자기조직화 신경망을 이용한 정규혼합분포의 추정

안성만^{1,a}, 김명균^a

^a국민대학교 경영대학

요약

본 연구에서는 자기조직화 신경망이 필요한 노드만을 가지고 최적화하여 정규혼합분포를 추정하는 모형을 제안한다. 제안한 모형은 SOMN모형과 별점가능도를 사용한 모형을 결합한 것이다. SOMN의 장점은 수렴속도가 빠르고 표본의 크기가 작아도 발산하는 가능성이 낮다는 것이며, 별점가능도를 사용한 모형은 필요 없는 성분의 수를 줄일 수 있다는 것이다. 모의실험을 통하여 제안한 모형이 기대한 결과를 얻음을 확인하였다.

주요용어: 자기조직화 신경망, 정규혼합분포, EM알고리즘.

1. 서론

혼합분포는 단일분포(성분)를 결합함으로써 다양한 형태의 확률분포를 나타낼 수 있다. 혼합분포의 성분으로는 정규분포를 많이 사용하고 있으며, 이를 정규혼합모형이라고 부른다.

정규혼합모형의 모수를 추정하는 방법으로 대표적인 것이 우도를 사용하는 EM방법이다(Dempster 등, 1977). EM방법은 자료가 불완전하거나 결측치가 존재하는 경우에 반복적인 방식으로 우도를 최대화하여 모수를 추정하는 방법이다. 정규혼합모형에서 결측치는 각 관측치가 어느 성분에서 나왔는지에 대한 값이다. EM방법에서 우도를 계산하려면 우선 결측치의 기댓값(혹은 평균)을 구해야 한다(E-step). 다음에 그 기댓값을 적용한 우도를 최대화하여 모수를 추정한다(M-step).

EM알고리즘은 벡터의 기울기(gradient)를 사용하는 탐색적 해법의 일종이며, 매단계에서의 우도는 같거나 증가하게 된다. 그러나 단점으로는 수렴속도가 느리다는 것과 일반적으로 사용되는 묶음(batch) 알고리즘은 표본우도를 최대화하므로 표본크기가 작으면 일반화하기 어렵다는 것 등이 있다(Yin과 Allinson, 2001a). 이런 문제는 특히 표본크기가 작거나 실제 분포에서 성분이 잘 분리되어 있지 않을 때 잘 나타난다(Redner과 Walker, 1984; Xu와 Jordan, 1996).

이런 문제에 대한 해결방안으로 Yin과 Allinson (2001a)은 Kohonen의 SOM모형에 근거한 자기조직화 신경망(SOMN; self-organizing mixture network)를 제안하였다. SOMN은 SOM모형을 바탕으로 K-L거리(Kullback과 Leibler, 1951)와 확률적 근사법(Robbins와 Monro, 1951)을 사용하여 신경망을 최적화하는 모형으로서 EM방법보다 효율적으로 정규혼합모형의 모수를 추정한다.

한편, 정규혼합모형의 모수를 추정하는 경우에는 성분의 수도 함께 추정해야 하며, 성분의 수를 추정하는 하나의 방법으로 Ahn과 Baik (2011)은 별점가능도를 사용하였다. 별점가능도는 로그우도 함수가 발산하지 않도록 하거나 혹은 분포를 평활하게 하는 등의 목적을 위하여 사용할 수도 있지만(Hathaway, 1985; Ciuperca 등, 2003), Ahn과 Baik (2011)은 필요 없는 성분을 제거하기 위한 목적으로

본 연구는 국민대학교 2011년 연구비 지원에 의해 수행되었음.

¹ 교신저자: (136-702) 서울시 성북구 정릉동 861-1, 국민대학교 경영대학, 부교수. E-mail: sahn@kookmin.ac.kr

사용하여 성과가 있음을 보였다. 그렇지만 그 모형은 EM방법을 사용하므로 표본수가 작을 때 최적해로 수렴하지 못하는 단점이 있었다.

본 연구에서는 SOMN과 Ahn과 Baik (2011)의 모형(이후에서는 Ahn모형이라고 하겠다)의 장점을 결합하여 표본의 크기가 상대적으로 작아도 정규혼합모형에서 성분의 수를 효율적으로 추정하는 모형을 제안할 것이다. 그를 위하여 2절에서는 SOMN과 Ahn모형을 간략히 소개하고, 3절에서 본 연구의 모형을, 그리고 4절에서는 모의실험결과를 설명하고, 마지막으로 5절에서 요약 하도록 한다.

