X-means 확장을 통한 효율적인 집단 개수의 결정

허경용* · 우영운**

Extensions of X-means with Efficient Learning the Number of Clusters

Gyeongyong Heo* · Young Woon Woo**

요 약

K-means는 알고리즘의 단순함과 효율적인 구현이 가능함으로 인해 군집화를 위해 현재까지 널리 사용되는 방법 중 하나이다. 하지만 K-means는 집단의 개수가 사전에 결정되어야 하는 근본적인 문제점이 있다. 이 논문에서는 BIC(Bayesian information criterion) 점수를 이용하여 효율적으로 집단의 개수를 추정할 수 있는 X-means 알고리즘을 확장한 두 가지 알고리즘을 제안한다. 제안한 방법은 기본적으로 X-means 방법을 따르면서 집단이 임의의 분산 행렬을 가질 수 있도록 함으로써 X-means 알고리즘이 원형 집단만을 허용함에 따른 over-fitting을 개선한다. 제안한 방법은 하나의 집단에서 시작하여 계속해서 집단을 나누어가는 하향식 방법으로, BIC score를 최대로 증가시키는 집단을 분할해 나간다. 제안한 알고리즘은 Modified X-means(MX-means)와 Generalized X-means(GX-means)의 두 가지로, 전자는 K-means 알고리즘을, 후자는 EM 알고리즘을 사용하여 현재 주어진 집단들에서 최적의 분할을 찾아낸다. MX-means는 GX-means보다 그 속도에서 앞서지만 집단들이 중첩된 경우에는 올바른 집단을 찾아낼 수 없는 단점이 있다. GX-means는 실행 속도가 느린 단점이 있지만 집단들이 중첩된 경우에도 안정적으로 집단들을 찾아낼 수 있다. 이러한 점들은 일련의 실험을 통해서 확인할 수 있으며, 제안한 방법들이 기존의 방법들에 비해 나은 성능을 보임을 확인할 수 있다.

ABSTRACT

K-means is one of the simplest unsupervised learning algorithms that solve the clustering problem. However K-means suffers the basic shortcoming: the number of clusters k has to be known in advance. In this paper, we propose extensions of X-means, which can estimate the number of clusters using Bayesian information criterion(BIC). We introduce two different versions of algorithm: modified X-means(MX-means) and generalized X-means(GX-means), which employ one full covariance matrix for one cluster and so can estimate the number of clusters efficiently without severe over-fitting which X-means suffers due to its spherical cluster assumption. The algorithms start with one cluster and try to split a cluster iteratively to maximize the BIC score. The former uses K-means algorithm to find a set of optimal clusters with current k, which makes it simple and fast. However it generates wrongly estimated centers when the clusters are overlapped. The latter uses EM algorithm to estimate the parameters and generates more stable clusters even when the clusters are overlapped. Experiments with synthetic data show that the purposed methods can provide a robust estimate of the number of clusters and cluster parameters compared to other existing top-down algorithms.

키워드

K-means, BIC, EM, 가우시안 혼합 모델

K-means, BIC, EM, Gaussian Mixture Model

* Dept. of Computer and Information Sci. and Eng., University of Florida

접수일자 : 2007. 12. 20

** 동의대학교 멀티미디어공학과(교신저자)

Ⅰ. 서 론

K-means[1]는 주어진 데이터를 K 개의 그룹(group) 또는 집단(cluster)으로 나누는 비지도 군집화(unsupervised clustering) 알고리즘으로, 단순한 구조를 가지면서도 국부 최대값에 수렴하는 것을 보장하기 때문에 현재에도 널리 사용되고 있다. 하지만 K-means 알고리즘은 집단의 개수가 사전에 결정되어야 하는 기본적인 문제점이 있다. 실제 문제에서는 항상 K 값을 사전에 결정할 수 있는 것이 아니므로 사전 지식에 기초하여 K 값을 결정하는 것이 최선이지만 이러한 정보 또한 항상 주어지지는 않는다.

