The Korean Communications
in Statistics Vol. 8, No. 1, 2001
pp. 173-184

A Penalized Likelihood Method for Model Complexity Reduction in Gaussian Mixture Density[1)]

Sung M. Ahn[2)]

Abstract

We present an algorithm for the complexity reduction of a general Gaussian mixture model by using a penalized likelihood method. One of our important assumptions is that we begin with an overfitted model in terms of the number of components. So our main goal is to eliminate redundant components in the overfitted model. As shown in the section of simulation results, the algorithm works well with the selected densities.

Keywords: complexity reduction, Dirichlet distribution, EM algorithm, penalized likelihood

1. 서 론

유한혼합모형(finite mixture model)에서 모수를 추정할 때, 일반적으로는 성분(component)의 수가 주어졌다고 가정하지만, 근본적으로는 성분의 수도 추정을 해야 한다. 이 논문에서는 성분의 수를 추정하는 한 방법을 제시하고자 한다. 우선 문제를 제시하기 위해 사용할 모형은 다음과 같다. (1)에 있는 정규혼합모형 (Gaussian mixture model)을 이용하는 데, 추정해야 할 모수는 π_j, μ_j 그리고 $\Sigma_j, j=1,\cdots,g$ 들이다.

$$f(x;\pi,\mu,\Sigma)=\sum_{j=1}^{g}\pi_j N(x;\mu_j,\Sigma_j) \tag{1}$$

여기서 중요한 문제는 성분의 수인 g조차 추정을 해야 한다는 것이다. 본 논문에서는 문제를 단순히 하기 위해서 한 가지 가정을 하는데, 그것은 과대적합모형(overfitted model)이 존재한다는 것이다. 과대적합모형이라고 함은 필요이상으로 많은 수의 성분을 사용해 최적화된 모형을 뜻한다. 그 과대적합모형에서 필요 없는 성분을 제거하여 가장 적절한 수의 성분을 찾아 내는 접근을 시도하고자 한다.

1) 본 연구는 2001년 국민대학교 교내학술연구비 지원으로 이루어졌음
2) Assistant Professor, School of Management Information Systems, Kookmin University, Seoul 136-702
E-mail: sahn@kmu.kookmin.ac.kr

과대적합모형을 만들기 위해 사용한 방법은 Adaptive mixtures (Priebe, 1994) 인데, 다음의 절에서 간단히 본 논문의 근간이 되는 다른 이론과 함께 요약을 하겠다. 그리고 나서 네 번째 절에서 본 논문의 접근방법을 설명한 뒤, 마지막으로 시뮬레이션 결과를 제시하도록 하겠다.

2. 유한혼합 모형과 최대우도추정

유한혼합 모형에서는 (1)과 같이 분포를 가정한다. (1)에서, g는 알고 있다고 가정하고, π_j는 합해서 1이 되는 조건이 포함된다. 이 경우에 전통적인 최대우도추정(maximum likelihood estimation)을 사용하면 한번에 분포의 모수(π_j, μ_j 그리고 $\Sigma_j, j=1,\cdots,g$)를 찾아 낼 수는 없고, 반복적 절차 (iterative procedure)를 이용해야 가능하다. 이러한 반복적 절차 중의 하나가 EM 알고리즘인데, Dempster, Laird and Rubin (1977)에 따르면, 각 데이터 $x_i, i=1,2,\cdots,n$ 는 관측되지 않는 또 다른 데이터 $z_i, i=1,2,\cdots,n$ 와 대응된다고 가정한다. 여기서 z_i는 인디케이터 벡터 (indicator vector, $z_i=(z_{i1}, z_{i2},\cdots,z_{ig})^T$)로서 $z_{ij}=1$이 의미하는 바는 x_i가 j번째 성분에 의해 만들어 졌다는 것을 의미한다. 정규혼합 분포에서 x_i와 z_i의 결합분포는 다음과 같다. (Titterington *et al.*, 1985)

$$f(x_i, z_i|\Theta)=\prod_{j=1}^{g}[\pi_j N(x_i;\mu_j,\Sigma_j)]^{z_{ij}} \tag{2}$$

그러므로 로그우도함수는 다음과 같다.

$$l_c(\Theta)=\sum_{i=1}^{n}\sum_{j=1}^{g} z_{ij}\log[\pi_j N(x_i;\mu_j,\Sigma_j)] \tag{3}$$

여기서 z_{ij}를 모르기 때문에 (3)을 직접 사용할 수는 없으며, 대신에 z_{ij}의 기대값을 대입하여 사용한다.

