던 데이터베이스를 이용하여 각각의 화자에 대해 HMM을 훈련시킴으로써 구축할 수 있으며, 각 화자별 모델의 가우시안 성분(mixture)의 수, 상태의 수, 모델 수 등은 모두 동일하게 한다.

둘째, 각각의 화자 종속 모델로부터 supervector를 구성한다. Supervector란 HMM 출력 가우시안의 모든 평균값들을 일정한 순서로 나열한 벡터이다. μ_r^m를 화자 r의 화자 종속 모델에 있는 m번째 가우시안 성분의 평균 벡터, n을 화자 종속 모델 r의 모든 가우시안 성분 수라고 하면 화자 r에 대한 supervector는 다음과 같은 식으로 나타낼 수 있다.

$$X_r = [\,(\mu_r^1)^T, (\mu_r^2)^T, \cdots, (\mu_r^n)^T\,]^T \tag{1}$$

셋째, R개의 모든 화자 종속 모델에 대한 R개의 supervector들에서 PCA를 이용하여 고유벡터(eigenvector), $e(1), e(2), \cdots, e(R)$를 추출한다. 이 R개의 고유벡터 중에 고유값(eigenvalue)이 큰 p개와 R개의 supervector들의 평균 벡터 $e(0)$를 각각 eigenvoice라 한다.

마지막으로, 특정 화자에 적응된 모델의 supervector는 다음식과 같이 eigenvoice들 간의 가중합으로 나타낼 수 있다.

$$X_t = e(0) + w(1)e(1) + w(2)e(2) + \cdots + w(p)e(p) \tag{2}$$

이때, $w(i)$는 maximum likelihood eigen decomposition (MLED) 방법[4]을 이용하여 계산할 수 있다.

2.2. SVD를 이용한 eigenvoice 계산 방법

Eigenvoice를 계산하기 위해서는 eigenvalue decomposition(EVD) 또는 singular value decomposition (SVD)를 이용하여 supervector들의 기저벡터를 구해야 한다. 본 논문에서는 SVD를 사용하여 eigenvoice를 계산하는 방법을 설명한다. 표기법을 통일하기 위하여, A^{i}_{mn}는 $m \times n$ 행렬의 i번째 열벡터를 $[A_{mn}\, b]$는 $m \times n$ 행렬에 벡터 b를 붙여 만든 $m \times (n+1)$ 행렬을 나타내기로 하자.

N개의 n차원 supervector x^{i}_{n}가 구성하는 행렬 X_{nN}있다고 가정하면, 평균벡터는 다음과 같은 식으로 계산할 수 있다.

$$\mu(X_{nN}) = \frac{1}{N}\sum_{i=1}^{N} x^{i}_{n} \tag{3}$$

이때 평균으로 이동된 행렬 Y_{nN}을 다음과 같이 정의할 수 있으며,

$$Y^{i} = x^{i} - \mu(X_{nN}) \tag{4}$$

SVD는 다음 식으로 정리할 수 있다.

$$Y_{nN} = U_{nn}\Sigma_{nN}V^{T}_{NN} \tag{5}$$

식 (5)에서 U_{nn}은 고유벡터들이 되고, Σ_{NN}는 특이값 (singular value)을 가지는 대각선행렬이 된다. 고유값이 매우 작은 고유벡터를 무시한다면 식 (5)는 다음 과 같이 쓸 수 있다.

$$Y_{nN} \approx U_{np}\Sigma_{pp}V^{T}_{Np} \tag{6}$$

여기서 행렬 U_{np}의 p개의 열벡터들과 평균벡터 $\mu(X_{nN})$를 합쳐서 p+1개의 eigenvoice로 정의한다.

Eigenvoice 화자 적응 방법의 성능을 높이기 위한 방법 중에 하나는 화자 적응에 사용하는 eigenvoice의 수를 증가시키는 것이다. 그러나 화자 종속 모델의 수가

충분하지 않은 경우 eigenvoice의 수가 일정 개수이상 증가하게 되면 적응 성능이 급격히 저하되는 결과가 발생하게 된다[6]. 따라서 화자 적응 성능을 높이기 위해서는 화자 종속 모델의 수를 늘리고, 여기에 따른 적절한 eigenvoice의 수를 선택하는 것이 중요하다.

