화자적응시스템을 위한 MLLR 알고리즘 연산량 감소

Reduction of Dimension of HMM parameters in MLLR Framework for Speaker Adaptation

김지운, 정재호
인하대학교 전자공학과 DSP Lab.
E-mail : G2001118@inhavision.inha.ac.kr, jhchung@inha.ac.kr

We discuss how to reduce the number of inverse matrix and its dimensions requested in MLLR framework for speaker adaptation. To find a smaller set of variables with less redundancy,we employ PCA(principal component analysis) and ICA(independent component analysis) that would give as good a representation as possible. The amount of additional computation when PCA or ICA is applied is as small as it can be disregarded. The dimension of HMM parameters is reduced to about 1/3 ~ 2/7 dimensions of SI(speaker independent) model parameter with which speech recognition system represents word recognition rate as much as ordinary MLLR framework. If dimension of SI model parameter is n, the amount of computation of inverse matrix in MLLR is proportioned to $O(n^4)$. So, compared with ordinary MLLR, the amount of total computation requested in speaker adaptation is reduced to about 1/80 ~1/150.

1. 서론

음성인식 시스템은 화자의 특성의 변화, 음향환경의 변화, 혹은 잘못된 모델링과 같은 훈련환경과 인식환경의 차이에 매우 민감하다. 따라서, 실제 환경에서 음성 인식 시스템은 이러한 환경의 불일치를 보상하는 것이 매우 중요하다. 이는 강인하고 환경에 따른 변화가 적은 음성 특징 추출, 모델링 방법의 향상, 적응이나 보상 방법을 이용한 인식 파라미터나 특징 벡터의 수정, 강인한 결정 방법 등 다양한 방법으로 행해진다.

HMM에 기반한 음성 인식 시스템에 적용되는 적응 방법은 크게 2가지로 분류된다. 그 중 하나는 변환에 기초한 적응방법으로서 HMM 파라미 터를 변환 함수에 의해 변환시키는 방법이다. MLLR, SM 혹은 constrained transform이 이 부류에 해당한다[1][6][7]. 또 다른 분류는 MAP적응 방법에 기인한 여러 가지 방법이다[2]. 일반적으로 적응 데이터가 제한된다면 변환에 기반한 적응이 cluster에 의존적인 변환 함수에 의해 모든 HMM 파라미터를 효과적으로 변환시킬 수 있다. 반면, 만일 충분한 데이터가 제공되면 MAP방법이 변환함수에 기반한 적응방법 보다 SI HMM 파라미터를 효과적으로 적응시킬 수 있다.[3][4]

본 논문은 적은 양의 적응 데이터가 제공되는 적응 환경을 목표로 MLLR방법을 이용한다. MLLR방법은 transform에 기반한 방법으로써 transform 행렬을 구할 때 많은 양의 역행열 연산을 수행해야 한다. 특히, 음성 특징 파라미터의 차수가 커질 수록, HMM 모델의 개수가 많아질 수록 기하 급수적으로 역행렬 연산횟수가 증가한다. 예를 들어, n차의 특징 파라미터를 사용하고 MLLR방법에서 regression tree의 base class가 m개라고 가정하면, $(n+1)*(n+1)$차원의 행열의 역행열 연산을 $m*n$회 행해야 한다. 따라서, MLLR 방법

에서 transform 행열을 구하는 데 요구되는 역행열의 연산 횟수를 줄이고 효율적으로 HMM 파라미터를 변환하는 방법이 요구된다. 잘 훈련된 SI모델의 각 mixture들을 몇 개의 class로 분류하고 이 class의 공간에 대한 사전정보를 이용하여 모델 및 음성 특징 파라미터의 차수를 줄이는 데에 사용한다. 즉, 각 class의 supervector에 주성분 분석이나 독립성분 분석을 이용하여 낮은 차수의 모델 파라미터를 얻을 수 있다.

본논문의 구성은 2절에서 모델 파라미터의 차수 감소를 추가한 MLLR알고리즘을 유도하고, 3절에서는 독립성분분석과 주성분분석을 이용해 모델 파라미터의 차수 감소를 설명한다. 4절에서 실험방법 및 결과를 설명하고, 5절에서 결론을 맺는다.