정규혼합 모형에서 EM알고리즘은 다음과 같다. (Dempster *et al.*, 1977; Xu and Jordan, 1996)

• E-step (z_{ij}의 조건부 기대값)

$$\widehat{z}_{ij}^{(k+1)}=\frac{\widehat{\pi}_j^{(k)} N(x_i;\widehat{\mu}_j^{(k)},\widehat{\Sigma}_j^{(k)})}{\sum_{j=1}^{g}\widehat{\pi}_j^{(k)} N(x_i;\widehat{\mu}_j^{(k)},\widehat{\Sigma}_j^{(k)})} \tag{4}$$

• M-step (로그우도함수를 최대화하는 모수값)

$$\hat{\pi}_j^{(k+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ij}^{(k+1)}}{n}$$

$$\hat{\mu}_j^{(k+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ij}^{(k+1)} x_i}{\sum_{i=1}^{n} \hat{z}_{ij}^{(k+1)}} \tag{5}$$

$$\hat{\Sigma}_j^{(k+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ij}^{(k+1)} [x_i - \hat{\mu}_j^{(k)}][x_i - \hat{\mu}_j^{(k)}]^T}{\sum_{i=1}^{n} \hat{z}_{ij}^{(k+1)}}$$

EM알고리즘은 E-step과 M-step 을 로그우도함수의 값이 더 이상 개선되지 않을 때까지 반복 수행하는 기법이다.

3. Adaptive Mixtures

유한혼합모형에서는 성분의 수를 미리 안다고 가정하였으나, 이는 비현실적인 가정이다. 그래서 Adaptive mixture (Priebe, 1994)에서는 주어진 데이터에 근거하여 성분의 수를 추정하였다. 이는 성분이 하나인 모형으로 출발하여, 데이터를 한번에 하나씩 입력 받아서 필요할 때마다 성분을 하나씩 추가하는 방식을 사용하였다. 성분의 추가를 위한 결정은 새로운 데이터가 현재의 모형으로 얼마나 잘 설명되는지를 판단해서 이루어진다. 예를 들면, 새로운 데이터와 현재 모형에 존재하는 성분들과의 Mahalanobis 거리를 계산하여 그 중 최소의 거리가 어떤 기준치(threshold)를 초과하면, 새로운 성분을 모형에 추가한다. 그렇지 않으면 기존의 모형은 새로운 데이터를 사용해 우도함수의 값을 개선하는 방향으로 수정(update)된다. 자세한 내용은 Priebe (1994)에 나와 있다. Adaptive mixture 의 특징은 그것이 과대적합모형을 제시한다는 것이다. 서론에서 언급한 바와 같이 Adaptive mixture 에 의해서 도출된 모형은 본 논문에서 제시하는 알고리즘의 출발점이 된다.

4. 최대벌점가능도 추정(Maximum penalized likelihood estimation)

Adaptive mixture 를 사용하여 얻어진 모형은 과대적합모형(즉 성분의 수가 과대 추정된 모형)이라고 말한 바 있다. 그러므로 본 논문의 목적은 필요 없는 성분을 제거하는 것인데, 이를 위하여 최대벌점가능도(Maximum penalized likelihood) 방법을 이용하려고 한다. 그를 위하여 벌점항목(penalty term)이 기존의 로그우도함수에 추가되는데, 그 추가될 항목은 성분을 줄이게 하는 효과를 얻도록 해야 한다. 본 논문에서는 (6)과 같은 벌점가능도를 제안하고자 한다.

$$\sum_{i=1}^{n} \sum_{j=1}^{c} z_{ij} \log \pi_j N(x_i; \mu_j, \Sigma_j) + \lambda n \sum_{j=1}^{c} (\alpha_j - 1) \log \pi_j \tag{6}$$

(6)의 첫번째 항목은 (3)에 주어진 로그우도함수이며, 두 번째 항목이 새로 추가된 벌점항목으로서, 직관적인 의미는 다음과 같다. 벌점항목은 π_j의 값이 감소하면 할수록 커진다 (α_j의 값은 0에서 1사이라고 가정). 그러므로 π_j의 값이 0이 될 때 벌점항목의 값은 최대화 된다. 여기서 π_j의 값이 0이 되면 j번째 성분을 삭제할 수 있음을 의미한다. 다음의 소절에서 (6)이 어떻게 도출되었는지를 자세히 설명한다.