식 (5)를 일반적인 Golub-Reinsch SVD 방법으로 계산하는 것은 $O(4n^2N+8nN^2+9N^3)$의 시간복잡도(time complexity)를 가진다[5]. Kuhn의 논문[3]에서처럼 단어 인식을 하는 경우에는 2808개의 비교적 작은 차원수 n을 가지는 supervector가 구성되었지만, 48개의 monophone, 모델당 3개의 상태를 가지는 HMM, 상태당 1개의 가우시안 성분을 사용하는 일반적인 연속 음성 인식 시스템에서 supervector의 차원수 n은 5655가 되고, eigenvoice 계산 시간이 n의 제곱에 비례하므로 4배에 가까운 계산량을 필요로 하게 된다. 더구나 인식 시스템의 성능을 높이기 위하여 triphone을 사용하고 mixture의 수를 6으로 늘려서 supervector의 차원수 n이 320112로 늘어나게 되면 eigenvoice를 구하는 데 걸리는 시간은 monophone, 한 개의 가우시안을 사용한 경우보다 약 3600배 증가하게 되어 화자 적응 시스템을 구성함에 매우 큰 문제가 된다.

Eigenvoice를 계산하는 과정이 화자가 적응 자료를 제공하기 이전인 off-line과정에서 이루어진다 하더라도 일반 PC에서 triphone, 6개의 mixture를 사용하는 연속 음성 시스템을 화자 적응에 사용하기 위해서 약 20시간이 소요되는 것은 부담이 아닐 수 없다. 또한, 핸드폰이나, PDA와 같이 처리속도가 떨어지는 시스템에서는 간단한 모델을 가지는 음성 인식 시스템이라 하더라도 많은 시간을 요구하게 될 것이다.

Eigenvoice 병합을 사용하지 않는 경우에는 화자 종속 모델이 추가되면 기존의 eigenvoice를 이용할 수 없고, 기존에 사용하였던 화자 종속 모델과 추가된 화자 종속 모델을 합하여 전체 화자 종속 모델에 대해 eigenvoice를 다시 계산해야 했다. 연속 음성 인식 시스템에서 eigenvoice를 계산하는 과정은 매우 많은 계산량을 필요로 하며, 화자 종속 모델의 수의 제곱에 비례하여 계산량이 증가하게 되므로, 이 과정의 계산량을 줄일 수 있다면 화자 적응 성능을 높이는 데에 따른 시스템의 재구성에 필요한 시간을 감소시킬 수 있을 것이다.

3. Eigenvoice들의 병합

이번 장에서는 추가된 화자 종속 모델만으로부터 eigenvoice를 계산하고 기존에 사용하던 eigenvoice와 병합하여 갱신함으로써 2.2절에서 언급한 eigenvoice 병합을 사용하지 않는 경우의 문제점을 해결하는 방법을 제안한다.

Eigenspace 모델을 병합하거나 나누는 많은 방법[7][8][9][10][11]이 있었지만, 이

방법들은 평균 벡터를 갱신할 수는 없었다. 음성 인식의 경우 가우시안 모델의 평균이 매우 중요하므로 평균 벡터가 갱신되지 않는다면 좋은 성능 향상을 기대할 수 없다. 하지만 최근 들어 평균 벡터를 갱신하면서 eigenspace 모델을 병합하는 방법이 제안되었다[12][13]. 이 방법은 기존의 eigenspace 구성시 사용하였던 자료를 참조하지 않고 병합된 eigenspace 모델을 구성할 수 있다.

3.1. 정규직교기저집합(orthonormal basis set) 구성

기존의 eigenvoice를 구성하기 위해 사용되었던 supervector들을 X_{nN}, 추가된 화자 종속 모델들의 supervector들을 Y_{nM}이라고 하고, 이를 이용해 구성된 eigenvoice 모델을 각각

$$\Omega(X) = (\mu(X), U(X)_{np}, \Sigma(X)_{p}, V(X)_{Np}, N) \tag{7}$$

$$\Omega(Y) = (\mu(Y), U(Y)_{nq}, \Sigma(Y)_{q}, V(Y)_{Mq}, M) \tag{8}$$

라고 하면, 병합된 모델을 다음과 같은 식으로 나타낼 수 있다.

$$\Omega(Z) = (\mu(Z), U(Z)_{ns}, \Sigma(Z)_{s}, V(Z)_{(M+N)s}, N+M) \tag{9}$$

이때, p, q, s는 각각 X, Y, Z에 대해 SVD를 적용하여 나온 고유벡터 중 eigenvoice로 사용할, 고유값이 큰 벡터의 수를 나타낸다.

