Subspace distribution clustering hidden Markov model을 위한 codebook design

조영규, 육동석
고려대학교 컴퓨터학과 음성정보처리 연구실

Codebook design for subspace distribution clustering hidden Markov model

Youngkyu Cho, Dongsuk Yook
Speech Information Processing Lab., Department of Computer Science and Engineering,
Korea Univ.

ccameo@voice.korea.ac.kr, yook@voice.korea.ac.kr

Abstract

Today's state-of-the-art speech recognition systems typically use continuous distribution hidden Markov models with the mixtures of Gaussian distributions. To obtain higher recognition accuracy, the hidden Markov models typically require huge number of Gaussian distributions. Such speech recognition systems have problems that they require too much memory to run, and are too slow for large applications. Many approaches are proposed for the design of compact acoustic models. One of those models is subspace distribution clustering hidden Markov model. Subspace distribution clustering hidden Markov model can represent original full-space distributions as some combinations of a small number of subspace distribution codebooks. Therefore, how to make the codebook is an important issue in this approach. In this paper, we report some experimental results on various quantization methods to make more accurate models.

Ⅰ. 서론

Hidden Markov model(HMM)은 연속 음성 인식을 위해 널리 사용되고 있다 [1]. HMM의 각 state는 Gaussian mixture로 이루어지는데 일반적으로 높은 인식 성능을 얻기 위해서는 많은 수의 Gaussian 분포가 필요하다. 이러한 모델은 많은 메모리 요구와 느린 수행 속도 때문에 많은 어플리케이션에 적용하기가 어렵다. 이를 해결하기 위해서 인식 성능의 저하 없이도 메모리 요구와 계산량을 줄일 수 있는 subspace distribution clustering hidden Markov model (SDCHMM)이 제안되었다 [2].

SDCHMM은 전체 multivariate Gaussian들을 subspace라는 feature space들로 나눈 뒤, 각 subspace에 속한 분포들을 quantize한다. SDCHMM은 quantize된 subspace distribution prototype들의 결합으로 원래의 full-space distribution을 표현한다. 이러한 결합 효과는 매우 강력한데, 예를 들어 128개의 prototype을 갖는 3-subspace SDCHMM 시스템은 128^3 = 2097152개의 서로 다른 full-space distribution을 표현할 수 있다

SDCHMM은 인식 시간 또한 줄일 수 있다. SDCHMM은 적은 수의 subspace Gaussian들이 많은 수의 full-space Gaussian들에 의해 공유되기 때문에 매 frame에서 단 한번 subspace Gaussian들의 log likelihood 값을 미리 계산해서 look-up table에 저장해 놓을 수가 있다. 이렇게 함으로써 full-space 만큼의 계산이 아닌 subspace Gaussian만큼의 계산이 필요해지며, 각 state에서는 단지 관계된 subspace들의 log likelihood 값을 조합해주기만 하기 때문에 계산량을 줄일 수 있다 [2].

위에서 언급한바와 같이 SDCHMM은 codeword가 full-space Gaussian에 의해 공유되기 때문에 높은 인식 성능을 보이기 위해서는 codeword의 quantization

error를 최소화 하는 것이 중요하다.

본 논문에서는 높은 인식 성능의 SDCHMM을 만들기 위하여 monophone 기반 및 vector quantization 방법을 이용한 codebook을 구성하여 인식 성능 비교 실험을 하였다.

본 논문의 구성은 2장에서 SDCHMM의 이론에 대하여 살펴보고, 3장에서는 실험에 사용된 codeword 구성 방법에 대하여 설명한다. 4장에서는 3장에서 설명한 codeword 구성 방법들을 이용한 SDCHMM의 인식 실험을 비교한다. 마지막으로 본 논문의 결론을 5장에서 기술한다.

Ⅱ. SDCHMM

본 장에서는 SDCHMM의 이론에 대하여 기술하고 SDCHMM의 이상적인 codeword에 대하여 알아본다.

