한국음향학회지 제33권 제3호 pp. 216~224 (2014)
The Journal of the Acoustical Society of Korea Vol.33, No.3 (2014)
http://dx.doi.org/10.7776/ASK.2014.33.3.216

pISSN : 1225-4428
eISSN : 2287-3775

Universal Background Model 클러스터링 방법을 이용한 고속 화자식별

Fast Speaker Identification Using a Universal Background Model Clustering Method

박주민, 서영주, 김회린†

(Jumin Park, Youngjoo Suh, and Hoirin Kim†)

한국과학기술원 전기및전자공학과

(접수일자: 2014년 1월 21일; 수정일자: 2014년 4월 4일; 채택일자: 2014년 4월 16일)

초 록: 본 논문은 Gaussian Mixture Model (GMM) 기반의 화자식별에서 급격한 계산 복잡도 감소를 위한 새로운 방법을 제안한다. 일반적으로 GMM 기반의 화자식별 시스템은 테스트 발성의 길이, 등록 화자의 수, GMM의 크기 등 크게 세 가지 요인에 비례하는 많은 계산 복잡도를 가진다. 이러한 점은 화자식별 시스템이 다양한 응용분야에 적용되는 것을 막는 큰 요인이기에 계산 복잡도와 식별 성능 사이의 trade-off 관계는 실제 적용을 위해 가장 중요한 고려요소이다. 식별 성능을 거의 그대로 유지하면서 최대한 계산 복잡도를 감소시키기 위해 우리는 Universal Background Model (UBM) 클러스터링 접근 방법을 제시하고, 또한 이 방법은 실시간 구조의 화자식별에 적용할 수 있다는 것을 보여준다. 제안한 방법의 실험을 통해 미미한 정도의 식별 성능 저하에서 speed-up factor 6의 결과를 얻을 수 있었다.
핵심용어: 화자식별, GMM, UBM 클러스터링, 계산 복잡도

ABSTRACT: In this paper, we propose a new method to drastically reduce computational complexity in Gaussian Mixture Model (GMM)-based Speaker Identification (SI). Generally, GMM-based SI systems have very high computational complexity proportional to the length of the test utterance, the number of enrolled speakers, and the GMM size. These make the SI systems difficult to be used in various real applications in spite of their broad applicability. Thus, a trade-off between computational complexity and identification accuracy is considered as a primary issue for practical applications. In order to reduce computational complexity sharply with a little loss of accuracy, we introduce a method based on the Universal Background Model (UBM) clustering approach and then we show that it can be used successfully in real-time applications. In experiments with the proposed algorithm, we obtained a speed-up factor of 6 with a negligible loss of accuracy.
Keywords: Speaker identification, GMM, UBM clustering, Computational complexity
PACS numbers: 43.72. Fx

I. 서 론

화자 인식은 발성된 음성이 어떤 화자의 것인지를 맞추는 분야로서 화자식별과 화자 검증 두 가지 분야로 더 나누어 볼 수 있다. 화자식별은 발성된 음성이 미리 등록되어 있는 화자 중에 어떤 화자의 것인지를 맞추는 분야이고 화자 검증은 어떤 화자의 것이라고 주장되어진 음성이 실제 그 화자의 음성이 맞는지 아닌지를 판별하는 분야이다. 화자 인식은 각 화자가 가진 발성기관의 고유한 생체 정보를 확인한다는 면에서 전통적으로 보안분야에 적용되어 왔고 최근 스마트 폰 등의 급격한 모바일 환경의 발전과 간편한 사용자 인터페이스 제공을 위한 목적으로 많은 적용들이 시도되고 있다. 예를 들면, 핸드폰이나 내비게이션의 경우 화자가 발성한 간단한 명령

†**Corresponding author:** Hoirin Kim (hrkim@ee.kaist.ac.kr)
Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, 291 Daehak-Ro, Yuseong-Gu, Daejeon 305-701, Republic of Korea
(Tel: 82- 42-350-7417, Fax: 82-42-350-7619)

을 인식하여 수행하고자 할 때 사용자의 음성을 인식하고 다른 사람의 음성은 거절하는 상황, 폰 뱅킹 등과 같이 사용자의 민감한 정보들을 요구하는 금융거래 시에 해당 화자의 음성이 맞는지 아닌지를 판단하여 보안성을 증대하기 위한 상황들을 생각해 볼 수 있다. 이러한 상황들은 인식 과정의 실시간 처리를 그 기본으로 하여 실제 적용을 하는 것으로 볼 수 있다.