4.1 Maximum *a posteriori* (MAP) 추정

최대벌점가능도 방식은 MAP 추정과 동등한 의미를 가진다. Ormoneit and Tresp (1998)에서는 (7)과 같은 로그사후함수(log-posterior)를 이용한 MAP 추정을 제시하였다.

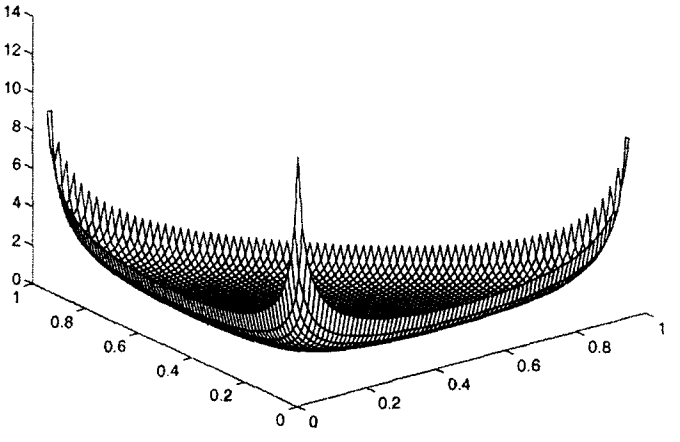
$$\begin{aligned} l_p(\Theta) = & \sum_{i=1}^{n}\sum_{j=1}^{g} z_{ij}\log \pi_j N(x_i;\mu_j,\Sigma_j) + \log D(\pi|\alpha) \\ & + \sum_{j=1}^{g}[\log N(\mu_j|\nu_j,\eta_j^{-1}\Sigma_j) + \log Wi(\Sigma_j^{-1}|\beta_j,\gamma_j)] \end{aligned} \tag{7}$$

(7)에서 $D(\pi|\alpha)$는 Dirichlet 분포를 가정한 π에 대한 사전분포(prior)를 나타내고, $N(\mu_j|\nu_j,\eta_j^{-1}\Sigma_j)$와 $Wi(\Sigma_j^{-1}|\beta_j,\gamma_j)$는 각각 μ_j와 Σ_j에 대한 사전분포로서 Gaussian과 Wishart 분포를 가정하였다. 본 논문에서는 논의를 간단히 하기 위하여 μ_j와 Σ_j에 대한 가정은 제외하고 π에 대한 가정만 중점적으로 고려하였다. 서론에서 언급한 바와 같이, Adaptive mixture로부터 만들어진 현재의 모형은 성분이 필요이상 많이 존재한다는 사전정보를 이용하는 것이다.

우선 Dirichlet 분포를 잠시 살펴보자. 확률벡터 $\pi=(\pi_1,\cdots,\pi_g)^T$는 (8)을 만족하면 $\alpha=(\alpha_1,\cdots,\alpha_g)^T$ $(\alpha_j>0;j=1,\cdots,g)$를 모수로 가지는 Dirichlet 분포를 가진다.

$$p(\pi|\alpha)=\frac{\Gamma(\alpha_1+\cdots+\alpha_g)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_g)}\pi_1^{\alpha_1-1}\cdots\pi_g^{\alpha_g-1},\ \pi_j>0,\ \sum_{j=1}^{g}\pi_j=1 \tag{8}$$

다차원에서 Dirichlet분포를 나타내기는 어렵지만, 성분이 3개 있는 모형에서 본 논문이 사용한 Dirichlet분포의 형태를 그림으로 표시하면 <그림 1>과 같다. <그림 1>에서 보듯이 각각의 π는 0이 될 확률이 높은 것이다. <그림 1>에서는 분포의 모수인 α_j가 모두 동일한 경우이며, 본 논문에서도 그 경우를 가정하였다. <그림 1>과 같은 볼록함수 형태의 Dirichlet분포는 α가 0보다 크거나 1보다 작다.