또한, SVD의 정의에 따라 다음과 같은 식을 쓸 수 있다.

$$X_{nN} - \mu(X)1_{N} \approx U(X)_{np}\Sigma(X)_{p}V(X)^{T}_{Np} \tag{10}$$

$$Y_{nM} - \mu(Y)1_{M} \approx U(Y)_{nq}\Sigma(Y)_{q}V(Y)^{T}_{Mq} \tag{11}$$

$$Z_{n(N+M)} - \mu(Z)1_{(N+M)} \approx U(Z)_{ns}\Sigma(Z)_{s}V(Z)^{T}_{(N+M)s} \tag{12}$$

식 (12)에서 구하고자 하는 eigenvoice $U(Z)_{ns}$는 다음 식을 만족하는 정규직교기저집합 T_{ns}와 회전 행렬 R_{ss}로 나뉠 수 있다.

$$U(Z)_{ns} = T_{ns}R_{ss} \tag{13}$$

이때 T_{ns}는 병합되는 두 eigenvoice 모델 $U(X)_{np}$, $U(Y)_{nq}$와 $\mu(X)-\mu(Y)$를 모두 포함한다. 여기에 $\mu(X)-\mu(Y)$가 포함되는 이유는 이 벡터가 두 eigenvoice 모델이

이루는 공간의 영공간(null space)이 될 수 있기 때문이다.

T_{ns}를 구하기 위하여, $\Omega(X)$에 대한 $U(Y)_{nq}$의 잉여(residue) H_{nq}를 계산하면

$$G_{pq} = U(X)^{T}_{np} U(Y)_{nq} \tag{14}$$

$$H_{nq} = U(Y)_{nq} - U(X)_{np} G_{pq} \tag{15}$$

이 된다.

그리고 $U(X)_{np}$에 대한 $\mu(X) - \mu(Y)$의 잉여 h_n를 계산하면 다음과 같은 식을 얻을 수 있다.

$$g_p = U(X)^{T}_{np} (\mu(X) - \mu(Y)) \tag{16}$$

$$h_n = (\mu(X) - \mu(Y)) - U(X)_{np} g_p \tag{17}$$

H_{nq}는 $U(X)_{np}$에 대해 직교(orthogonal)이지만, $\Omega(X)$와 $\Omega(Y)$에는 공통인 eigenvoice가 존재할 수 있으므로 영벡터(zero vector)가 있을 수 있다. 따라서 행렬 $[H_{nq}, h_n]$의 열벡터 중 크기가 매우 작은 벡터를 제거하는 작업이 필요하다.

$$\nu_{nt} = Orthobasis(\zeta[H_{nq}, h_n]) \tag{18}$$

이때 ζ는 매우 공통인 eigenvoice를 제거하기 위해 매우 작은 열벡터(column vector)를 제거하는 함수이고, *Orthobasis*는 서로 직교인 단위 벡터를 구하는 함수이다. 본 논문에서는 Gram-Schmidt orthogonalization 방법을 사용하였다. t는 마지막으로 얻어진 벡터의 개수를 나타내며, 다음과 같은 관계식을 쓸 수 있다.

$$s = p + t \le p + q + 1 \le \min(n, M+N) \tag{19}$$

식 (18)을 이용하면 다음과 같은 식으로 정규직교기저집합을 나타낼 수 있다.

$$T_{ns} = [\, U(X)_{np}, \nu_{nt}] \tag{20}$$

3.2. 작은 크기의 행렬에 대한 SVD 계산

식 (10),(11),(12),(13),(20)을 이용하면 다음과 같은 식으로 정리할 수 있다.