1. SDCHMM의 개념

SDCHMM의 개념은 이미 만들어져 있는 continuous distribution HMM(CDHMM)으로부터 유도된다. CDHMM의 i번째 state의 관측 확률은 식 (1)과 같이 표현된다.

$$P_i^{CDHMM}(O)=\sum_{m=1}^{M} c_{im} N(O;\mu_{im},\sigma_{im}^2), \quad (1)$$

$P_i(O)$는 state i에서의 관측 데이터 O의 관측 확률이고, c_{im}은 state i의 m번째 Gaussian의 component weight이다. μ_{im}와 σ_{im}^2은 state i의 m번째 Gaussian의 mean과 variance이다.

CDHMM으로부터 K-subspace SDCHMM을 유도하기 위해서 CDHMM의 각 Gaussian들을 K-subspace로 구성하고 각 subspace의 Gaussian들을 tying하여 각 subspace의 full-space Gaussian들이 tying된 subspace Gaussian을 공유한다. SDCHMM의 i번째 state의 관측 확률은 식 (2)와 같다.

$$P_i^{SDCHMM}(O)=\sum_{m=1}^{M} c_{im}\left(\prod_{k=1}^{K} N^{tied}(O_k;\mu_{imk},\sigma_{imk}^2)\right), \quad (2)$$

O_k는 관측 데이터 O의 k번째 subspace이고, μ_{imk}와 σ_{imk}^2는 각각 state i의 m번째 Gaussian의 k번째 subspace에 대한 mean과 variance이다. CDHMM으로부터 K-subspace SDCHMM을 유도하기 위해서 원래의 subspace Gaussian은 가장 가까운 subspace Gaussian codeword로 근사화 된다. 따라서 SDCHMM은 full-space Gaussian이 아닌 K개의 subspace Gaussian codeword와 그것들을 가리키는 인덱스로 구성된다.

2. 분포 클러스터링

SDCHMM은 CDHMM의 각 subspace Gaussian들을 quantize해서 만들어질 수 있다. 다시 말해 full-space Gaussian distribution의 CDHMM은 K 개의 subspace로 구성되고, 각 subspace Gaussina들은 quantize되어 L개의 codeword로 만들어진다.

$$N^{quantized}(O_k;\mu_{kl},\sigma_{kl}^2) \qquad 1\le l\le L, 1\le k\le K \ . \quad (3)$$

각각의 CDHMM의 subspace Gaussian은 식 (4)와 같이 만들어진 codeword의 가장 가까운 subspace Gaussian으로 근사화 된다.

$$N(O_k;\mu_{imk},\sigma_{imk}^2)\approx N^{quantized}(O_k;\mu_{kl},\sigma_{kl}^2) \quad (4)$$

다시 말해 식 (5)와같이 두 Gaussian의 거리 측정 방법 $dist(\cdot)$를 이용하여 CDHMM의 subspace Gaussian과의 거리가 최소가 되는 codeword subspace Gaussian을 찾는다.

$$l=\underset{1\le q\le L}{\operatorname{argmin}} dist(N(O_k;\mu_{imk},\sigma_{imk}^2), N^{quantized}(O_k;\mu_{kl},\sigma_{kl}^2)) \quad (5)$$

따라서 좋은 성능의 SDCHMM을 만들기 위해서는 적절한 codeword의 구성 방법과 거리 측정 방법 $dist(\cdot)$을 사용해야 한다.

Ⅲ. Codeword 구성 방법

본 장에서는 실험에 사용된 codeword 구성 방법인 monophone을 기반으로 하는 방법과 modified k-means Gaussian 클러스터링 방법[2], LBG 클러스터링 방법[3]에 대하여 기술한다.