2. 파라미터 차수 감소를 적용한 MLLR

MLLR알고리즘의 목적은 각 mixture component의 평균 벡터의 변환을 추정하여, SI모델을 새로운 화자의 SD모델로 바꾸는 것이다. 평균 μ_s를 갖는 mixture component s에 대해 적응된 평균 $\hat{\mu}_s$는 다음과 같이 나타난다.

$$\hat{\mu}_s = P^{-1}W_s P\xi_s \tag{1}$$

여기서 W_s는 차수$m\times(m+1)$인 변환 행렬이고 P는 차수 $(m+1)\times(n+1)$인 차수감소 행렬이다.(단, $m<n$) ξ_s는 평균벡터를 확장한 것으로

$$\xi_s = \{w, \mu_{s1}, \mu_{s2}, \ldots, \mu_{sn}\}' \tag{2}$$

여기서 w는 offset항목 이다.

적응데이터를 $O=\{o_1, o_2, \ldots, o_T\}$라고 가정하자. 여기서 T는 관측 벡터의 개수이다.

또한, 모델 파라미터를 재추정하기 위한 auxiliary 함수는 다음과 같이 나타낼 수 있다.

$$Q(\lambda,\bar{\lambda}) = a - \frac{1}{2}F(O|\lambda)\sum_{j=1}^{S}\sum_{t=1}^{T}\gamma_j(t)[n\log(2\pi) + \log|\Sigma_j| + h(o_t, j)] \tag{3}$$

$$h(o_t, j) = (o_t - P^{-1}W_j P\xi_j)'\Sigma_j^{-1}(o_t - P^{-1}W_j P\xi_j) \tag{4}$$

$$\frac{d}{d\hat{W}_s}Q(\lambda,\bar{\lambda}) = F(O|\lambda)\sum_{t=1}^{T}\gamma_s(t)\Sigma_s^{-1}[o_t - P^{-1}W_s P\xi_s]\xi_s' P' \tag{5}$$

$$\sum_{t=1}^{T}\gamma_s(t)\Sigma_s^{-1}o_t\xi_s' P' = \sum_{t=1}^{T}\gamma_s(t)\Sigma_s^{-1}A_s P\xi_s\xi_s' P' \tag{6}$$

$Q(\lambda,\bar{\lambda})$를 $P^{-1}W_j$에 대해 일차미분을 행하면 (5)와 같고,(5)의 최대값을 구하면 (6)과 같다. 여기서 $A_s = P^{-1}W_s$이다. 식(6) 의 해를 구하는 자세한 방법은 C. J. Leggetter의 논문을 참고하라[1]. 따라서, 화자 적응의 성능과 연산량 감소는 차수감소행렬 P에 의존한다. 본 논문에서는 차수감소행렬P를 구하기 위해 주성분분석 및 독립성분분석을 이용하였다. 자세한 차수감소행렬 P를 구하는 방법은 다음 절에서 설명한다.

3. 파라미터 차수 감소

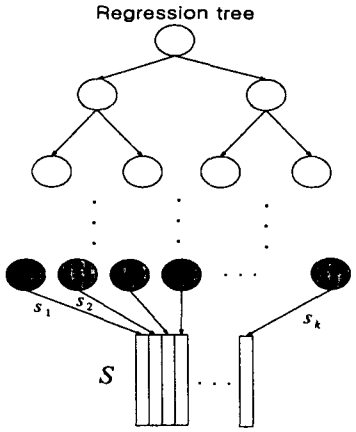

그림 1. Supervector matrix형성

본 논문에서는 파라미터의 차수 감소를 위해 주성분분석 및 독립성분 분석을 적용한다. 주성분 분석과 독립성분 분석을 적용하기 전에 먼저 SI모델의 평균값을 몇 개의 class로 그룹화하는 작업이 필요한데 본 논문에서는 MLLR방법을 수행하기 전에 만들어진 regression tree를 이용한다.

즉, 그림 1과 같이 regression tree의 base class의 각 평균 벡터들을 연결하여 하나의 supervector, s_i로 만든 후 이들을 하나의 column 벡터로 하는 matrix, $S=[s_1, s_2, \cdots, s_k]$를 형성한다. 여기서 k는 regression tree의 base class의 개수이다. 이와 같이 형성된 matrix S의 공분산 행렬을 구하여 주성분분석 또는 독립성분분석을 행한다.