$$
\begin{aligned}
Z_{n(N+M)} - \mu(Z)1_{(N+M)} &\approx [\,U(X)_{np}\Sigma(X)_{p}V(X)^{T}_{Np} + \mu(X)1_{N},\ U(Y)_{nq}\Sigma(Y)_{q}V(Y)^{T}_{Mq} + \mu(Y)1_{M}\,] \\
&= U(Z)_{ns}\Sigma(Z)_{s}V(Z)^{T}_{(N+M)s} \\
&= [\,U(X)_{np},\ \nu_{nt}]R_{ss}\Sigma(Z)_{s}V(Z)^{T}_{(N+M)s}
\end{aligned} \tag{21}
$$

여기서 양변에 $[\,U(X)_{np},\ \nu_{nt}]^{T}$를 곱하면

$$
\begin{aligned}
&[\,U(X)_{np},\ \nu_{nt}]^{T}[\,U(X)_{np}\Sigma(X)_{p}V(X)^{T}_{Np} + \mu(X)1_{N},\ U(Y)_{nq}\Sigma(Y)_{q}V(Y)^{T}_{Mq} + \mu(Y)1_{M}] \\
&= R_{ss}\Sigma(Z)_{s}V(Z)^{T}_{(N+M)s}
\end{aligned} \tag{22}
$$

와 같은 식을 얻을 수 있는데, 이 식에서 좌변에 대해 SVD를 취하면 R_{ss}를 계산할 수 있게 된다.

마지막으로 병합된 eigenvoice $U(Z)_{ns}$는

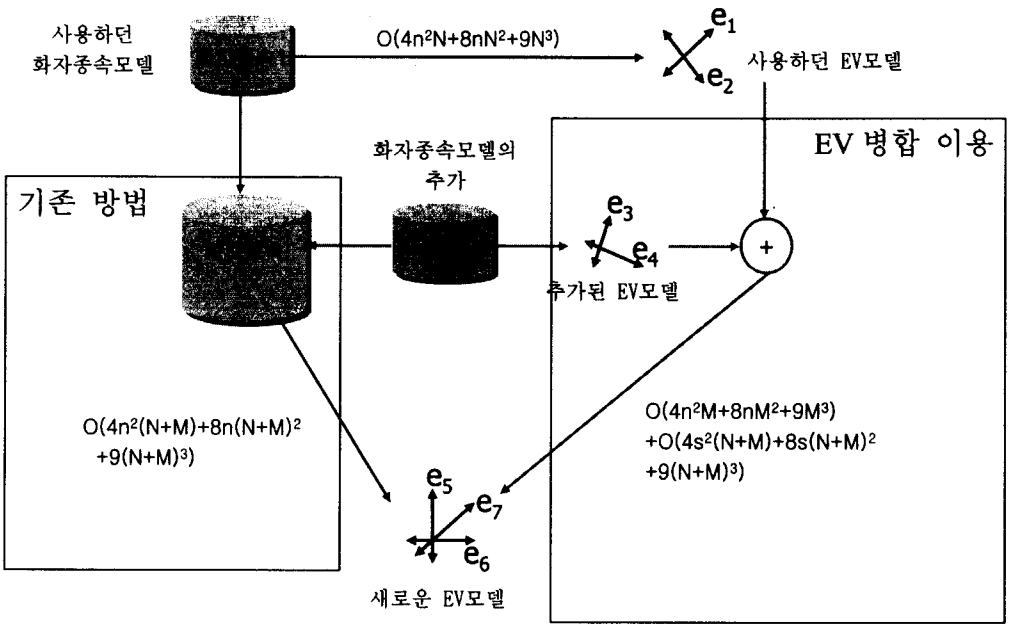
$$
U(Z)_{ns} = [\,U(X)_{np},\ \nu_{nt}]R_{ss} \tag{23}
$$

를 이용하여 계산할 수 있다.