2. SOMN과 정규혼합모형에 대한 성분의 수

2.1. SOMN

유한혼합모형의 추정을 위한 신경망은 M 개의 입력노드(M 은 성분의 수와 같지 않아도 됨)로 구성되며, 각 노드는 하나의 정규분포 핵(Gaussian kernel)을 나타낸다. 핵의 모수는 평균벡터(μ_i)와 공분산행렬(Σ_i)이고 신경망의 출력은 정규분포의 핵을 혼합비율(P_i)로 선형결합한 혼합모형이며 혼합비율도 학습된다. 각 학습단계에서 신경망의 입력값은 표본에서 무작위로 선택되며 학습되는 노드는 승리한 노드, 즉 다음의 값이 가장 큰 노드와 그 이웃노드(η_c)이다.

$$\hat{P}(i|x) = \frac{\hat{P}_i \hat{p}_i(x|\hat{\theta}_i)}{\hat{p}(x|\hat{\Theta})}.$$

SOMN을 사용해서 확률분포의 추정을 하려면 노드의 개수를 실제 모형의 성분의 수보다 같거나 크도록 해야 한다. 그러나 경우에 따라서는 필요한 성분의 수를 정하여 그 수만큼 노드를 설정할 수도 있다.

실제 확률분포와 추정확률분포를 각각 $p(x)$ 와 $\hat{p}(x)$ 라고 하면, 두 함수의 Kullback-Leibler 거리는 다음과 같이 정의된다 (Kullback과 Leibler, 1951).

$$I = - \int \log \frac{\hat{p}(x)}{p(x)} p(x) dx.$$

이는 상대적 엔트로피로서 표본자료에서 확률분포추정을 위해 사용되고 남은 정보의 평균을 의미한다. 이 값은 항상 양수이며 추정확률분포가 실제확률분포와 같을 때 0이 된다. 표본우도(sample likelihood)는 표본확률의 시평균(time average)을 통하여 Kullback-Leibler 거리를 근사계산하며, 표본이 무한대이고 에르고딕 과정인 경우에 Kullback-Leibler 거리와 같아진다 (Yin과 Allinson, 2001a).

신경망에서 모수는 I 를 최소화하여 찾는다. 즉, I 를 각 모수에 대하여 편미분을 하고 그것을 0로 하여 Robins-Monro 법 (Robbins와 Monro, 1951)을 적용하면 다음과 같은 갱신 함수를 얻는다.

$$\begin{aligned} \hat{\theta}_{ij}(n+1) &= \hat{\theta}_{ij}(n) + \alpha(n) \left[\frac{1}{\hat{p}(x|\hat{\Theta})} \frac{\partial \hat{p}(x|\hat{\Theta})}{\partial \theta_{ij}(n)} \right] \\ &= \hat{\theta}_{ij}(n) + \alpha(n) \left[\frac{\hat{P}_i(n)}{\hat{p}(x|\hat{\Theta})} \frac{\partial \hat{p}_i(x|\hat{\theta}_i)}{\partial \theta_{ij}(n)} \right]. \end{aligned} \quad (2.1)$$

$$\begin{aligned} \hat{P}_i(n+1) &= \hat{P}_i(n) + \alpha(n) \left[\frac{\hat{p}_i(x|\hat{\theta}_i) \hat{P}_i(n)}{\hat{p}(x|\hat{\Theta})} - \hat{P}_i(n) \right] \\ &= \hat{P}_i(n) + \alpha(n) [\hat{P}(i|x) - \hat{P}_i(n)]. \end{aligned} \quad (2.2)$$

위에서 $\alpha(n)$ 은 학습률이고 $0 < \alpha(n) < 1$ 이며 지속적으로 감소한다. 정규혼합모형을 사용하여 식

(2.1)을 평균벡터와 공분산행렬에 대하여 표현하면 다음과 같다.

$$\hat{\mu}_i(n+1) = \hat{\mu}_i(n) + \alpha(n)\hat{P}(i|x)[x(n) - \hat{\mu}_i(n)], \quad (2.3)$$

$$\hat{\Sigma}_i(n+1) = \hat{\Sigma}_i(n) + \alpha(n)\hat{P}(i|x)\{[x(n) - \hat{\mu}_i(n)][x(n) - \hat{\mu}_i(n)]^T - \hat{\Sigma}_i(n)\}. \quad (2.4)$$

식 (2.2)–(2.4)를 사용하여 신경망의 모수를 찾을 수 있으며 결과적으로 정규혼합모형의 확률분포를 추정할 수 있게 된다. Yin과 Allinson (2001a)에 의하면 이 모형의 장점은 표본우도를 사용하여 묶음(batch)방식을 사용하는 EM알고리즘에 비하여 수렴속도가 빠르고 표본의 크기가 작아도 수렴하지 못하는 가능성이 낮다. 그렇지만 정규혼합모형의 성분의 수를 사전에 모르는 경우에는 많은 수의 노드(혹은 핵)를 사용하여야 하며 그렇게 되면 노드의 수가 많아져서 결과적으로 신경망이 비효율적이 된다. 다음에는 정규혼합모형에서 불필요한 성분을 수를 없애는 모형을 보도록 하겠다.