1. Monophone 기반

Monophone 기반 방법은 SDCHMM으로 구성하고자 하는 CDHMM을 학습시킨 데이터를 이용하여 monophone을 만들고 만들어진 monophone Gaussian 분포를 codeword로 사용하는 방법이다. 만들어진 monophone의 Gaussian 분포가 구성하고자 하는 L개의

codeword보다 많다면 Gaussian component weight가 가장 높은 L개의 Gaussian 분포를 codeword로 이용한다

2. Modified k-means Gaussian 클러스터링 방법

Modified k-means Gaussian 클러스터링 방법은 triphone CDHMM의 전체 Gaussian들을 이용하여 top-down 클러스터링을 하는 방법이다. Modified k-means Gaussian 클러스터링 방법은 i cluster에 속한 subspace Gaussian 분포들과 i cluster의 codeword 사이의 거리의 평균이 가장 큰 clsuter의 codeword를 split 함으로써 codeword의 개수를 증가시킨다. 알고리즘 1과 같은 순서로 수행된다.

Step1 : Initialization
1개의 subspace로 구성된 L개의 Gaussian mixture model을 학습시킨다. 학습된 L개의 Gaussian 분포들을 정의된 K개의 subspace로 구성한다. 각 subspace Gaussian 분포에 대하여 step2, setp3, step4를 반복한다.

Step2 : Classification
각 subspace Gaussian 분포들을 거리 측정 방법을 이용하여 가장 가까운 codeword로 분류한다.

Step3 : Update
분류된 각 cluster의 subspace Gaussian 분포들을 이용하여 codeword를 update 한다.

Step4 : Termination
Update된 codeword와 기존의 codeword를 비교하여 일정 threshold 값에 도달하면 멈추고, 그렇지 않으면 step2를 수행한다.

[알고리즘 1] Modified k-means Gaussian 클러스터링

Step2에서 사용하는 거리 측정 방법으로 두 Gaussian의 거리를 측정하는 Bhattacharyya distance를 사용한다 Bhattacharyya distance는 식(6)과 같이 정의된다.

$$D=\frac{1}{8}(\mu_2-\mu_1)^T\left[\frac{\Sigma_1+\Sigma_2}{2}\right]^{-1}(\mu_2-\mu_1) \quad (6)$$
$$+\frac{1}{2}\ln\frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{\sqrt{|\Sigma_1||\Sigma_2|}}.$$

μ_i : cluster i의 mean vector
Σ_i : cluster i의 covariance matrix

Bhattacharyya distance는 많은 음성 관련 실험에 사용되었으며, 좋은 성능을 보였다 [4]-[6].

3. LBG 클러스터링 방법

Modified k-means Gaussian 클러스터링 방법은 한 번에 1개씩 codeword를 증가시키는 반면 LBG 클러스터링 방법은 2의 배수만큼씩 codeword를 증가시킨다. 이 방법은 2의 배수만큼씩 codeword를 증가시키기 때문에 modified k-means Gaussian 클러스터링 방법보다 빠르지만 modified k-means Gaussian 클러스터링 방법보다 더 잘 클러스터링 할 수 없다

Ⅳ. 실험

본 논문에서 성능평가를 위해서 Resource Management(RM) 데이터베이스를 이용하여 실험 하였다. 실험을 위한 음성 벡터 추출 방법은 음성 신호에 대하여 25msec의 크기로 10msec씩 이동하면서 MFCC 12차와 에너지, 그리고 그것들의 1차 및 2차 미분값을 이용하여 39차 벡터열을 만들었다. CDHMM과 SDCHMM의 성능 비교 실험 결과는 [표 1]과 같다 실험은 state tying에 의해 1439개의 state로 구성된 CDHMM을 이용하였으며, 각 state의 Gaussian 개수는 CDHMM 실험을 통해 가장 좋은 성능을 보인 6개를 사용하였다. SDCHMM의 구성은 512개의 codeword를 사용하였고 가장 좋은 성능을 보인다고 알려진 모든 차원을 subspace로 나누는 39개의 subspace로 구성하였다. SDCHMM의 codeword는 monophone을 이용한 방법을 사용하였으며 CDHMM과 SDCHMM은 각각 6.1%와 5.9%의 인식 에러율을 보여 SDCHMM이 더 좋은 성능을 보임을 알 수 있었다.