3.3. Eigenvoice 병합의 시간복잡도

<그림 1>은 eigenvoice 병합을 사용한 경우와 제안한 방법을 이용하여 eigenvoice 병합을 사용한 방법의 차이를 나타내는 시스템 구성과 각각의 시간복잡도를 나타내는 그림이다. 기존에 화자 종속 모델인 "화자종속1"로부터 eigenvoice를 추출하여 화자 적응에 사용하고 있었다고 가정하자. 화자 적응 성능을 높이기 새로운 화자 종속 모델인 "화자종속2"가 추가되었다고 할 때, 병합을 이용하지 않는 경우에서는 "화자종속1"과 "화자종속2"를 합쳐 "화자종속1+2"를 만들고 이것으로부터 eigenvoice를 추출하여 사용해야 했다. 이 과정에서 대부분의 계산시간을 차지하는 SVD과정의 시간복잡도는 $O(4n^{2}(N+M)+8n(N+M)^{2}+9(N+M)^{3})$가 된다. 반면에, eigenvoice 병합을 이용한 방법에서는 추가된 "화자종속2"만 이용하여 eigenvoice를 추출한 다음, 기존에 사용하던 eigenvoice와 병합하여 새로운 eigenvoice를 구성하게 된다. 이때 "화자종속2"로부터 eigenvoice를 추출하는데 걸리는 시간복잡도가 $O(4n^{2}M+8nM^{2}+9M^{3})$이고, 두 eigenvoice들을 병합하는데 걸리는 시간복잡도는, 가장 계산량을 필요로 하는 부분이 식 (22)의 좌변에 SVD를 적용하는데 걸리는 시간이므로 $O(4s^{2}(N+M)+8s(N+M)^{2}+9(N+M)^{3})$이 된다. 일반적인 eigenvoice 화자 적응에서 $s<p+q+1\leq N+M+1\ll n$이므로

eigenvoice 병합에 걸리는 시간은 무시할 수 있다. 따라서 제안한 eigenvoice 병합을 이용하면 기존에 사용하던 eigenvoice를 구하는 데 걸린 시간만큼이 절약됨을 알 수 있다.



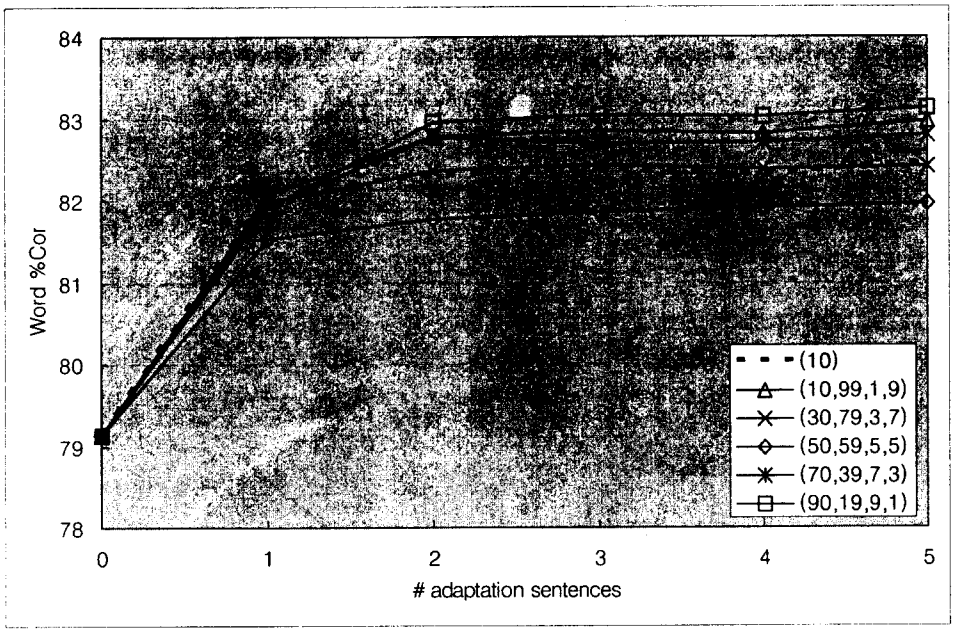
<그림 1> Eigenvoice병합을 사용하지 않은 경우와 사용한 방법의 시스템 구성 및 시간복잡도 비교