2.2. 정규혼합모형과 성분의 수

정규혼합모형에서 까다로운 문제는 성분의 수를 추정해야 하는 것인데 이를 위해서는 일반적인 로 그우도함수에 별점항목을 추가하여 별점가능도 추정을 하는 것이 하나의 방법이다. 이를 위하여 새로운 별점가능도를 정의한 것이 다음과 같다 (Ahn과 Baik, 2011).

$$\sum_{j=1}^n \sum_{i=1}^g z_{ij} \log P_i \Phi[x(j); \mu_i, \Sigma_i] + \lambda n \sum_{i=1}^g (\alpha - 1) \log P_i. \quad (2.5)$$

식 (2.5)에서 오른쪽에 있는 항이 추가된 별점항목이며, 이를 간단히 설명하면 다음과 같다. EM알고리즘을 사용하면 식 (2.5)를 최대화하는 모수의 값을 찾아내는데, 식 (2.5)에서 별점항목은 P_i 의 값이 0일 때 최대가 된다 ($0 < \alpha < 1, \lambda > 0$). 이때 $P_i = 0$ 이 의미하는 것은 i 번째 성분의 영향력이 0이라는 뜻이며, 결국은 식 (2.5)를 최대화함으로써 가능한 많은 수의 성분을 제거하게 된다. 모든 P_i 가 0이 되지 않는 이유는 $\sum P_i = 1$ 이라는 제약조건이 있기 때문이다. 위의 별점가능도를 최대화하는 모수를 찾으면 다음과 같다.

$$\hat{z}_{ij}^{(k+1)} = \frac{\hat{P}_i \hat{p}_i(x(j)|\hat{\theta}_i)}{\hat{p}(x(j)|\hat{\Theta})} \quad (2.6)$$

$$\hat{P}_i = \frac{1/n \sum_{j=1}^n \hat{z}_{ij}^{(k+1)} + \lambda(\alpha - 1)}{1 + \lambda g(\alpha - 1)} \quad (2.7)$$

$$\hat{\mu}_i = \frac{\sum_{j=1}^n \hat{z}_{ij}^{(k+1)} x(j)}{\sum_{j=1}^n \hat{z}_{ij}^{(k+1)}} \quad (2.8)$$

$$\hat{\Sigma}_i = \frac{\sum_{j=1}^n \hat{z}_{ij}^{(k+1)} [x(j) - \hat{\mu}_i^{(n)}][x(j) - \hat{\mu}_i^{(n)}]^T}{\sum_{j=1}^n \hat{z}_{ij}^{(k+1)}}. \quad (2.9)$$

식 (2.6)은 EM알고리즘의 E-step에 해당하는 것으로서 i 번째 데이터값이 어느 성분에서 나왔는지를 의미하는 기댓값을 계산한 것이다. 식 (2.7)–(2.9)는 EM알고리즘의 M-step에 해당하는 것으로서, 이 식을 사용하면 별점가능도를 최대화하는 모수를 계산할 수 있다. Ahn모형을 사용하면 필요 없는 성분을 제거할 수 있으며, 일치성과 시뮬레이션 결과는 Ahn은 Baik (2011)에서 확인할 수 있다.

Ahn모형의 단점은 표본크기가 작은 경우에 추정모형과 원래 분포의 오차가 커지는 경향이 있고, 또 비정칙성(singularity)이 자주 발생하여 그에 따른 추정모형의 성분의 증가를 초래한다는 것이다. 이는 일반적으로 표본우도함수를 최대화하는 추정방법이 공통적으로 가지는 문제이기도 하다.

3. 자기조직화 신경망과 성분의 수

앞에서 유한혼합모형에 대한 SOMN모형과 별점가능도를 사용한 모형을 보았다. SOMN의 장점은 수렴속도가 빠르고 표본의 크기가 작아도 발산하는 가능성이 낮다는 것이며, 별점가능도를 사용한 모형은 필요 없는 성분의 수를 줄일 수 있다는 것이다. 이제 이 두 모형의 장점을 가지는 모형을 제안하고자 한다. 즉 SOMN의 형태를 가지는 자기조직화 신경망이 필요한 노드만을 가지고 최적화 되는 모형을 구축할 것이다.