한편 최근 음성 인식, 영상 등 다양한 분야에서 신경망, Support Vector Machine(SVM) 등을 이용한 연구들이 수행되고 성공적인 결과들이 보고되고 있다.[1-3] 새로운 알고리즘의 계속되는 연구와 발전에도 불구하고 Gaussian Mixture Model(GMM)이 가진 개념적인 명료함과 유용한 통계적인 특성들로 인해 여러 분류, 인식 분야에서 아직도 일반적으로 폭넓게 사용되고 있다. 화자 인식에서도 역시 GMM은 가장 널리 사용되고 있는 방법이다.[4] GMM 기반의 화자 검증은 그 분야의 특성상 claimed speaker model과 UBM 모델에 대한 계산을 수행하고 비교, 판별하기 때문에 작은 계산 복잡도를 가지는데 비해 화자식별은 식별 결과를 얻기 위해 기본적으로 모든 등록된 화자 모델들에 대한 계산을 수행해야 하기에 화자 검증에 비해 상대적으로 많은 계산 복잡도를 필요로 하는 분야이다. 화자식별에서 이러한 많은 계산 복잡도로 인해 식별 결과가 지연되는 병목현상(bottleneck effect)이 발생한다. 화자식별에 있어 병목현상에 영향을 끼치는 요인에는 테스트 발성의 길이, 등록 화자의 수, 화자 모델의 크기(가우스 분포의 개수) 등 크게 이 세 가지를 들 수 있고, 계산 복잡도는 이들에 비례하는 관계를 가진다. 만약 좋은 식별 성능을 가지는 시스템을 구성했을지라도 계산 복잡도의 문제로 인식 결과가 크게 지연될 경우 이는 실제 적용분야에 적절한 활용을 기대할 수 없다. 따라서 성공적으로 실제 적용분야에 활용하기 위해서는 이 계산상에서의 병목현상을 억제하는 것이 중요하다. 정리하면, 좋은 식별 성능을 얻기 위한 시스템을 구성하기 위해서는 일반적으로 병목현상이 심화되는 방향으로 시스템을 설계해야 하는데 반해 계산 복잡도를 작게 가져가기 위해서는 시스템의 설계 조건에 제한을 가하게 되고 결과적으로 식별 성능이 저하되는 시스템을 구성 할 수 밖에 없다. 일반적으로 식별 성능과 계산 복잡도에는 trade-off 관계가 성립한다. 본 논문에서는 최적의 식별 성능과 비교해 동등하거나 미미한 정도의 성능 저하에서 최대한 빠른 식별 결과를 얻기 위해 급격한 계산 복잡도 감소를 달성하는 soft trade-off 관계를 목표로 연구를 수행하였다.

본 논문의 구성은 다음과 같다. II장에서 GMM 기반의 화자식별 시스템에 대한 간략한 설명, III장에서는 계산 복잡도 감소와 관련된 선행 연구들에 대한 소개와 각 연구들이 가지는 의미에 대해 논하고 제안된 여러 알고리즘들에 대하여 설명한다. IV장에서는 본 논문에서 제안하는 방법인 Universal Background Model(UBM) 클러스터링 알고리즘 소개, V장에서 관련된 실험과 VI장에서의 결론으로 구성하였다.

II. GMM 기반 화자식별 시스템

GMM은 아래의 식과 같이 정의 된다.

$$p(\boldsymbol{x}|\lambda)=\sum_{i=1}^{M} w_i b_i(\boldsymbol{x}). \quad (1)$$

$\boldsymbol{x}$는 특징 벡터, M은 가우스 분포의 개수, w_i는 i번째 혼합의 가중치(weight), $b_i(\boldsymbol{x})$는 D 차원 가우시안, λ는 GMM 확률밀도함수의 파라미터 집합 $\lambda=\{w_i,\boldsymbol{\mu_i},\Sigma_i\}, i=1,2,...,M$을 의미한다.

이때 혼합 성분은 D 차원 가우시안으로 다음과 같이 정의한다.

$$b_i(\boldsymbol{x})=\frac{1}{(2\pi)^{D/2}\left|\Sigma_i\right|^{1/2}} exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^t \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_i})\right\}, \quad (2)$$

여기서 $\boldsymbol{\mu_i},\Sigma_i$는 i번째 혼합의 평균 벡터, 공분산 행렬(covariance matrix)을 의미한다.

화자식별은 훈련 과정(training stage)과 인식 과정(test stage) 두 단계로 나눌 수 있다. GMM 기반의 화자식별의 훈련 단계는 UBM 구성과 훈련된 UBM을 기반으로 최대사후(MAP: Maximum *a Posteriori*) 적응 기법을 이용한 화자 모델을 훈련하는 과정으로 요약할 수 있고, 인식 과정은 최대우도(ML: Maximum Likelihood) 조건을 만족하는 화자를 식별 화자로 인식하는 과정이다.

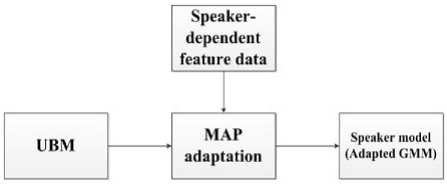

Fig. 1. Speaker model training.