4. 실험 및 결과

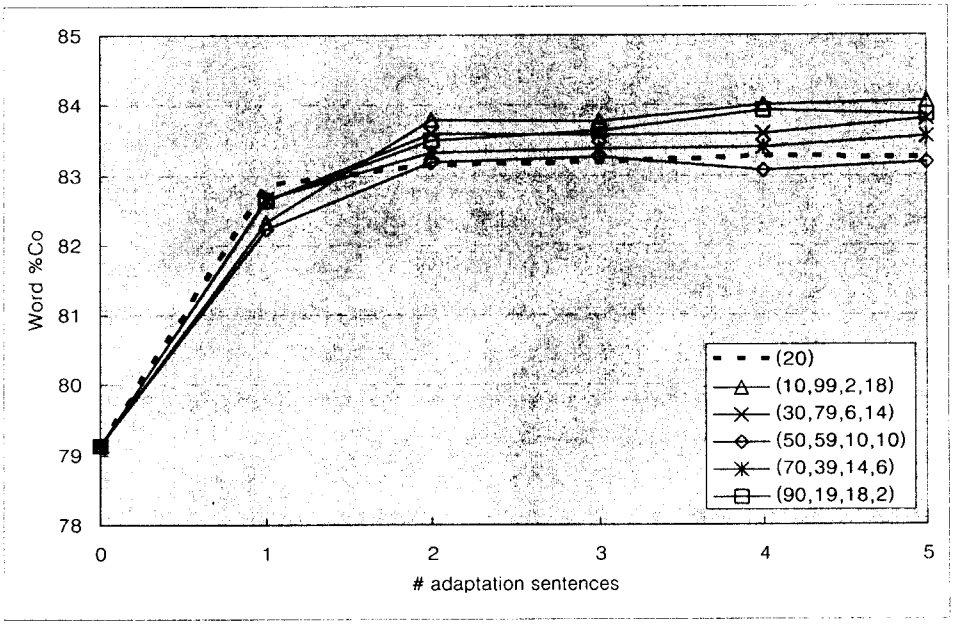
실험에 사용된 데이터베이스는 ARPA Resource Management task (RM)이다. 이 자료는 약 1000단어로 이루어진 미국식 영어 문장으로 구성되어 있고, 미국의 여러 가지 사투리가 포함되어 있다. 화자 독립 자료는 109명의 화자로부터 녹음되었고, 화자 종속 자료는 12명의 화자로 이루어져 있다.

모든 음성 자료로부터 12차 MFCC와 로그 에너지, 그리고 이 값들의 1차, 2차 차분을 이용해 총 39차 벡터를 추출하여 사용하였다. 음향 모델은 1개의 가우시안을 가지는 monophone으로 훈련하였으며, 각 HMM 모델은 3개의 상태를 가지도록 구성하였다.

109명분의 훈련 자료를 이용하여 화자 독립 모델을 훈련시켰고, 구성된 화자 독립 모델에 훈련 자료로 사용했던 각각의 화자별 훈련 자료로 MLLR과 MAP 화자 적응을 차례로 적용하여 109개의 화자 종속 모델을 구성하였다.



<그림 2> s=10일 때, eigenvoice 병합을 사용하지 않은 경우와 사용한 경우의 적응 자료 양(문장 수)의 변화에 따른 단어 인식률(%)



<그림 3> s=20일 때, eigenvoice 병합을 사용하지 않은 경우와 사용한 경우의 적응 자료 양(문장 수)의 변화에 따른 단어 인식률(%)

제안한 방법의 유효성을 검증하기 위하여 RM DB의 12명분의 화자 종속 모델용 자료를 사용하였다. 이 중에서 훈련용 자료중 일부를 적응 자료로 사용하였고, 평가용 자료인 화자당 100문장, 총 1200문장을 평가 자료로 이용하였다.

<그림 2>, <그림 3>은 eigenvoice 병합을 사용하지 않은 경우와 eigenvoice들을 병합한 후 MLED를 이용해 적응된 모델을 구성한 경우 각각에 대한 적응 자료 양에 따른 단어 인식률을 나타내는 그래프이다. Eigenvoice는 각각 10개와 20개를 사용하였고, 점선은 eigenvoice 병합을 사용하지 않은 겨우를, 실선은 eigenvoice 병합을 사용한 경우이다. 이때, 실선에 대한 설명중 괄호 안의 4개의 수치는 앞에서부터 순서대로 기존 eigenvoice를 구성할 때 사용한 화자 종속 모델의 수, 추가된 화자 종속 모델의 수, 기존 eigenvoice의 수, 추가된 eigenvoice의 수를 나타낸다. 실험 결과에서 알 수 있듯이 기존의 eigenvoice수나 추가된 eigenvoice의 수와 관계없이 비슷한 단어 인식률을 나타냄을 알 수 있다.