본 논문에서의 모형을 제시하기 전에 SOMN과 EM을 이용하는 정규혼합모형의 연관성을 살펴보기로 하자. EM을 이용하는 정규혼합모형에서는 일반적으로 묶음 알고리즘을 사용하지만 그것을 다음과 같이 재귀적(recursive) 알고리즘(혹은 온라인 알고리즘이라고도 부름)으로 표현할 수도 있다 (Titterton, 1984).

$$\hat{\mu}_i(n+1) = \hat{\mu}_i(n) + \frac{1}{n\hat{P}_i(n)} \hat{P}(i|x) [x(n) - \hat{\mu}_i(n)], \quad (3.1)$$

$$\hat{\Sigma}_i(n+1) = \hat{\Sigma}_i(n) + \frac{1}{n\hat{P}_i(n)} \hat{P}(i|x) \{ [x(n) - \hat{\mu}_i(n)] [x(n) - \hat{\mu}_i(n)]^T - \hat{\Sigma}_i(n) \}, \quad (3.2)$$

$$\hat{P}_i(n+1) = \hat{P}_i(n) + \frac{1}{n} [\hat{P}(i|x) - \hat{P}_i(n)]. \quad (3.3)$$

식 (3.1)–(3.3)과 SOMN의 갱신 알고리즘(식 (2.2)–(2.4))을 비교해 보면, SOMN에서 학습률로 정의된 $\alpha(n)$ 에 해당하는 부분이 다른 것을 알 수 있다. 이것은 Yin과 Allinson (2001a)에서도 언급하고 있는데, 학습률의 관점에서 보면 재귀적 알고리즘에서 μ 와 Σ 에 대한 학습률은 $1/(nP_i)$ 이며, P_i 에 대한 학습률은 $1/n$ 이다. 학습률은 n 이 커짐에 따라 감소하게 되는데, 이 성질은 SOMN에서도 마찬가지로 적용되지만 SOMN에서의 학습률인 $\alpha(n)$ 는 $1/(nP_i)$ 보다 훨씬 크다는 차이가 있다. 따라서 SOMN알고리즘의 수렴속도가 상대적으로 빠르다.

한편 Ahn모형의 해와 일반 유한혼합모형의 해와 비교했을 때 차이점은 P_i 를 추정하는 식임을 알 수 있다. 따라서 SOMN의 갱신 알고리즘에 Ahn모형에서 사용한 별점가능도의 기능을 추가하려면 P_i 를 재귀적 알고리즘의 형태로 변환하고 학습률을 수정하면 된다. P_i 를 재귀적 알고리즘의 형태로 변환하면 다음과 같다(부록 참조).

$$\hat{P}_i(n+1) = \hat{P}_i(n) + \frac{1}{n+1} \left[\frac{\hat{P}(i|x) - \lambda(1-\alpha)}{1 - g\lambda(1-\alpha)} - \hat{P}_i(n) \right]. \quad (3.4)$$

위의 식에서 $1/(n+1)$ 을 $\alpha(n)$ 으로 대체하면 다음과 같은 갱신 식을 얻는다. 식에서 λ 와 α 는 성분을 제거하는 강도를 나타내는 상수로서 $\lambda = 0.0012$ 그리고 $\alpha = 0.01$ 로 하였다.

$$\hat{P}_i(n+1) = \hat{P}_i(n) + \alpha(n) \left[\frac{\hat{P}(i|x) - \lambda(1-\alpha)}{1 - g\lambda(1-\alpha)} - \hat{P}_i(n) \right]. \quad (3.5)$$

따라서 본 연구에서 제시하는 모형은 정규혼합모형을 추정하는 자기조직화 신경망으로서 SOMN과 같은 망구조를 가지면서 불필요한 노드가 활성화되는 것을 방지하는 성질을 가진다. 이 모형에서 모수를 추정하는 갱신 알고리즘은 식 (2.3), (2.4), (3.5)로 정의된다.