이 논문에서는 기존의 X-means[2] 알고리즘을 확장한 두 개의 알고리즘, MX-means (Modified X-means)와 GX-means (Generalized X-means)를 제안한다. X-means 알고리즘은 하나의 집단에서 시작하여 반복적으로 집단을 분할해 나가는 하향식 알고리즘으로 집단의 분할을 위한 기준으로 BIC(Bayesian Information Criterion) 점수(score)를 사용한다. X-means 알고리즘이 효율적으로 데이터를 묘사하는 가우시안 컴포넌트의 개수를 추정할 수 있지만, X-means는 기본적으로 모든 집단들이 동일한 대각 행렬의 분산 행렬을 가진다고 가정함으로써 원형이 아닌 가우신안 컴포넌트의 경우 심각한 과대적합(over-fitting)이 발생하는 문제점이 있다. 제안한 알고리즘은 X-means와 마찬가지로 가우시한 형태를 가지는 집단의 개수와 파라미터 값을 자동으로 결정해 주지만, X-means의 분산 행렬에 대한 제한을 없앰으로써 타원형의 가우시안 집단에서 심각한 과대적합이 발생하는 문제점을 해결한다. 또한 이들 알고리즘은 X-means와 마찬가지로 각기 K-means와 EM(Expectation Maximization) 알고리즘의 wrapper 알고리즘이다. MX-means는 K-means 알고리즘을 사용하여 현재 집단의 개수 k가 주어졌을 때 최적의 분할을 찾아낸다. 하지만 K-means의 hard assignment는 집단들이 겹쳐진 경우 정확한 집단의 형태를 찾아낼 수 없는 문제점이 있다. GX-means 알고리즘은 EM 알고리즘을 사용하여 soft assignment를 수행함으로써 이러한 문제점을 해결한다. GX-means는 실행속도가 느린 단점이 있지만 MX-means에 비해 안정적으로 집단의 개수와 파라미터 값을 찾아낼 수 있으며, MX-means 또한 기존의 X-means에 비해 나은 성능을 보여주며 집단들이 겹쳐져 있지 않은 경우에는 GX-means와 유사한 성능을 보여준다. 두 알고리즘은 기본적으로 X-means와 마찬가지로 하나의 집단에서 시작하여 BIC 점수를 최대로 하는 집단의 분할을 반복적으로 시행해 나가는 하향식 알고리즘이다. 집단의 병합을 이용하는 상향식 알고리즘들도 여러 가지 존재하지만, 상향식 알고리즘의 경우 최대 집단의 개수를 지정해 주어야 하며 하향식 알고리즘에 비해 초기 집단의 설정에 민감한 단점이 있다. 따라서 이 논문에서는 집단의 개수를 추정하는 기존의 하향식 알고리즘만을 대상으로 비교 실험하였다. 알려진 하향식 알고리즘으로는 X-means[2], G-means (Gaussian means)[3], PG-means (Projected Gaussian means)[4], BK-means (Bayesian K-means)[5] 등이 있다. 제안한 두 알고리즘은 X-means와 G-means에 비해 나은 성능을 보였으며, 특히 GX-means는 다른 모든 알고리즘에 비해 비슷하거나 나은 성능을 보였다. 또한 MX-means, GX-means 알고리즘은 다른 알고리즘이 몇 개의 파라미터 설정을 필요로 하는 것과 달리 추가적인 파라미터를 필요로 하지 않는 장점이 있다.

Ⅱ. 관련 연구

데이터가 주어졌을 때 집단의 개수 k를 자동으로 결정하기 위해 지금까지 많은 알고리즘이 제시되었다. 기존 알고리즘의 대부분은 고정된 k 값에서 최적의 집단을 찾아내는 K-means나 EM 알고리즘의 wrapper 알고리즘으로 특정 조건을 만족할 때까지 집단의 분할/병합을 통해 k를 증가/감소시켜 나간다.

Pelleg와 Moore[2]가 제시한 X-means 알고리즘은 일정 범위의 k 값에 대해 K-means 알고리즘을 수행하고 각 결과를 BIC score[6, 7]를 이용하여 평가한다. X-means 알고리즘의 기본적인 문제점은 모든 집단이 동일한 분산 행렬을 가지며, 분산 행렬은 단위행렬의 상수배로 주어진다는 점이다. 즉 모든 집단이 동일한 분포를 가지며 이 분포는 구형 가우시안 분포라고 가정한다. 이러한 가정은 데이터가 구형으로 분포하지 않는 경우 심각한 과대적합을 유발하는 문제점이 있다. 하지만 X-means는 여타의 추가적인 파라미터를 필요로 하지 않는 장점이 있다.