2.1 훈련 과정

일반적으로 화자인식에서 화자 모델을 훈련하는 첫 단계는 화자 독립의 UBM을 훈련하는 것이다. UBM은 불특정 다수의 화자를 나타내는 큰 하나의 GMM으로 이후 화자 종속적 특징 데이터를 이용해 MAP 적응을 수행하여 해당하는 화자 모델을 훈련하는 단계를 거친다.[5,6] 이 과정을 Fig. 1에서 보여준다.

2.2 인식 과정

인식 단계에서는 테스트 발화 음성과 화자모델간의 우도(likelihood) 계산을 수행한다. 시간 T까지의 음성 특징 벡터열 $X=\{\boldsymbol{x_1},\boldsymbol{x_2},\cdots,\boldsymbol{x_T}\}$, λ_s는 화자 s의 모델 파라미터 집합을 의미 할 때, GMM 우도를 구하는 식은 다음과 같이 나타난다.

$$p(X|\lambda_s)=\prod_{t=1}^{T}p(\boldsymbol{x_t}|\lambda_s), \tag{3}$$

이때 인식화자를 판별하는 결정 조건은 최대우도 조건을 따르고 다음과 같이 나타낼 수 있다.

$$\hat{s}=\arg\max_{1\le s\le N} p(X|\lambda_s). \tag{4}$$

곱의 연산을 합의 연산으로 바꿔주기 위해 로그를 취하면 로그우도를 구하는 식은

$$\log p(X|\lambda_s)=\sum_{t=1}^{T}\log p(\boldsymbol{x_t}|\lambda_s), \tag{5}$$

이고, 인식 화자를 판별하는 결정 조건은 다음과 같이 나타난다.

$$\hat{s}=\arg\max_{1\le s\le N}\sum_{t=1}^{T}\log p(\boldsymbol{x}_t\,|\,\lambda_s). \tag{6}$$

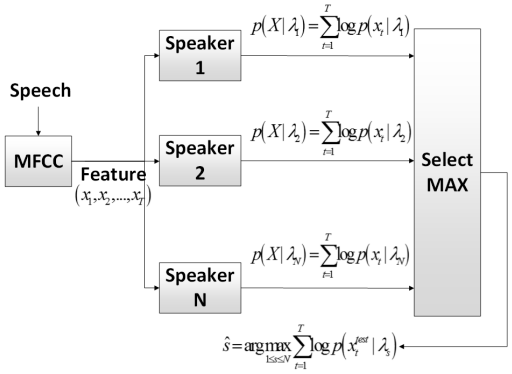

Fig. 2. Test stage of an SI system.

화자식별의 인식 과정은 Fig. 2와 같이 나타낼 수 있다.

III. 기존 연구

화자식별 분야에 계산 복잡도 감소를 위한 의미 있는 여러가지 연구들이 진행되어 왔다. 대표적으로 2006년도에 실시간 화자식별을 위한 논문이 발표 된 바 있다.[7] 참고문헌 7에서는 다양한 Pre-Quantization (PQ), speaker pruning 알고리즘을 소개하고 이 알고리즘들을 이용해 실시간 화자식별 구조를 제시한 뒤 관련된 실험과 결과를 제시하고 있다. 먼저 pre-quantization은 원래의 음성 벡터열 X를 원래보다 더 짧은 길이의 음성 벡터열 $\hat{X}$로 줄이는 알고리즘이다. 이는 물리적인 발성기관의 점진적인 dynamics의 특성상 인접한 음성 특징 벡터들은 비슷한 음향학적 특성을 가진다는 점을 고려해 특징 벡터열의 redundancy를 줄이는 것을 기본원리로 한다. 이와 관련하여 7에 앞서 McLaughlin 등[8]은 화자식별의 성능에 영향을 끼치는 요인 중에 주목해야 할 점으로 계산에 참여하는 절대적인 특징벡터의 수보다 특징 벡터들이 가진 음향학적인 다양성이 식별 시스템의 성능에 크게 영향을 끼칠 것이라는 점을 또한 언급하고 있다. 간략히 몇 가지 PQ 알고리즘에 대해 설명한다. Static PQ는 Fig. 3과 같이 표현할 수 있다. 여기서 PQ의 파라미터 R은 3일 경우이다.

Fig. 3과 같이 일정한 간격으로 특징 벡터를 선택하고 특징벡터열의 길이를 줄이는 것이 static PQ이다. Averaging PQ 알고리즘은 일정 간격의 벡터들을 평균하여 대표하는 벡터를 선택하는 방법으로 벡터

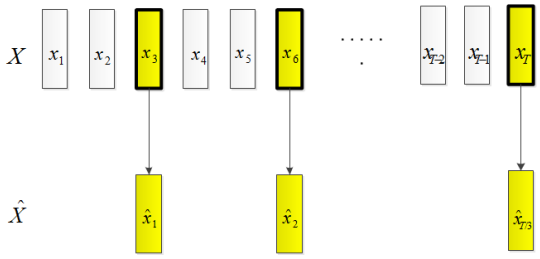

Fig. 3. Example of static PQ (R=3).