<그림 1> Dirichlet 분포 (α=.5)

그러므로 (7)에서 μ_j와 Σ_j에 대한 사전분포는 상수로 하고, π에 대한 사전분포를 Dirichlet분포를 가정하면 (9)를 얻을 수 있다.

$$\sum_{i=1}^{n}\sum_{j=1}^{g} z_{ij}\log \pi_j N(x_i;\mu_j,\Sigma_j)+\sum_{j=1}^{g}(\alpha_j-1)\log\pi_j \tag{9}$$

(9)를 극대화하는 모수의 값은 다음과 같다.

$$\widehat{\pi_j}=\frac{\frac{1}{n}\sum_{i=1}^{n} z_{ij}+\frac{1}{n}(\alpha_j-1)}{1+\frac{1}{n}(\sum_{j=1}^{g}\alpha_j-g)} \tag{10}$$

$$\widehat{\mu_j}=\frac{\sum_{i=1}^{n} z_{ij}x_i}{\sum_{i=1}^{n} z_{ij}}$$

$$\widehat{\Sigma_j}=\frac{\sum_{i=1}^{n} z_{ij}[x_i-\mu_j][x_i-\mu_j]'}{\sum_{i=1}^{n} z_{ij}} \tag{11}$$

(10)과 (11)은 EM 알고리즘으로 결과를 얻어낼 수 있다. 그런데 기대와는 달리 이를 이용한 결과는 만족스럽지 못한 것이었다. 시뮬레이션을 한 결과, 표본의 크기가 커짐에 따라 좋지않은 결과가 얻어졌다. 즉 표본의 크기가 커지면 알고리즘의 성분 제거능력이 떨어지는 것이었다. 그래서 우리는 (9)에 약간의 수정을 필요로 하게 되었다.

4.2 최대벌점가능도 추정

(9)를 살펴보면, 첫번째 항목은 우도함수(likelihood)의 의미대로 데이터를 통한 경험적 증거와 관련이 되어 있으며, 두 번째 항목은 우리의 사전적 믿음을 뜻한다. 즉 (9)가 베이지안의 관점에서 도출이 되었기 때문에 필연적으로 베이즈 추정값의 특징 (표본의 크기가 커짐에 따라 사전적 믿음의 영향이 점점 없어 지는 것) 을 가지게 된 것이다 . 그래서 표본의 크기가 결과에 영향을 미치지 못하도록 (9)에 약간의 수정을 한 결과가 (12)이다.

$$\sum_{i=1}^{n}\sum_{j=1}^{g} z_{ij}\log \pi_j N(x_i;\mu_j,\Sigma_j) + \lambda n\sum_{j=1}^{g}(\alpha_j-1)\log \pi_j \tag{12}$$

(12)를 보면 두 번째 항목에 n과 λ가 곱해졌는데, n은 표본의 크기이며, λ는 상수로서 성분의 제거 수준을 결정하는 수치이다. λ가 크면 상대적으로 많은 성분을 제거할 수 있게 된다. (12)를 이용하여 새로 계산한 π_j의 추정치는 다음과 같다

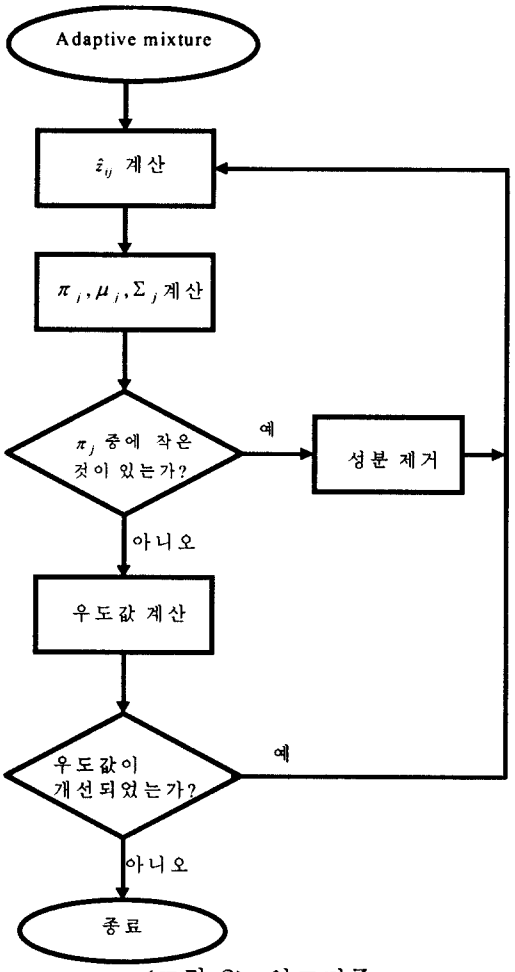
$$\widehat{\pi_j} = \frac{\frac{1}{n}\sum_{i=1}^{n} z_{ij} + \lambda(\alpha_j-1)}{1+\lambda(\sum_{j=1}^{g}\alpha_j - g)} \tag{13}$$

(13)에서 보듯이 $\widehat{\pi_j}$은 표본의 크기에 영향을 받지않는 추정치이다.