주성분분석의 기본목표는 서로 상관성을 가지는 데이터들을 상관성이 없게 하는데 있다. 또한 데이터를 새로운 하위영역으로 사상하는 것이기 때문에 각 데이터의 차원을 줄일 수 있다. 수학적으로 말하면 데이터의 분산을 최대로 하는 축을 찾아서 축 변환에 의해서

각 데이터들의 상관성을 최대로 줄이는 것이다. 주성분분석을 통해 얻어진 고유 벡터 행렬 $V=[v_1,v_2,...,v_k]$을 이용하여 각 mixture component의 평균 벡터를 고유 벡터 영역으로 사상 시킬 수 있다. 이때, 고유값에 의거하여 사용하는 고유벡터의 수를 조절할 수 있다. 즉, 데이터의 분포 정도가 상대적으로 작은 고유 벡터는 사용하지 않음으로써 mixture component의 평균 벡터의 차수를 줄일 수가 있다. 따라서, m개의 고유 벡터를 사용한다면 차수감소행렬 P는 다음과 같이 나타낼 수 있다.

$$P=\begin{bmatrix}1 & 0\cdots 0\\ 0 & \\ \vdots & V'\\ 0 & \end{bmatrix} \qquad (6)$$

여기서 $V'=[v_1,v_2,...,v_m]$ 이다.($m<k$)

독립성분분석에 의해 찾은 성분은 데이터의 상관성을 줄일 뿐만 아니라 서로 독립인 성분을 찾는다. 즉, 독립성분분석이 주성분분석의 확장된 이론이라는 것을 알 수 있다[5]. 독립성분분석은 독립성분분석에 의한 결과 신호가 통계적으로 독립인 선형축을 찾는 방법이다. 주성분분석과 같은 상관관계에 의거한 변환 방법과 비교하였을 때, 독립성분분석은 신호의 상관관계--통계학적으로2차원적인 분석방법이다.-- 를 줄일 뿐 아니라, 더 높은 차원의 통계적 의존관계를 줄일 수 있다. 따라서, 일반적으 로 신호의 차원을 줄이는 데에는 독립성분 분석이 주성분 분석 보다 좋은 성능을 나타낸다. 그러나, 입력 데이터에 독립성분이 충분히 포함되어 있지 않을 경우, PCA가 ICA보다 좋은 성능을 나타낸다[5]. 독립성분분석 에 의해 얻어진 가중행렬 A를 이용하여 주성분분석에서와 마찬가지로 차수감소행렬 P를 구할 수 있다.

4. 실험 및 실험 결과

본 논문에서 사용한 data base는 DARPA Resource Management Continuous Speech Database (RM) 이다. 80명의 화자가 발성한 42개의 문장을 훈련 데이터로 사용하였다. 화자 적응용으로 12명의 화자로부터 각 화자마다 612개 의 문장을 사용하였다. 화자적응 성능 테스트용으로는 100개의 문장이 사용되었다. 사용한 음성의 특징 파라메타는 18차의 LPCC(Linear Predictive Cepstral Coefficient)와 delta계수를 사용하였고, 잡음 제거를 위해 CMS(Cepstral Mean Subtraction)을 행하였다.

음성인식은 49개의 monophone으로 훈련된 모델을 이용하여 행하였다. 각 monophone모델은 3개의 state를 가지고 있고 각 state는 8개의 mixture component를 포함한다. MLLR방법을 이용한 실험에서는 block의 개수가 2개인 block diagonal transform matrix를 사용하였고, 제안한 알고리즘에서는 full transform matrix를 사용하였다.

표1.최대 차수일 때 PCA와 ICA 및 MLLR의성능비교

# utterances	1	2	3	4
ICA(16)	92.9	95.1	95.9	96.5
PCA(36)	93.4	95.5	96.2	96.7
MLLR(36)	90.77	92.79	94.32	95.95
SD	92.12	92.12	92.12	92.12

표1은 PCA와 ICA 의 차수를 최대로 사용했을 때의 MLLR과의 단어 인식률 성능을 비교하였다. MLLR과 PCA는36차를 사 용하였으며, ICA는 16차를 사용하였다. PCA와 ICA 모두 MLLR에 비해 적응 속도가 빠른 것으로 나타났다. 특히, ICA방법은 MLLR과 비교하였을 때, MLLR의 절반의 차수 보다 작은 차수로 MLLR보다 높은 성능을 나타내었다.

표2는 차수에 따른 단어 인식률을 보이고 있다. ICA에서는 10차의 독립성분이, PCA에서 는 12차의 주성분이 MLLR과 유사한 성능을 나타내었다. 더 높은 차수의 주성분이나 독립성분이 사용될 때는 MLLR보 다 높은 인식률을 나타내었다. PCA방법 의 결과와 ICA방법의 결과를 비교해 보면 ICA방법이 PCA방법 보다 성능이 좋게 나타났다. 이는 독립성분분석이 통계학적으로 2차원적인 분석방법인 신호의 상관관계를 줄였을 뿐 아니라, 더 높은 차원의 통계적 의존관계를 줄였기 때문으로 사료된다. 또한, PCA방법은 분산이 상대적으로 적은 축의 고유 벡터는 사용하지 않기 때문에 차원의 감소에 의해 데이터의 손실을 초래하고, ICA는 차원의 감소에 의해 서로 비슷한 분포를 갖는 데이터는 같은 독립 성분으로 가정하므로 PCA방법에 비해 데이터의 손실을 적게 초래한 것으로 사료된다.