4. 모의 실험

본 연구에서 제안한 모형을 성과를 비교하기 위하여 Xu와 Jordan (1996)에서 사용한 2차원 정규혼합분포를 사용해 보겠다. 분포는 세 개의 성분으로 이루어진 정규혼합분포이며 모수의 값은 표 1과 같

표 1: 실제 분포의 모수

P_1	P_2	P_3	μ_1	μ_2	μ_3	Σ_1		Σ_2		Σ_3	
0.345	0.32	0.335	2.5	-1.8	-0.5	4.0	-0.9	3.50	0.75	2.0	0.2
			1.0	2.2	-0.5	-0.9	0.3	0.75	0.30	0.2	0.3

그림 1: 실제 분포

그림 2: Ahn모형 추정의 예

으며, 분포를 3차원 공간에서 시각적으로 나타내면 그림 1과 같다. 하단의 그림에 있는 원은 성분의 평균(원의 중심)과 혼합비율(원의 상대적 크기)을 나타낸다.

표 1에서 정의된 정규혼합분포를 사용해서 1000개 크기의 표본을 만들고 본 연구에서 사용한 두 개의 모형(SOMN과 Ahn모형)을 사용하여 결과를 확인하고 본 연구에서 제안한 모형(편의상 SOMN-Ahn모형이라고 부른다)의 결과와 비교하기로 한다.

우선 Ahn모형은 같은 표본에 대하여는 항상 동일한 결과가 나오므로 10개의 서로 다른 표본에 대하여 모형에 적용하였다. Ahn모형은 초기 해에서 많은 수의 성분을 가지고 시작하여 필요 없는 성분을 제거(혼합계수의 비율이 0.03보다 작으면 제거하였다)하면서 로그우도의 값이 더 이상 개선되지 않을 때까지 반복한다. 서로 다른 표본에 대해 모형을 적용한 결과 최종 모형의 성분의 수는 각각 6, 4, 9, 3, 6, 10, 5, 8, 4, 6이었다. 각 실행에서 결과로 얻어진 모형의 성분의 수는 다르며 그 중에서 한번의 경우만 실제 분포와 동일한 수의 성분으로 이루어진 모형을 추정하였다. 그렇지만 성분의 수가 실제보다

그림 3: SOMN 추정의 예

그림 4: SOMN-Ahn 추정의 예

큰 모형도 추정된 모형의 전체적인 모양은 실제 모형과 비슷하였다. 그림 2는 그 중에서 6개의 성분으로 추정된 모형을 예로 든 것이다.

다음으로 SOMN을 적용해 보겠다. SOMN에서는 5개의 성분을 가진 정규혼합모형으로 추정을 하며 초기값으로 혼합계수는 1/5, 평균은 0을 기준으로 무작위로 선택한 값으로, 공분산행렬에서 분산은 8, 공분산은 0으로 하였다. 학습률은 Yin과 Allinson (2001b)에서 제시한 것과 같이 $\alpha(n) = \alpha_0/(1 + n/\tau)$ 에서 $\tau = 100$ 으로 하고 μ 에 대하여는 $\alpha_0 = 0.5$ 를 Σ 와 P 에 대하여는 $\alpha_0 = 0.1$ 을 사용하였다. 또 학습률은 표본크기의 역수보다 작아지지 못하게 하였다. 동일한 데이터에 대하여 10번 실행(각 실행은 40번의 epoch만큼 반복하였다)을 한 결과, 모든 경우에서 5개의 성분에 대하여 모수의 값을 추정하였고 혼합계수의 값이 0.05보다 작은 경우는 발생하지 않았다. 그림 3은 추정된 모형 중 하나를 예로 든 것이다.

이제 SOMN-Ahn모형의 실험결과를 보자. 이 모형의 기본적인 틀은 SOMN과 동일하므로 SOMN에서 사용한 것과 동일한 학습률을 적용하였으며, 동일한 표본(이 표본을 사용했을 때 Ahn모형은 6개

표 2: 표본크기가 2000일 때의 결과

모형	최종 모형의 성분의 수
Ahn모형	3, 3, 3, 4, 3, 3, 3, 3, 3
SOMN	5, 5, 5, 5, 5, 5, 5, 5, 5
SOMN-Ahn모형	3, 3, 3, 4, 3, 3, 3, 3, 4

표 3: 서로 다른 표본크기에 따른 결과

표본크기	최종 모형의 성분의 수	반복수 (epoch)	평균 실행속도 (초)
1000	3, 4, 3, 3, 3, 3, 3, 3, 3	100	48
500	3, 3, 3, 4, 3, 3, 3, 3, 4, 3	100	24
100	4, 3, 3, 4, 3, 4, 4, 3, 4, 4	200	10
50	3, 3, 4, 4, 2, 4, 3, 3, 4, 4	400	10