5. 시뮬레이션

5.1 알고리즘

알고리즘은 <그림 2>에 보여진 것과 같이 EM 알고리즘을 조금 수정한 것이다. EM 알고리즘과의 차이는 <그림 2>의 중앙부분에서 π의 크기를 확인하여 그것이 어떤 기준 값보다 작을 때 대응하는 성분을 제거하는 데에 있다. 본 논문의 시뮬레이션에서는 π의 값이 0.01보다 적으면 그에 해당하는 성분을 제거하였다. 즉 한 성분의 공헌도가 1%이하이면 그 성분은 무시해도 좋다는 가정을 하였다. z_{ij}의 기대값은 (4)를 이용해 구한다.



<그림 2> 알고리즘

5.2 시뮬레이션 결과

시뮬레이션을 위하여 Marron and Wand (1992)에 제시된 두개의 분포함수를 선택하였다. 선택된 분포함수는 'Asymmetric claw' 분포와 'Strongly skewed' 분포이며, 다음과 같이 정의된다:

$$\frac{1}{2}N(0,1)+\sum_{i=-2}^{2}(2^{1-i}/31)N(i+\frac{1}{2},(2^{-i}/10)^2)\text{: Asymmetric Claw}$$

$$\sum_{i=0}^{7}\frac{1}{8}N[3\{(\frac{2}{3})^i-1\},(\frac{2}{3})^{2i}]\text{: Strongly Skewed}$$

<그림 3>과 <그림 5>에서 분포의 형태와 성분의 구조를 알 수 있다. 성분 구조의 표현방식은 Solka *et al.* (1995)에 설명되어 있는데 간단히 설명하면 다음과 같다. 그림에서 각각의 원은 하나

의 성분을 설명하고 있는데, 각 원의 중심점의 $x-y$ 좌표는 해당하는 성분의 평균과 분산을 나타낸다. 원의 크기는 π의 상대적 크기를 나타낸다. 이 두개의 분포함수는 다른 분포들과 비교해서 상대적으로 근사하게 추정되기 어려운 이유로 선택이 되었다. Asymmetric claw 는 서로 다른 혼합비율을 가지는 성분들로 구성되어 있고, 그 중 혼합비율이 가장 큰 성분은 분산도 크기 때문에 그 영향력이 전체 분포에 미치는 형태로서 추정하기에 어려운 특징을 가진다 하겠다. 한편, Strongly Skewed는 혼합비율이 균등한 성분이 연속적으로 배열되어 있는 형태이라 성분을 추출하기 쉬워 보이지만, <그림 5>에서 보듯이 왼쪽 부분에 위치해 있는 성분들의 평균이 서로 매우 가까이 위치해 있으므로, 서로 다른 성분으로 분리 추정되기가 쉽지 않은 특성을 가진다.

먼저 Asymmetric claw 분포의 결과를 보자. <그림 3>에서 그 분포는 1개의 큰 성분과 5개의 작은 성분을 가지고 있다. 1개의 큰 성분은 분포의 가운데 부분에 위치해 있으며, 나머지 5개의 성분은 산재 되어 있다. 시뮬레이션에서 사용된 상수는 <표 1>과 같다.