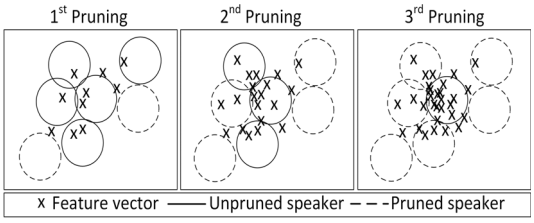

Fig. 4. Example of speaker pruning (pruning interval =10, pruning size=2).

열의 길이를 줄인다. Variable Frame Rate(VFR) PQ는 일반적으로 자음과 같이 음성의 스펙트럼 변화가 큰 구간에서는 촘촘하게 뽑아내고 모음이나 준정적인 (quasi-stationary) 특성의 구간에서는 듬성듬성 뽑아내어 길이를 줄이는 방법이다.[9,10]

Speaker Pruning(SP) 알고리즘은 인식 과정에서 현저히 점수가 낮은 화자모델을 계산 과정에서 과감히 제외하는 간단한 static pruning algorithm이다. 이 알고리즘의 작동 예는 Fig. 4에서 표현했다. 여기서 pruning stage를 거치는 테스트 벡터의 수를 나타내는 파라미터인 pruning interval은 10, 해당 pruning stage에서 pruned speaker의 수를 의미하는 pruning size는 2일 경우이다. 이에 따라 이 알고리즘에서는 Fig. 4와 같이 일정한 개수의 테스트 벡터가 확보됨으로써 정해지는 pruning level마다 pruning size 개수만큼의 화자모델을 단계적으로 감소시킨다.

참고문헌 7에서는 PQ, speaker pruning에 대한 소개와 더불어 두 가지 방법을 적용한 실시간 화자식별의 구조에 대해서도 설명하고 있다. 이는 Fig. 5와 같이 구조로 구성되어 있다. 실시간 구조의 순서는 먼저 실시간으로 입력되는 음성의 일정 구간을 정하고 특징 추출을 수행한다. 그 후 VAD를 통해 음성 구간을 검출하고 PQ를 통해 음성 특징 벡터열의 길이를 짧게 만든다. 이 짧아진 특징 벡터열을 대상으로 인

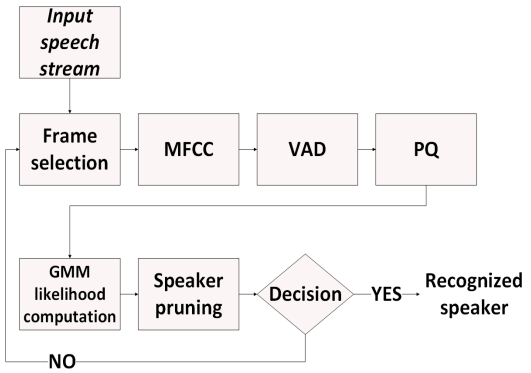

Fig. 5. Diagram of real-time speaker identification.

식 과정에서 모든 화자들에 대한 우도 계산을 거치고 점수가 현저히 낮은 화자들은 speaker pruning 과정을 통해 제외시킨 후, 살아남은 화자가 한명이거나 더 이상의 입력 음성이 없는지를 판정(decision)하여 반복과정을 끝내고 화자를 식별한다.

이 논문에서는 TIMIT DB를 이용하여 GMM 기반으로 0 %에 가까운 오류율에서 최대 speed-up factor 10의 결과를 제시하였고 NIST DB를 이용하여 GMM 기반으로 약 17~19 %의 오류율에서 최대 speed-up factor 34의 결과를 보고하고 있다.

최근에는 화자모델에 대한 클러스터링 알고리즘을 적용한 연구들이 보고되고 있다. 대표적으로 2009년도에 소개된 11에서는 훈련 과정에서 화자 모델을 클러스터링하는 방법을 소개하고 있다. 훈련된 화자 모델들에 대해 클러스터링 알고리즘을 적용하고 테스트 음성과 비교해 음향학적 특성이 비슷한 몇몇의 화자 모델들을 골라내는 방법으로 계산 복잡도를 효과적으로 줄일 수 있는 실험 결과를 보고하고 있다. 이 논문에서는 약 20 %의 탐색 클러스터만을 이용해서 기존 베이스라인과 비교해 같은 인식 성능에서 speed-up factor 4.4를 얻은 것으로 보고 하고 있다. 결과적으로 충분히 효과적인 결과를 제시하고 있긴 하지만 테스트 발화가 끝나야 우도 계산을 수행하는 이 방법은 실제 적용을 위한 실시간 구조에서 작동하기 힘든 치명적인 단점을 가지고 있다.