<표 1>, <표 2>는 <그림 1>, <그림 2>에서 보인 실험 결과에서 추가되는 화자 종속 모델의 수와 eigenvoice 수에 따른 여러 가지 경우에 대한 평균 인식률을 정리한 것이다. 적응 자료의 양과 관계없이 제안한 eigenvoice 병합 방법을 사용하지 않은 경우와 인식률 차이가 거의 없음을 알 수 있다.

<표 1> s=10일 때, 적응 자료의 양에 따른 기존 방법과 제안한 eigenvoice 병합을 이용한 경우의 평균 인식률 비교 (적응 전 인식률 : 79.12%)

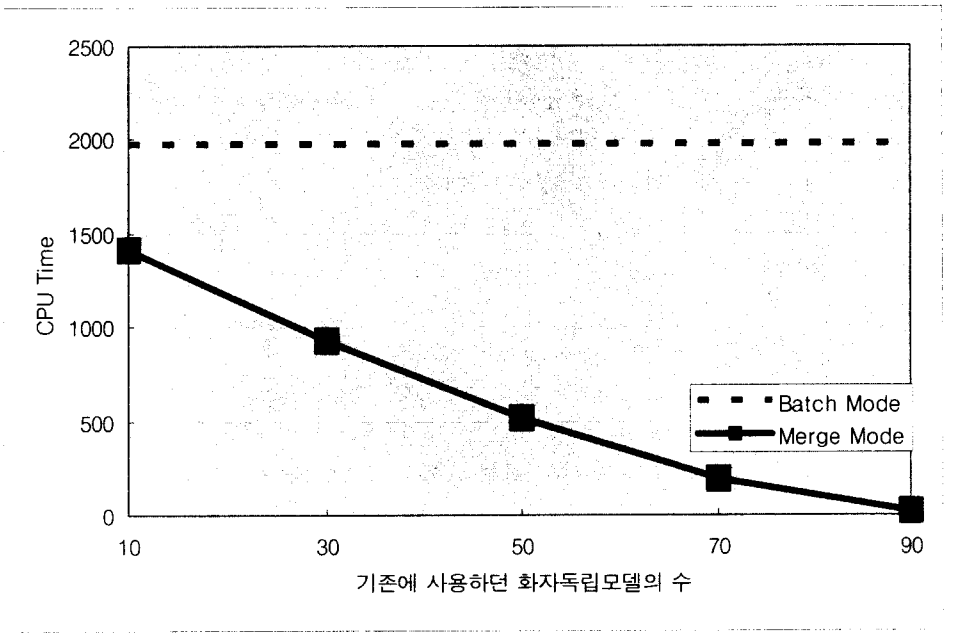
적응자료의 양 (문장)	1	2	3	4	5
eigenvoice 병합 미사용 (%)	82.11	82.31	82.38	82.32	82.47
eigenvoice 병합 사용 (%)	81.89	82.53	82.57	82.57	82.65

<표 2> s=20일 때, 적응 자료의 양에 따른 기존 방법과 제안한 eigenvoice 병합을 이용한 경우의 평균 인식률 비교 (적응 전 인식률 : 79.12%)

적응자료의 양 (문장)	1	2	3	4	5
eigenvoice 병합 미사용 (%)	82.86	83.14	83.20	83.27	83.25
eigenvoice 병합 사용 (%)	82.50	83.48	83.52	83.60	83.70

<그림 4>는 기존에 사용하던 화자 종속 모델과 추가된 화자 종속 모델을 합하여 총 109개의 모델을 이용하여 eigenvoice 화자 적응을 하려고 할 때, eigenvoice 병합을 사용하지 않은 경우와 사용한 경우에 대한 계산량의 차이를 보이고 있다. 가로축은 기존에 사용하던 화자 종속 모델의 수를 나타내며, 109에서 이 수를 뺀 수가 추가되는 화자 종속 모델의 수가 된다. CPU time을 측정하기 위해 같은 프로그램을 여러 번 반복하여 측정하였으며 다른 프로세스의 실행에 따른 영향을 최