의 성분을 가지고 종료하였다)에 대해 모형을 10번 적용한 결과 최종 모형의 성분의 수는 각각 3, 3, 3, 5, 3, 4, 3, 4, 3, 4였다. 이 결과를 보면 10번의 실행 중에서 6번에서 실제 분포와 성분의 수가 일치하였음을 알 수 있다. 이 경우는 처음에는 다섯 개의 성분으로 시작하였다가 40번의 epoch만큼 반복되는 동안에 두 개의 성분의 혼합비율이 0이 되었음을 의미한다(실험에서는 혼합비율이 0.01보다 작아지면 0으로 간주하였다). 다른 세 번의 경우는 4개의 성분을 가지는 분포로 추정하였다. 비록 추정된 분포의 성분의 수가 3보다 큰 경우에도 실제 분포의 모양과 유사하게 추정하였다. 그림 4는 추정된 모형 중 3개의 성분을 가지는 것을 예로 든 것이다.

실험에서 다른 값은 그대로 두고 표본크기를 2000으로 하였을 때의 실험결과를 요약하면 표 2와 같다. 기대하였던 것과 같이 SOMN은 항상 5개의 유효한 성분을 유지하였음을 볼 수 있다. 그런데 Ahn모형은 표본의 크기가 커짐에 따라 정확도가 높아졌음을 알 수 있으며, 10번의 모의실험 중에서 9번에서 추정한 성분의 수가 실제 모형의 성분의 수와 일치하였다. 한편 SOMN-Ahn모형은 10번 중에서 8번을 동일하게 추정하였고 다른 2번의 경우에는 4개로 추정하였다. 이 두 개의 경우에서 가장 작은 혼합비율은 각각 0.0308과 0.0216로 매우 낮았다.

이제 SOMN-Ahn모형이 표본의 크기가 더 작아졌을 때 어떤 성과를 보이는지 확인해 보고자 한다. 이를 위하여 크기가 1000, 500, 100, 50인 표본을 각 경우에 대하여 10개씩 만들어 실험을 하였으며 그 결과는 표 3과 같다. 실험에서는 각 표본크기에 따라 서로 다른 반복횟수가 적용되었는데, 그 이유는 표본크기가 작아지면 상대적으로 많은 반복횟수에서 수렴하였기 때문이다. 그렇지만 표본크기가 크면 한번 반복하는 데 걸리는 시간이 길어지므로 평균실행속도는 표본크기가 커지면 늘어난다. 실험은 Intel 1.30Ghz CPU에서 실행되는 Matlab(버전 r2009b)으로 구현하였으며, 표 3에서 평균실행속도는 Matlab의 함수(tic과 toc)를 사용하여 측정되었다. 표에서 보면 표본크기가 500과 1000일 때 추정된 모형은 실제모형과 성분의 수가 많은 경우에 일치함을 알 수 있다. 그리고 추정된 성분의 수가 3이 아닌 경우에도 추정된 분포의 모양은 실제 분포와 근사하였다. 한편 표본의 크기가 100인 경우에는 4번의 실험에서 추정된 모형의 성분의 수가 3이 되었으며 나머지 6번에서는 4개로 추정하였으며 표본의 크기가 50인 경우에도 유사한 결과를 얻었다. 추정된 성분의 수를 보면 표본크기가 50과 100일때의 추정된 모형의 성과가 비슷해 보이지만 실제 추정된 분포는 차이가 있는데, 그것을 그림 7과 그림 8을 통하여 확인할 수 있다.

SOMN-Ahn모형이 필요 없는 성분을 제거하는 과정을 예시하기 위하여, 각 표본크기에 대하여 처음 4번의 실험결과를 그림 5~그림 8에서 보였다. 그림에서 왼쪽 그래프는 5개의 성분에서 혼합비율(P_i)의 값이 변화하는 과정을 보여주며, 오른쪽의 그래프는 추정된 분포를 3차원상에 그린 것이다. 그림 5에서 보면 두 번째 실험결과에서는 100epoch 후에도 여전히 한 개의 P_i 만 제거되었음을 볼 수 있

그림 5: 표본크기가 1000일 때의 실험결과

그림 6: 표본크기가 500일 때의 실험결과

다. 그렇지만 추정된 분포의 모양은 전반적으로 실제분포와 근사함을 알 수 있다. 그림 6에 있는 결과를 보면 표본크기가 500인 경우에도 추정된 모형은 모든 경우에서 실제분포와 유사함을 알 수 있다. 표본크기가 1000일 때와의 차이는 수렴하는 epoch수가 좀 길어진 것이다. 이제 그림 7을 보면 두 개의

그림 7: 표본크기가 100일 때의 실험결과

그림 8: 표본크기가 50일 때의 실험결과

경우는 성분의 수를 올바르게 추정하였고, 나머지 두 개의 경우는 그렇지 못하며 이 경우는 추정된 분포도 실제 분포와 모양이 조금 다를 수 있다. 마지막으로 표본크기가 50인 그림 8을 보면 추정된 성분의 수는 3개 혹은 4개로 그림 7의 결과와 유사하지만 추정된 모수의 값이 실제분포의 값과 차이가 있어서, 그 결과 추정된 분포의 모양이 실제분포와 많이 다를 수 있다.