	CDHMM	SDCHMM
전체 state 개수	1439	
각 state Gaussian 개수	6	
Phone	문맥 종속(triphone)	
Codeword 수		512
Subspace 수		39
에러율	6.1%	5.9%
HMM 메모리 요구량	2.8M	1.1M

[표 1] CDHMM과 SDCHMM의 성능 비교

또한 메모리 요구량에 있어서 CDHMM과 SDCHMM이 각각 2.8M와 1.1M로 SDCHMM이 CDHMM에 비해 메모리를 50%이상 줄일 수 있음을 알 수 있었다.

[표 2]는 3장에서 언급한 3가지 codebook 구성 방법인 monophon 기반, modified k-means 클러스터링 방법, LBG 클러스터링 방법을 이용하여 구성된 codeword를 이용하여 인식 에러율을 비교 실험한 결과이다.

	Monophon 기반	Modified k-means	LBG
에러율	5.9%	5.6%	5.7%

[표 2] monophon 기반, modified k-means 클러스터링, LBG 클러스터링 방법을 이용하여 구성된 codeword를 이용한 실험 결과

[표 2]에서와 같이 modified k-means 클러스터링을 이용하여 codeword를 구성한 것이 5.6%로 monophone 기반 방법인 5.9%와 LBG 클러스터링 방법을 이용한 5.7%에 비해 가장 좋은 성능을 보임을 알 수 있었다. 위 결과에서 보듯이 학습된 monophone을 codeword로 사용하는 것 보다 CDHMM의 모든 Gaussian 분포를 quantize하여 codeword를 구성하는 것이 더 좋은 성능을 보이며, 2의 배수로 codeword의 수를 증가시키는 LBG 방법보다 1개씩 codeword를 증가시켜나가는 modified k-means 클러스터링 방법이 더 좋은 성능을 보임을 알 수 있다.

V. 결론

일반적인 CDHMM의 많은 메모리 요구와 느린 수행 속도와 같은 문제점을 해결하기 위해 SDCHMM이 제안되었다. SDCHMM은 적은 수의 codeword가 full-space Gaussian에 의해 공유되기 때문에 높은 인식 성능을 보이기 위해서는 codeword의 quantization 에러를 최소화하는 것이 중요하다. 본 논문에서는 CDHMM과 SDCHMM의 비교 실험 및 monophone 기반, modified k-means 클러스터링, LBG 클러스터링 방법을 이용하여 구성된 codeword를 이용한 SDCHMM의 인식 실험을 하였다. 실험에서 SDCHMM의 메모리 요구량은 CDHMM의 메모리 요구량에 비해 50%이상 감소하는 것을 확인할 수 있었고, modified k-means 클러스터링을 이용하여 codeword를 구성한 방법이 5.6%의 에러로 monophone 기반 방법, LBG 클러스터링 방법을 이용해 codeword를 구성한 방법인 5.9%, 5.7%의 에러에 비해 더 좋은 성능을 보임을 알 수 있었다.

참고문헌

[1] L Gu and K. Rose, "Substate tying with combined parameter training and reduction in tied-mixture HMM design", *Proceedings, IEEE Transactions on Speech And Audio Processing*, vol. 10, pp. 137-145, 2002.

[2] E. Bocchieri, and B. K-W Mak, "Subspace distribution clustering hidden Markov model", *IEEE Transaction on speech and Audio Processing*, vol. 9, pp 264-276, 2001.

[3] X. Huang, A. Acero, and H-W. Hon, *Spoken language processing*, Prentice-Hall, 2001.

[4] P. C. Loizou and A. S. Spanias, "High-performance alphabet recognition", *IEEE Transactions on Speech And Audio Processing*, vol. 4, pp. 430-445, 1996

[5] B. Mak and E. Barnard, "Phone clustering using the Bhattacharyya distance", *International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2005-2008, 1996.

[6] L. Hirschman, "Multi-site date collection and evaluation in spoken language understanding", *International Conference on ARPA Spoken Language Technology Workshop*, pp. 19-24, 1993