<표1 > 시뮬레이션 상수

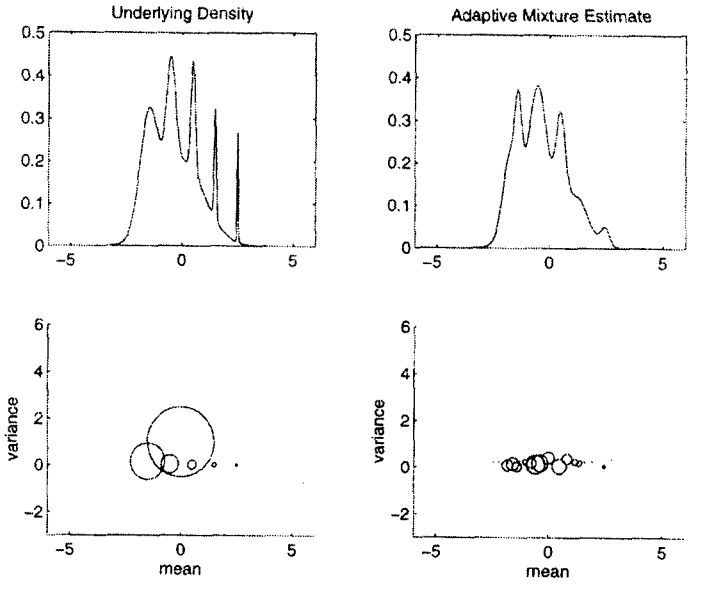
표본크기	4000
α	0.1
λ	0.002

<그림 3>의 우측부분이 Adaptive mixture로부터 나온 결과이며 그것은 26개의 작은 크기의 성분으로 이루어져 있는데, 이것이 본 논문에서 제안한 알고리즘의 시작점(initial solution)이 된다. <그림 4>가 결과를 보여주고 있는데, 왼쪽부분이 500번째 반복의 결과이며, 오른쪽 부분이 673번째 반복의 최종 결과이다. 즉 673번째 반복이후에는 더 이상 (12)의 값이 더 이상 개선되지 못했다. 그림에서 보여지듯이 본 논문의 결과는 원래의 분포와 상당히 근사함을 육안으로 알 수 있는데, 이를 (14) (IAE, integrated absolute error)를 사용한 측도로 추정오류를 계산해 보았다.

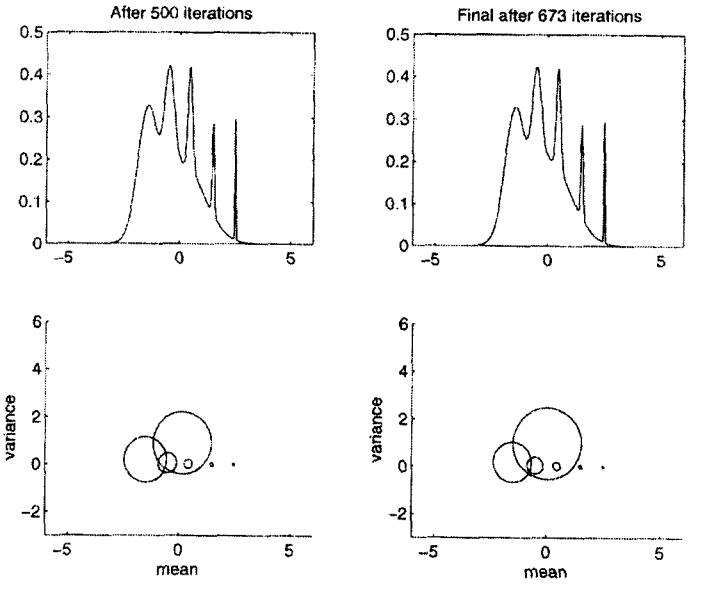
$$\int_R |\hat{f}(x) - f(x)| dx \tag{14}$$

이 경우에 추정오차는 0.0505로 계산되었다. 참고로 IAE의 최대값은 2이다.