모의실험결과를 종합하면 SOMN-Ahn모형은 SOMN과 Ahn모형의 장점을 가지면서 단점을 보완하고 있음을 확인할 수 있다. 즉 SOMN-Ahn모형은 정규혼합모형의 모수를 자기조직화 신경망의 형태로 추정을 하면서 상대적으로 작은 크기의 표본에서도 필요 없는 성분을 제거할 수가 있다. 표본크기를 세분화해서 실험을 한 결과, 실험에서 사용한 분포에 대하여는 표본크기가 1000과 500인 경우에는 추정된 모형의 성과가 좋았으나 표본의 크기가 더 작아지면 성과가 낮아져서 50일 때는 신뢰하기 어려운 결과가 나왔다.

5. 요약 및 맺음말

본 연구에서는 자기조직화 신경망이 필요한 노드만을 가지고 최적화 되는 모형을 제안하였다. 제안한 모형은 SOMN모형 (Yin과 Allinson, 2001a)과 별점가능도를 사용한 모형 (Ahn과 Baik, 2011)을 결합한 것이며, SOMN의 장점은 수렴속도가 빠르고 표본의 크기가 작아도 발산하는 가능성이 낮다는 것이며, 별점가능도를 사용한 모형은 필요 없는 성분의 수를 줄일 수 있다는 것이다. 제안한 모형은 이 두 모형의 장점을 가지는 모형으로서, SOMN의 형태를 가지는 자기조직화 신경망이 필요한 노드만을 가지고 최적화 되는 장점을 가지게 되었다.

모의실험결과에 따르면 제안한 모형은 SOMN모형과 별점가능도를 사용한 모형의 장점을 가지면서 단점을 보완하고 있음을 확인할 수 있다. 즉 제안한 모형은 정규혼합모형의 모수를 자기조직화 신경망의 형태로 추정을 하면서 상대적으로 작은 수의 표본에서도 필요 없는 성분을 제거할 수가 있었다.

부록:

P_i 를 재귀적 알고리즘의 형태로 변환하면 다음과 같다.

Ahn과 Baik (2011)에서 $\hat{P}_i(n)$ 은 다음과 같이 정의된다.

$$\hat{P}_i(n) = \frac{1/n \sum_{j=1}^n \hat{P}(i|x(j)) - \lambda(1 - \alpha)}{1 - g\lambda(1 - \alpha)}.$$

이것을 다시 정리하면 다음과 같다.

$$\hat{P}_i(n) = \frac{1}{1 - gk} \left(\frac{1}{n} \sum_{j=1}^n z_{ij} - k \right).$$

단, $z_{ij} = \hat{P}(i|x(j))$ 그리고 $k = \lambda(1 - \alpha)$.

그러면 $\hat{P}_i(n+1)$ 은 다음과 같이 정의된다.

$$\hat{P}_i(n+1) = \frac{1}{1 - gk} \left(\frac{1}{n+1} \sum_{j=1}^{n+1} z_{ij} - k \right). \quad (\text{A.1})$$

식 (A.1)의 오른쪽 괄호속에 처음 항을 다시 쓰면 다음과 같다.

$$\begin{aligned} \frac{1}{n+1} \sum_{j=1}^{n+1} z_{ij} &= \frac{1}{n+1} \left(\sum_{j=1}^n z_{ij} + z_{i(n+1)} \right) \\ &= \frac{1}{n+1} \left(\frac{n+1-1}{n} \sum_{j=1}^n z_{ij} + z_{i(n+1)} \right) \\ &= \frac{1}{n+1} \left(\frac{n+1}{n} \sum_{j=1}^n z_{ij} + z_{i(n+1)} - \frac{1}{n} \sum_{j=1}^n z_{ij} \right) \end{aligned}$$

$$= \frac{1}{n} \sum_{j=1}^n z_{ij} + \frac{1}{n+1} \left(z_{i(n+1)} - \frac{1}{n} \sum_{j=1}^n z_{ij} \right). \quad (\text{A.2})$$