연산량에 대해 고려하면, 역행열 연산은 $O(n^3)$에 비례하므로 n회의 역행열 연산이 필요한 MLLR 화자적응을 위한 연산량은 $O(n^4)$에 비례한다. 본 실험에서는 하나의 변환 행렬을 구하기 위해 MLLR에서 역행열 연산은(36 + 1)* (36 + 1) 차원의 행열의 역행열 연산을 36회 요구하는 반면, 주성분분석을 이용한 화자적응에서의 역행렬 연산은 (12 +1)*(12+1) 차원의 행렬의 역행렬 연산을 12 회 요구하였고, ICA에서는

(10 +1)*(10+1) 차원의 행렬의 역행렬 연산을 10 회 요구하였다. 결과적으로 연산량은 기존의 MLLR알고리즘에 비해 연산량을 약 1/80~1/150 로 감소하였다. 제안한 알고리즘에서는 차원을 감소하기 위해 각 평균벡터 μ_s 와 차원감소행렬 P 의 행열의 곱 연산이 추가로 요구되지만 이는 $O(n^2)$에 비례하므로 전체 연산량에 큰 영향을 미치지 않는다. 또한, 차원감소 행렬을 구하기 위한 주성분 분석과 독립성분 분석에 요구되는 연산량은 SI모델을 훈련하는 과정에 추가 되므로 화자적응 시 요구되는 연산은 없다.

표2. 차수에 따른 단어인식률

	1	2	3	4
PCA(10)	90.7	92.2	93.8	95.1
PCA(12)	91.2	92.9	94.5	95.6
PCA(14)	91.5	93.7	94.9	96.01
PCA(16)	91.8	94.5	95.4	96.01
ICA(10)	91.2	92.6	94.1	95.9
ICA(12)	91.8	93.4	94.9	96.1
ICA(14)	92.4	93.9	95.4	96.2
ICA(16)	92.9	95.1	95.9	96.5
MLLR(36)	90.77	92.79	94.82	95.95
SD	92.12	92.12	92.12	92.12

5. 결론 및 향 후 과제

본 논문은 MLLR의 연산량 감소를 위해 주성분분석과 독립성분분석을 이용하여 모델의 차수를 줄임으로서 MLLR에서 요구하는 역행열 연산의 횟수를 줄였다. 주성분분석이나 독립성분분석에서 요구되는 연산량은 SI모델을 훈련할 때 요구되므로 화자적응 시 추가되지 않고 연산량 감소 행열을 곱하는 데에 요구되는 연산량은 역행열 연산에 비해 상대적으로 매우 적은 양이므로 영향을 미치지 않는다. 주성분분석과 독립성분분석을 행하면 차수의 감소 뿐만 아니라 각 성분끼리의 변별성이 높아진다. 즉, regression tree를 형성할 때 더 깊은 tree를 형성 할 수 있으므로 화자적응 성능을 더욱 높일 수 있다. 따라서, 주성분분석과 독립성분분석에 적합한 regression tree를 형성하는 연구가 요구된다.

참고문헌

[1] C. J. Leggetter, *Improved acoustic modelling for HMMs using linear transforms*, PhD Thesis, Univ. of Cambridge Feg. 1995.

[2] 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. on Signal Processing*, vol. 39, No. 4, April 1991, pp 806-814.

[3] O. Siohan, C. Chesta, and C. H. Lee, "Joint maximum a posteriori adaptation of transformation and HMM parameters," *IEEE Trans. on Speech and Audio Processing*, vol. 9, No. 4, May 2001, pp 417-428.

[4] J. T. Chien, " Online hierarchical transformation of hidden Markov models for speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol.7 No. 6, Nov. 1999, pp 656-667.

[5] Aapo Hyvarinen, Juha Karhunen, and Erkki Oja, *Independent Component Analysis*. Willy-Interscience.

[6] A. Sankar and C. H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. on Speech Audio Processing*, vol. 4, pp. 190-202, 1996.

[7] V. Digalakis, "On-line adaptaion of hidden Markov models using incremental estimation algorithms," *Proc. 5th Eur. Conf. Speech Communication and Technology*, Sept. 1997, vol. 4, pp. 1859-1862.