IV. 제안 알고리즘

본 논문에서 제안하는 알고리즘의 이름은 UBM

clustering 기반 Gaussian Pruning(GP) 기법이다. 가우스 분포의 수에 해당하는 계산 복잡도 문제를 완화시키기 위한 방법이다. UBM의 mixture들에 클러스터링을 적용하고 테스트 특징벡터를 훈련된 UBM의 코드북과 비교해 가까운 혼합 성분을 정한 뒤 해당 혼합 성분의 정보를 모든 화자 모델에 넘겨 계산시에는 이미 적응된 동일한 성분의 혼합들만 우도 계산에 참여 시키는 방법이다. GMM 기반의 화자식별 시스템에서 GMM은 해당화자에 대한 음향학적인 특성을 반영한다고 볼 수 있다. 이점은 테스트 발화와 GMM의 우도 계산값이 모든 혼합에 의해 동등하게 결정되는 것이 아니라 테스트 벡터와 가장 유사한 음향학적 특성을 가진 혼합들이 우도 점수에 가장 결정적인 기여를 할 것이라고 생각할 수 있다. 따라서 본 논문에서는 계산 복잡도를 감소시키기 위한 방법으로 테스트 벡터와 혼합들의 평균벡터를 이용해 가장 가까운 혼합 성분들 만을 우도 계산에 참여하는 방법을 고안하였다. 기존의 화자 모델 클러스터링 알고리즘[11]은 식별 성능을 충분히 유지하면서 계산 복잡도를 감소시키는데 의미있는 성과를 거두었음에도 불구하고 그 알고리즘의 특성상 화자의 발성이 끝나야 인식 과정을 거치기 때문에 실제 적용에 실시간으로 이용하기 위한 측면에서 한계점을 가지고 있다고 할 수 있다. 본 논문에서 제안하는 알고리즘의 중요한 점은 각 화자 모델의 혼합들을 따로 클러스터링 하는 것이 아니라 MAP 적응하기 전의 UBM을 기반으로 혼합 성분들을 클러스터링하고 계산시에는 UBM에서 선택된 혼합들을 화자 모델에 넘겨 화자 모델에서는 적응된 동일한 혼합 성분들만을 계산에 참여시킨다는 것이다. 이렇게 하는 데에는 몇 가지 이유가 있다. 첫 번째로 각 화자 모델의 코드북을 훈련할 때는 각 화자마다 다른 혼합 성분들을 가지고 계산을 하게 될 수 있다는 점이다. 이는 서로 다른 화자들이 각각 서로 다른 가중을 갖는 혼합 성분들의 우도 값을 계산한다는 의미인데 인식 과정에서 이는 식별 성능의 저하를 불러 일으킬 수 있다. 두 번째로 UBM을 기반으로 MAP 적응 기법을 이용해 화자 모델을 훈련할 때 해당 화자 모델이 UBM과 비교해 특징공간상에서 크게 이동하지 않기 때문에 충분히 적용 가능성이 존재한다는 점이다. 만약 화자 모델이 UBM에서 크게 이동한다면 식별의 우도 값이 크게 변할 것이고 이는 화자식별에서 우도 점수의 신뢰성 측면에서 문제점으로 지적할 수 있어 식별 성능을 저하 시키게 될 것 이라고 예상할 수 있다. 하지만 소량의 적응 데이터로 UBM을 적응시켜 훈련하는 해당 화자 모델들의 특성상 원래의 UBM에서 큰 움직임을 보이지 않는다고 볼 수 있고 이는 UBM의 혼합 성분들을 클러스터링 하는 것이 또한 타당하다는 것을 의미한다. 추가적으로 새로운 화자를 등록할 때 UBM을 클러스터링 함으로서 같은 UBM에서 해당 화자 모델이 훈련되었다면 새로운 코드북을 훈련하지 않아도 되는 장점이 있다. 제안하는 알고리즘의 작동 순서는 Table 1과 같다.

Table 1. UBM clustering-based Gaussian pruning algorithm.

Step 1: Evaluate distances (Euclidean) between a test feature vector and pre-trained centroids of UBM.
Step 2: Choose the N nearest centroids.
Step 3: Choose the corresponding UBM mixture components of the N centroids and then calculate the likelihoods between the test vector and the enrolled speaker models' corresponding mixture (adapted mixture) components of N centroids.
Step 4: Repeat Steps 1 through 3 until the last test vector.

본 논문에서 제안하는 UBM clustering 기반 Gaussian pruning 방법을 이용해 [7]에서 제안한 계산 복잡도 감소 알고리즘을 적용한 실시간 구조에 적용할 수 있다. Fig. 6은 기존에 제안된 실시간 구조에서 Gaussian pruning 방법이 적용된 화자식별의 실시간 구조이다. Gaussian pruning을 적용해서 혼합의 수로 인한 계산 복잡도를 감소시킬 수 있다.