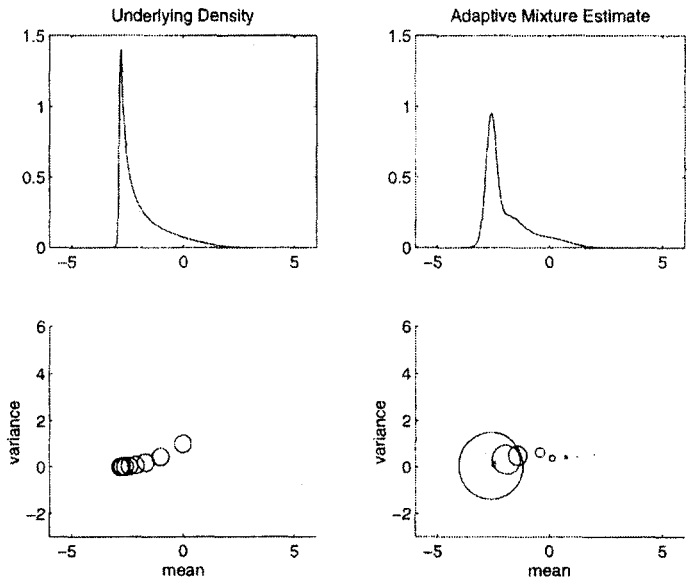
이번에는 Strongly skewed 분포의 경우를 보자. 이는 <그림 5>에 나와 있는데, 이 분포는 앞의 경우와 달리 성분의 구조를 분포 자체만 보아서는 쉽게 판단할 수 없다는 것이다. 이 경우는 동일한 크기의 성분 8개를 가진다. 이번에는 λ=0.001를 사용하여 시뮬레이션을 행하였다. <그림 5>의 우측부분에서 보여지듯이 Adaptive mixtures는 12개의 성분을 가지는 모형을 만들어 내었으며, 시뮬레이션 결과는 <그림 6>에 보여진다. 알고리즘은 1003번째 반복수행을 한 뒤에 끝났으며, 최종 결과는 <그림 6>의 우측부분에 보여지듯이 6개의 성분을 가지는 분포이다. 이 경우에 IAE는 0.0605이었다. Strongly skewed 분포는 성분이 잘 분리되지 않는 경우를 나타내는 좋은 예라고 할 수 있다. 이 경우에 본 논문의 알고리즘은 실제보다 2개 더 많은 성분을 제거하였다. 그러나 IAE를 근거해서 판단하면, 비교적 원래의 분포와 근사한 결과를 얻었다고 할 수 있다.



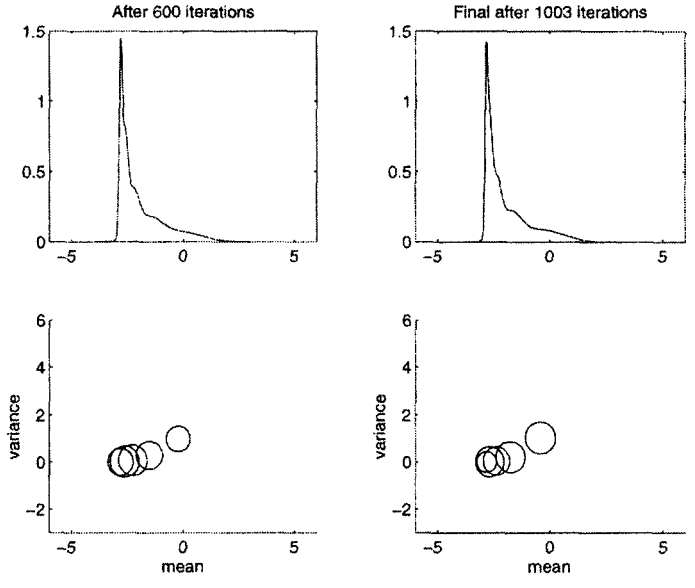
<그림 3> Asymmetric Claw 분포와 Adaptive Mixture 추정



<그림 4> 시뮬레이션 결과



<그림 5> Strongly Skewed 분포와 Adaptive Mixture 추정



<그림 6> 시뮬레이션 결과

<표 2>와 <표3>은 서로 다른 표본크기 (2000, 3000, 4000)에 따른 반복 결과를 보여주고 있다. 각각의 표본 크기에 대하여 10번씩 반복한 결과가 한 줄로 표시되어 있는데, <표 2>의 세 번째 줄의 경우는 10번 중 9번은 성분이 6개로 추정되었고 (0.0410의 그 9번의 경우의 IAE의 평균), 나머지 1번은 성분이 7개로 추정되었다는 의미이다. 평균 IAE의 결과에서 보여지듯, 본 논문의 알고리즘은 표본크기가 커질수록 오류가 적어짐을 알 수 있다.

<표 2> Asymmetric Claw 분포에 대한 결과

표본크기	반복수	성분의 수	평균 IAE
2000	10	5(1), 6(9)	0.1514, 0.0673
4000	10	6(9), 7(1)	0.0445, 0.0850
6000	10	6(9), 7(1)	0.0410, 0.0633

<표 3> Strongly Skewed 분포에 대한 결과

표본크기	반복수	성분의 수	평균 IAE
2000	10	4(3), 5(6), 6(1)	0.1147 , 0.0979, 0.0995
4000	10	5	0.0719
6000	10	5	0.0713

6. 요약 및 논의

지금까지 정규혼합분포에서 복잡성을 감소하는 알고리즘을 제시하였다. 시뮬레이션 결과에서 보여지듯이 이 알고리즘은 테스트된 두개의 분포에 대해서 IAE의 관점에서 보면 상당히 근사한 추정치를 제시하고 있다. 그러나 Strongly Skewed에 대해서는 원래의 성분 8개를 다 추출하지는 못하였다. Roeder and Wasserman (1997)도 표본의 크기가 1000일 때, 5개의 성분을 추출한 것으로 보고하고 있다.