이제 식 (A.2)를 이용하여 식 (A.1)을 다시 쓰면 다음과 같다.

$$\begin{aligned} \hat{P}_i(n+1) &= \frac{1}{1-gk} \left\{ \frac{1}{n} \sum_{j=1}^n z_{ij} + \frac{1}{n+1} \left(z_{i(n+1)} - \frac{1}{n} \sum_{j=1}^n z_{ij} \right) - k \right\} \\ &= \frac{1}{1-gk} \left(\frac{1}{n} \sum_{j=1}^n z_{ij} - k \right) + \frac{1}{1-gk} \frac{1}{n+1} \left(z_{i(n+1)} - \frac{1}{n} \sum_{j=1}^n z_{ij} \right) \\ &= \hat{P}_i(n) + \frac{1}{n+1} \frac{1}{1-gk} \left(z_{i(n+1)} - k - \frac{1}{n} \sum_{j=1}^n z_{ij} + k \right) \\ &= \hat{P}_i(n) + \frac{1}{n+1} \left\{ \frac{1}{1-gk} (z_{i(n+1)} - k) - \frac{1}{1-gk} \left(\frac{1}{n} \sum_{j=1}^n z_{ij} - k \right) \right\} \\ &= \hat{P}_i(n) + \frac{1}{n+1} \left\{ \frac{z_{i(n+1)} - k}{1-gk} - \hat{P}_i(n) \right\}. \end{aligned} \quad (\text{A.3})$$

식 (A.3)을 $z_{ij} = \hat{P}_i(x(j))$ 와 $k = \lambda(1-\alpha)$ 를 이용하여 다시 쓰면 다음과 같다.

$$\hat{P}_i(n+1) = \hat{P}_i(n) + \frac{1}{n+1} \left[\frac{\hat{P}_i(x) - \lambda(1-\alpha)}{1-g\lambda(1-\alpha)} - \hat{P}_i(n) \right].$$

참고 문헌

- Ahn, S. and Baik, S. W. (2011). Estimating parameters in multivariate normal mixtures, *The Korean Communications in Statistics*, **18**, 357–366.
- Ciuperca, G., Ridolfi, A. and Idier, J. (2003). Penalized maximum likelihood estimator for normal mixtures, *Scandinavian Journal of Statistics*, **30**, 45–59.
- Dempster, A., Laird, N. and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society: Series B*, **39**, 1–38.
- Hathaway, R. J. (1985). A constrained formulation of maximum likelihood estimation for normal mixture distributions, *Annals of Statistics*, **13**, 795–800.
- Kullback, S. and Leibler, R. A. (1951). On information and sufficiency, *The Annals of Mathematical Statistics*, **22**, 79–86.
- Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum Likelihood and the EM algorithm, *SIAM Review*, **26**, 195–239.
- Robbins, H. and Monro, S. (1951). A stochastic approximation method, *The Annals of Mathematical Statistics*, **22**, 400–407.
- Titterton, D. M. (1984). Recursive parameter estimation using incomplete data, *Journal of the Royal Statistical Society: Series B*, **46**, 257–267.
- Yin, H. and Allinson, N. (2001a). Self-organizing mixture networks for probability density estimation, *IEEE Transactions on Neural Networks*, **12**, 405–411.
- Yin, H. and Allinson, N. (2001b). Bayesian self-organizing map for Gaussian mixtures, *IEE Proceedings - Vision, Image, and Signal Processing*, 234–240.

Xu, L. and Jordan, M. I. (1996). On convergence properties of the EM algorithm for Gaussian mixtures, *Neural Computation*, **8**, 129–151.

2011년 7월 접수; 2011년 10월 채택

A Self-Organizing Network for Normal Mixtures

SungMahn Ahn^{1,a}, Myeong-Kyun Kim^a

^aCollege of Business Administration, Kookmin University

Abstract

A self-organizing network is designed to estimate parameters of normal mixtures. SOMN achieves fast convergence and low possibility of divergence even when sample sizes are small, while PMLE eliminate unnecessary components. The proposed network effectively combines the good properties of SOMN and PMLE. Simulation verifies that the proposed network eliminates unnecessary components in normal mixtures when sample sizes are relatively small.

Keywords: Self-organizing network, normal mixtures, EM algorithm.

This research was supported by the research fund of Kookmin University in 2011.

¹ Corresponding author: Associate Professor, College of Business Administration, Kookmin University, 861-1, Jeongneung-dong, Seongbuk-gu, Seoul 136-702, Korea. E-mail: sahn@kookmin.ac.kr