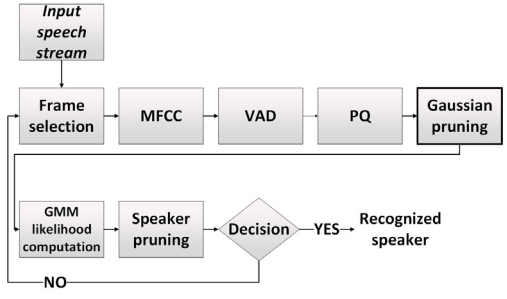

Fig. 6. Diagram of real-time speaker identification with Gaussian pruning.

V. 실험 및 실험 결과

5.1 데이터 베이스

실험에 이용한 데이터 베이스는 연구 목적으로 수집한 clean DB이다. 본 DB는 임베디드 보드와 컴퓨터를 이용해 수집한 DB로 이루어져 있고 임베디드 보드를 통해 수집한 DB에는 전체적으로 음질에 약간의 열화가 존재한다. 본 논문에서는 임베디드 보드를 통해 수집한 DB의 문장 독립 데이터를 이용해 실험을 수행했다. 문장 독립 데이터는 글을 읽는 낭독체 타입으로 약 700명의 화자로 이루어져 있고 이 중 화자당 30 발화씩으로 이루어져 있으며 각 발화당 최대 약 10초 정도의 길이를 가지고 있다. 음성 데이터의 표본화 주파수는 16 kHz, bit 수는 16이다.

5.2 실험 환경

189명의 화자, 화자 당 30개 발화를 이용하여 UBM을 훈련했고, 화자모델을 훈련할 때 500명에 대해 화자 당 20개 발화를 적응데이터로, 각 화자 당 나머지 10개의 발화를 테스트 발화로 사용했다. 실험을 위해 사용한 특징은 20 ms의 frame size와 10 ms의 shift rate, 19차의 static, 19차의 델타 정보를 이용한 38차 MFCC를 활용했다. 음성구간 판별을 위해 에너지 기반의 VAD를 적용했다. 실험에 사용된 워크스테이션은 DELL 컴퓨터의 T1600이다. 이 컴퓨터의 CPU는 quad-core Intel(R) Xeon(R) CPU E312345@3.30 GHz 이고 테스트 환경은 Windows7 OS에서 C/C++언어 기반의 프로그램을 이용했다.

5.3 실험 결과

화자식별의 베이스라인 실험 결과는 Table 2와 같다. 가우스 분포의 개수가 64, 128, 256, 512개의 성능 평가 결과 64개 일 때 0.36 %, 128개 이상일때는 모두 0.34 %의 오류율로 성능이 포화되는 것을 확인할 수 있었다. 한편 이 실험에서의 Real-Time Factor(RTF)는 0.46, 0.87, 1.64, 3.19로 나타났다.

Pre-quantization을 적용했을 때 static, averaging, VFR에 대한 각 혼합별 실험결과를 베이스라인과 비교했을 때의 결과가 Fig. 7에 주어졌다. 오류율만 고려한 결과 베이스라인과 비교해 가장 좋은 성능을 보인 것은 static 이었다. 그 다음이 VFR, averaging으로 나타났다. Averaging 방법에서는 일정한 길이의 음성특징벡터열에 포함된 특징벡터들을 계수별로 시간에 따라 이동평균한 값을 사용한다. 이러한 평균화 과정을 통해서 음성의 단구간(short time) 스펙트럼 특성은 사라지게 된다. 따라서 이 averaging 방법에서의 두드러지게 저조한 성능은 음성신호에서 화자정보가 시불변적인(time invariant) 공통의 형태로 존재하는 것이 아니라 단구간 스펙트럼의 특성에

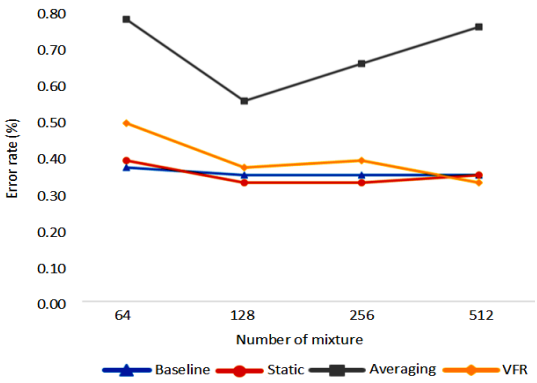

Fig. 7. PQ result.

Table 2. Baseline result.

Number of mixture	64	128	256	512
Error rate (%)	0.36	0.34	0.34	0.34
RTF	0.46	0.87	1.64	3.19

Table 3. SP result (pruning interval=5, pruning size=10).

Number of mixture	64	128	256	512
Error rate (%)	0.44	0.34	0.38	0.44
Speed-up factor	4.66x	4.71x	4.65x	4.73x

Table 4. SP result (pruning interval=10, pruning size=10).