소화하기 위하여 가장 작은 값을 실험값으로 취하였다. Eigenvoice 병합을 사용하지 않은 경우(점선) 추가된 음성 자료의 양과 관계없이 기존의 음성자료와 합하여 다시 eigenvoice를 계산해야 하므로 일정한 계산량을 나타낸다. 반면에 eigenvoice 병합을 이용한 경우는 추가된 화자 종속 모델에 대한 eigenvoice만 계산하면 되고, 추가된 모델의 양이 적으면 적을수록 eigenvoice 계산에 소요되는 시간이 적어지므로 최종 eigenvoice를 얻는데 걸리는 시간이 점차 줄어듦을 알 수 있다.



<그림 4> Eigenvoice 병합을 사용하지 않은 경우와 사용한 경우의 계산량

5. 결 론

본 논문에서는 eigenvoice를 이용한 화자 적응에서 문제되었던 계산량을 줄이기 위해 화자 종속 모델의 추가시 기존에 사용하던 것과 추가된 것을 합한 전체 화자 종속 모델로부터 새로 eigenvoice를 추정하지 않고, 추가된 자료에 대한 eigenvoice만 계산한 후 기존의 것과 병합하는 방법을 제안하였다.

이를 이용하여 eigenvoice를 계산하기 위한 시간을 추가된 자료의 양에 따라 감소시켰으며, 성능 저하는 거의 일어나지 않았다. 실험 결과 기존에 사용하던 화자 종속 모델의 수와 같은 수의 화자 종속 모델이 추가되었을 때, 인식률의 변화는 거의 없는 정도였지만, 계산하는데 걸린 시간은 약 1/4로 감소시킬 수 있었다.

참고문헌

[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, Apr. 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, Apr. 1995.

[3] R. Kuhn, J. Junqua, et al., "Rapid speaker adaptation in eigenvoice space", *IEEE Tran. Speech and Audio Proc.*, vol. 8, no. 6, pp. 695-707, Nov. 2000.

[4] R. Kuhn, P. Nguyen, et al., "Eigenvoices for speaker adaptation", *Int. Conf. Speech Language Processing*, vol. 5, pp. 1771-1774, Dec. 1998.

[5] G. H. Golub and C. F. V. Loan, *Matrix Computations.* (3rd ed.), Johns Hopkins, 1996.

[6] R. Westwood, "Speaker adaptation using eigenvoices", MS thesis, Cambridge University, 1999.

[7] J. R. Bunch, C. P. Nielsen, and D. C. Sorenson, "Rank-one modification of the symmetric eigenproblem", *Numerische Mathematik*, vol. 31, pp. 31-48, 1978.

[8] J. R. Bunch and C. P. Nielsen, "Updating the singular value decomposition", *Numerische Mathematik*, vol. 31, pp. 111-129, 1978.

[9] S. Chandrasekaran, B. S. Manjunath, et al., "An eigenspace update algorithm for image analysis", *Graphical Models and Image Processing*, vol. 59, no. 5, pp. 321-332, Sep. 1997.

[10] R. D. DeGroat and R. Roberts, "Efficient, numerically stablized rank-one eigenstructure updating", *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 38, no. 2, pp. 301-316, Feb. 1990.

[11] H. Murakami and B. V. K. V. Kumar, "Efficient calculation of primary images from a set of images", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, no. 5, pp. 511-515, Sep. 1982.

[12] 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, Sep. 2000.

[13] 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, Dec. 2002.

접수일자 : 2005년 2월 10일
게재결정 : 2005년 3월 15일

▶ 최동진(Dong-jin Choi)
주소: 305-701 대전광역시 유성구 구성동 373-1번지 한국과학기술원
소속: 한국과학기술원(KAIST) 전자전산학과 전산학전공 음성언어연구실
전화: 042) 869-3556
E-mail: cdjin@speech.kaist.ac.kr

▶ 오영환(Yung-Hwan Oh)
주소: 305-701 대전광역시 유성구 구성동 373-1번지 한국과학기술원
소속: 한국과학기술원(KAIST) 전자전산학과 전산학전공 음성언어연구실
전화: 042) 869-3516
E-mail: yhoh@speech.kaist.ac.kr