Hamerly와 Elkan[3]은 통계적인 테스트(hypothesis test)를 이용하는 G-means 알고리즘을 제안하였다.

G-means 알고리즘은 Anderson-Darling 테스트(AD test)를 이용하여 국부적인 가설, 즉 집단을 이루는 데이터들이 하나의 가우시안 분포로부터 생성된 점들인지를 검사한다. 이 가설을 만족시키지 못하는 집단들은 K-means 알고리즘을 이용하여 두 개의 집단으로 분할하며, 이 과정은 집단들이 모두 AD 테스트를 만족할 때까지 반복적으로 시행된다. G-means 알고리즘은 X-means 알고리즘과 같은 분산 행렬의 제약이 없으므로 X-means에 비해 과대적합이 적지만, K-means 알고리즘의 hard/crisp assignment는 집단들이 중첩되어 존재하는 경우에 X-means와 마찬가지로 잘못된 집단을 생성할 수 있다. 또한 통계적 테스트의 특성상 집단에 포함되는 데이터 포인트의 개수가 적은 경우 가설을 만족시키지 못함으로 인해 과대적합이 발생하며, 경험적 또는 실험적으로 유의 수준(significance level)을 설정해야 하는 문제점이 있다.

Feng과 Hamerly[4]는 G-means를 개선한 PG-means (Projected Gaussian means) 알고리즘을 제안하였다. PG-means 알고리즘 역시 집단의 분할을 위해 통계적인 테스트를 이용한다. 하지만 G-means가 국부적으로 하나의 집단에 속하는 데이터가 하나의 가우시안 분포에서 생성되었는지를 AD 테스트로 검사하는 반면, PG-means는 전역적으로 모든 데이터가 가우시안 혼합(mixture) 분포에서 생성되었는지를 Kolmogorov-Smirnov 테스트(KS test)를 통해 검사한다. 또한 중첩된 집단 문제를 해결하기 위해 K-means가 아닌 EM 알고리즘을 이용하여 최적의 분할을 찾아낸다. PG-means는 연산의 효율성을 위해 데이터를 1차원으로 무작위 투영(random projection)을 행한 후 이 데이터를 이용하여 KS 테스트를 시행한다. 이러한 무작위 투영 방향에 따라 실제 데이터의 가우시안 혼합 분포가 훼손될 수 있으므로, PG-means는 서로 다른 방향으로 여러 번 투영을 시행하고 각각 KS 테스트를 시행한다. PG-means는 EM 알고리즘을 통해 K-means 알고리즘의 hard assignment 문제를 해결하였지만, 유의 수준과 투영의 횟수 등 여러 파라미터를 필요로 하는 단점이 있다.

Welling과 Kurihara[5]는 기존의 EM 알고리즘의 역할을 바꾼 Maximization-Expectation 알고리즘을 이용하여 집단의 개수를 찾아낸다. ME 알고리즘은 가변 베이시안(variational Bayesian) 방법의 특별한 경우로 PG-means와 비슷하거나 나은 결과를 보이지만 다른 알고리즘에 비해 수행 속도가 느리고 hyper-parameter 설정에 그 결과가 영향을 받는 단점이 있다.

일반적으로 가우시안 형태의 집단을 추정하는 성능은 X-means < G-means < PG-means < BK-means의 순서이다. X-means, G-means는 K-means의 wrapper 알고리즘이고 PG-means는 EM 알고리즘의 wrapper 알고리즘이며 BK-means는 가변 기법(variational method)을 사용한다. 이 논문에서는 이들 알고리즘 중 K-means를 사용하는 G-means와 기존의 알고리즘 중 가장 나은 성능을 보이는 BK-means를 그 비교 대상으로 실험하였다.