Number of mixture	64	128	256	512
Error rate (%)	0.34	0.34	0.36	0.38
Speed-up factor	2.45x	2.52x	2.53x	2.53x

Table 5. SP result (pruning interval=15, pruning size=10).

Number of mixture	64	128	256	512
Error rate (%)	0.34	0.34	0.34	0.36
Speed-up factor	1.78x	1.81x	1.79x	1.79x

따라 각각 다르게 나타난다는 것을 의미한다.

다음은 speaker pruning 실험 결과로 pruning size를 10으로, pruning interval을 5, 10, 15로 달리 했을 때의 실험 결과가 Tables 3~5에 있다. Pruning size와 pruning interval 이 각각 10, 5 일 때, 그만큼 pruning stage의 수가 많아지기 때문에 베이스라인 시스템 대비 speed-up factor 가 높게 나왔지만 오류율은 가장 높게 나온 것으로 확인할 수 있다. 반면 pruning size와 pruning interval이 각각 10, 15일 때 speed-up factor가 가장 낮게, 오류율은 가장 낮게 나온 결과를 확인할 수 있었다.

본 논문에서 제안한 방법인 UBM clustering 기반 GP 알고리즘의 실험 결과가 Tables 6~8에 주어져 있다. 혼합이 각각 128, 256, 512개 일 때의 표에서 혼합 대비 총 클러스터 비율(총 클러스터 수)과 테스트에서 탐색 클러스터의 개수를 달리 하면서 실험결과를 얻었다. 탐색 클러스터 수를 작게 할수록 속도 향상이 크게 되는 것을 볼 수 있다. 가장 작은 5 %의 탐색 클러스터에서 약 5 ~ 8의 speed-up factor를 얻었고 오류율은 미미한 성능저하가 나타나는 것을 확인할 수 있었다.

128, 256, 512 혼합에 대한 알고리즘 조합별 실험 결과를 Table 9와 같이 얻었다. PQ의 경우 파라미터

Table 6. 128 mixture GP result.

Percentage of clusters to no. of clusters	5 %	10 %	20 %	30 %	50 %	Avg error rate (%)
10 % (13)	0.36 (4.58x)	0.34 (2.56x)	0.34 (1.89x)	0.34 (1.61x)	0.34 (1.23x)	0.344
20 % (26)	0.40 (8.09x)	0.36 (3.18x)	0.34 (2.07x)	0.34 (1.52x)	0.34 (1.12x)	0.356
30 % (39)	0.36 (6.35x)	0.38 (3.56x)	0.34 (2.06x)	0.34 (1.59x)	0.34 (1.14x)	0.352

Table 7. 256 mixture GP result.

Percentage of clusters to no. of clusters	5 %	10 %	20 %	30 %	50 %	Avg error rate (%)
10 % (26)	0.38 (7.17x)	0.32 (2.95x)	0.34 (2.05x)	0.32 (1.51x)	0.34 (1.15x)	0.340
20 % (52)	0.40 (6.09x)	0.36 (4.15x)	0.32 (2.46x)	0.34 (1.82x)	0.34 (1.25x)	0.352
30 % (78)	0.34 (5.89x)	0.32 (3.67x)	0.32 (2.37x)	0.34 (1.72x)	0.34 (1.18x)	0.332

Table 8. 512 mixture GP result.

Percentage of clusters to no. of clusters	5 %	10 %	20 %	30 %	50 %	Avg error rate (%)
10 % (51)	0.36 (5.54x)	0.32 (3.68x)	0.32 (2.16x)	0.32 (1.59x)	0.34 (1.11x)	0.332
20 % (102)	0.36 (6.44x)	0.32 (3.74x)	0.32 (2.17x)	0.34 (1.61x)	0.34 (1.11x)	0.336
30 % (156)	0.36 (6.33x)	0.32 (3.66x)	0.32 (2.11x)	0.32 (1.55x)	0.34 (1.10x)	0.332

Table 9. Summary of the results.

Algorithm \ Mixture	128 mixture	256 mixture	512 mixture
Baseline	1x (0.34)	1x (0.34)	1x (0.34)
PQ	9.00x (0.32)	8.67x (0.32)	8.57x (0.34)
SP	1.81x (0.34)	1.79x (0.34)	1.79x (0.36)
PQ+SP	9.13x (0.30)	9.42x (0.30)	9.67x (0.32)
GP	6.35x (0.36)	5.9x (0.34)	6.33x (0.36)
GP+PQ	53x (0.42)	58.42x (0.34)	55.2x (0.34)
GP+SP	24.07 (0.40)	30.22x (0.36)	23.5x (0.48)
GP+PQ+SP	59.6x (0.40)	72.15x (0.32)	60.14x (0.32)