한편 본 논문에서 벌점가능도 함수로 사용된 (12)에서 λ의 의미는 모형선택(model selection)을 위해서 사용되는 모수로서, 논의에 따라 hyperparameter (Bishop, 1995) 혹은 regularization parameter (Poggio and Girosi, 1990)라고 불리어 지며, 양의 실수 값을 가진다. λ의 역할은 그것이 사용된 항목의 가중치 혹은 중요도를 의미하며, λ는 최대우도방법을 통해서 추정될 수는 없고 일반적으로 모형 설계자의 의도에 따라 결정될 수 있다. 본 논문의 벌점가능도 함수인 (12)에 있는 2개의 항목 중, 첫번째 항목의 의미는 경험적으로 혹은 데이터를 통하여 판단할 수 있는 모수 값의 있을 법한 정도를 뜻하고, 두 번째 항목의 의미는 사전에 우리가 알고 있는 정보 (즉, 현재 필요 이상으로 많은 성분들이 있다는 사실) 가 모수 값에 영향을 주는 정도를 뜻한다. 그러므로 λ의 값을 크게 한다는 뜻은 우리의 사전적 믿음에 더 많은 가중치를 준다는 의미이며, 따라서 (12)를 최대화 함으로써 더 많은 성분을 제거하게 되는 결과를 가져온다. 본 논문에서는 <표 1>

에서 보여 지듯이 λ=0.002를 사용하였으며, 이는 여러 다른 값을 사용한 시뮬레이션을 통하여 적절하다고 생각되어 결정하였다.

마지막으로, 이 알고리즘의 단점이라고 할 수 있는 것은 표본의 크기가 커야지만 믿을만한 결과를 얻을 수 있다는 것인데, 이는 일반적으로 우도함수를 최대화하는 형태의 추정방법이 공통적으로 가지는 문제이기도 하다. 예를 들면, 본 논문의 시뮬레이션에서 표본크기가 1000인 경우에는 많은 경우에 IAE가 상당히 높았으며 또 비정칙성(singularity)이 자주 발생하여 그에 따른 추정모형의 성분의 증가를 초래하기도 하였다. 비정칙성은 EM 알고리즘을 통한 분포추정에서 자주 나타나는 문제인데, 본 논문의 시뮬레이션에서는 표본크기가 커짐에 따라 비정칙성의 발생횟수가 감소하였다.

References

[1] Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Oxford.

[2] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of Royal Statistical Society(B)*, 39, 1-38.

[3] Marron, J. S. and Wand, M. P. (1992), Exact Mean Integrated Squared Error, *Annals of Statistics*, 20(2), 712-736.

[4] Ormoneit, D. and Tresp, V. (1998), Averaging, Maximum Penalized Likelihood and Bayesian Estimation for Improving Gaussian Mixture Probability Density Estimates, *IEEE Transactions on Neural Networks*, 9(4), 639-649.

[5] Poggio, T. and Girosi. F. (1990), Networks for approximation and Learning, *Proceedings of the IEEE* 78 (9), pp. 1481-1497.

[6] Priebe, C. E. (1994), Adaptive Mixtures, *Journal of American Statistical Association*, 89(427), 796-806.

[7] Roeder, K. and Wasserman, L (1997), Practical Bayesian Density Estimation Using Mixtures of Normals, *Journal of American Statistical Association*, 92, pp. 894-902

[8] Solka, J. L., Poston, W. L., and Wegman, E. J. (1995), A Visualization Technique for Studying the Iterative Estimation of Mixture Densities, *Journal of Computational and Graphical Statistics*, 4, 180-198.

[9] Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985), *Statistical Analysis of Finite Mixture Distributions*, Wiley.

[10] Xu, L. and Jordan, M. I. (1996), On Convergence Properties of the EM Algorithm for Gaussian Mixtures, *Neural Computation*, 8, 129-151.