R은 9로, speaker pruning의 경우 pruning interval은 15, pruning size 는 10 으로 정했다. 본 논문에서 제안하는 알고리즘인 UBM clustering 기반 Gaussian pruning 기법의 클러스터 개수는 혼합 개수의 30 %, 탐색 클러스터는 총 클러스터 대비 5 %로 정하고 실험을 수행했다. 실험 결과 UBM clustering 기반 Gaussian pruning 방법은 128, 256, 512 혼합별로 0.36 %, 0.34 %, 0.36 % 의 오류율에서 약 6 의 speed-up factor를 얻을 수 있었고 추가적으로 기존에 제시된 PQ와 speaker pruning 알고리즘을 동시에 적용할 경우 베이스라인 실험과 비교해 크게 나빠지지 않는 오류율에서 약 60~72의 speed-up factor를 얻을 수 있었다. 이 결과가 Table 9에 정리되어 있다. 표 안의 숫자는 speed-up factor를 나타내고 괄호 안의 숫자는 오류율(%)을 나타낸다.

VI. 결 론

본 논문에서는 UBM clustering 기반 Gaussian pruning 기법을 소개했다. 대규모 DB를 통해 실험한 결과, 100개 이상의 혼합에서 6의 speed-up factor 결과를 얻을 수 있었다. 그에 따른 성능의 저하가 미미했을 뿐 아니라 이 방법은 또한 실시간 구조에 적용할 수 있다는 장점을 가지고 있다. 또한 추가적으로 기존에 제안된 방법인 PQ, speaker pruning 방법을 모두 조합해 얻은 속도향상 효과는 60~72의 speed-up factor를 얻어 미미한 성능의 저하에서 급격한 계산 복잡도 감소를 이루어 낼 수 있었다.

추후 이 연구에 이어 추가적으로 확인해야 하는 사항들은 여러 잡음 부가 상황을 고려한 DB를 통한 추가적인 검증, 실제 적용 상황을 고려해 테스트 발성의 길이가 더 짧은 경우에 대한 실험 등을 들 수 있다.

감사의 글

본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음(10041807, 지식학습 기반의 다국어 확장이 용이한 관광/국제행사 통역률 90 % 급 자동 통번역 소프트웨어 원천 기술 개발).

References

1. W. M. Campbell, D. E. Sturim, and D. A. Reynolds, “Support vector machines using GMM supervectors for speaker verification,” IEEE Signal Process. Lett. **13**, 308-311 (2006).
2. B. Xiang and T. Berger, “Efficient text-independent speaker verification with structural Gaussian mixture models and neural network,” IEEE Trans. Speech Audio Process.**11**, 447-456 (2003).
3. K. R. Farrell, R. Mammone, and K. Assaleh, “Speaker recognition using neural networks and conventional classifiers,” IEEE Trans. Speech Audio Process. **2**, 194-205 (1994).
4. D. A. Reynolds and R. Rose, “Robust text-independent speaker identification using Gaussian mixture models,” IEEE Trans. Signal Process. **3**, 72-83 (1995).
5. D. A. Reynolds, T. F. Quatieri, and R. Dunn, “Speaker verification using adapted Gaussian mixture models,” Digit. Signal Process. **10**, 19-41 (2000).
6. J. L. Gauvain and C. H. Lee, “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains,” IEEE Trans. Speech Audio Process. **2**, 291-298 (1994).
7. T. Kinnunen, E. Karpov, and P. Franti, “Real-time speaker identification and verification,” IEEE Trans. Audio Speech Lang. Process. **14**, 277-288 (2006).
8. J. McLaughlin, D. A. Reynolds, and T. Gleeson, “A study of computation speed-ups of the GMM-UBM speaker recognition system,” in Proc. Eur. Conf. Speech Commun. Technol. (Eurospeech), 1215-1218 (1999).
9. Z.-H. Tan and B. Lindberg, “Low-complexity variable frame rate analysis for speech recognition and voice activity detection,” IEEE J. Sel. Top. Signal Process. **4**, 798-807 (2010).
10. Q. Zhu and A. Alwan, “On the use of variable frame rate analysis in speech recognition,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. 1783-1786 (2000).
11. V. R. Apsingekar and P. L. D. Leon, “Speaker model clustering for efficient speaker identification in large population application,” IEEE Trans. Audio Speech Lang. Process.**17**, 848-853 (2009).

저자 약력

▸ 박 주 민(Jumin Park)

2012년: 서울과학기술대학교 전기공학과 학사
2014년: KAIST 전기및전자공학과 석사

▸ 서 영 주(Youngjoo Suh)

1991년: 경북대학교 전자공학과 학사
1993년: 경북대학교 전자공학과 석사
2006년: KAIST 정보통신공학과 박사
2006년 ~ 현재: KAIST 전기및전자공학과 연구부교수

▸ 김 회 린(Hoirin Kim)

1984년: 한양대학교 전자공학과 학사
1987년: KAIST 전기및전자공학과 석사
1992년: KAIST 전기및전자공학과 박사
2000년 ~ 현재: KAIST 전기및전자공학과 교수