Ⅲ. Modified X-means

MX-means 알고리즘은 하나의 집단에서 시작하여 BIC 점수를 최대로 증가시키는 집단을 반복적으로 분할하는 하향식 방법으로, 더 이상의 점수 증가가 없을 때까지 분할은 계속된다. 그림 1은 MX-means의 수행 과정을 나타낸 의사코드(pseudo code)이다. 하나의 집단이 두 개로 분할된 후에 각 집단에 할당되는 데이터들은 K-means 알고리즘을 통해 결정된다. MX-means 알고리즘은 G-means와 마찬가지로 하나의 집단을 두 개로 나눌 것인지를 국부적으로 결정한다. 즉, 이전의 집단들 각각을 두 개의 집단으로 나누며 이 때 다른 집단에 속한 데이터는 분할을 위한 결정에 영향을 미치지 않는다. 그림 1의 line 13에서 K-means 알고리즘은 현재 관심의 대상이 되는 집단을 2개로 나누고 있으며 이는 국부적으로 하나의 집단을 두 개의 집단으로 나눌지 결정하는 것을 의미한다. 반면 line 22에서는 전체 데이터에 대해 K-means 알고리즘을 수행하고 있으며 이는 국부적인 집단 분할 중 BIC 점수를 최대로 하는 분할에 대해 전체적으로 집단들을 갱신하는 과정이다.

두 개의 집단이 중첩되거나 인접하여 나타나는 경우 K-means 알고리즘은 초기값의 설정에 따라 상이한 결과를 보여준다. 따라서 line 11은 동일한 집단을 서로 다른 초기화 값을 통해 여러 번 나누어 보고 그 중 BIC 점수를 최대로 하는 분할을 선택한다. 초기값의 설정은 X-means의 경우 무작위로 집단의 중심을 두 개의 중심으로 나누었지만, 여기서는 PCA를 통해 데이터가 분포되는 방향을 참고하여 중심을 분할하였다. 동일한 집단에 대한 분할 시도 횟수는 데이터의 차원으로 설정되었다.

3.1 BIC 점수

MX-means는 BIC 점수를 최대로 하는 군집화를 찾아내는 방법이다. 데이터 X와 서로 다른 값의 k를 가지는 군집화 모델 M_k가 주어질 경우, 모델들을 비교하는 방법에는 여러 가지가 있다. 이 논문에서 모델의 적합성을 판단하는 기준은 BIC로 X-means에서 사용한 것과 같은 식 (1)로 주어진다.

$$BIC(M) = \hat{l}(X) - \frac{n_p}{2} logN \quad (1)$$

이 때 n_p는 파라미터의 개수를, N은 데이터의 개수를, 그리고 $\hat{l}$은 데이터의 log-likelihood 값을 나타낸다.

```
1  k = 1; // number of clusters
2  p_score = -∞; // previous score
3  c_score = evaluate_BIC(data, k); // current score
4  p_mean = mean(data); // previous mean
5
6  while c_score > p_score
7      p_score = c_score;
8      for cl = 1 to k // cluster index
9       pca = do_pca(data(cl)); // local split direction
10
11          for sd = 1 to data_dimension
    // split direction
12              c_mean = remove(p_mean, cl);
     // remove one cluster
13              c_mean =
       add_mean(c_mean, c_mean ± pca(sd));
       // add two clusters
14              s_mean(cl, sd) =
       K_means(data(cl), c_mean, 2);
       // split
15              s_score(cl, sd) =
       evaluate_BIC(data, k+1, s_mean(cl, sd));
       // evaluate
16          end;
17        end;
18
19      [c_score, c_mean] = max(s_score, s_mean);
20      if c_score > p_score
21        k = k + 1; // increase the number of clusters
22        p_mean = K_means(data, c_means, k + 1);
23      else
24        break;
25      end;
26      end;
27
28  return p_mean;
```

그림 1. MX-means 알고리즘
Fig. 1. MX-means algorithm

log-likelihood는 주어진 데이터 X를 가장 잘 표현하는 모델에서 최대값을 가지게 된다. 하지만 집단의 개수가 증가함에 따라 log-likelihood는 비례해서 증가하게 되며, 최악의 경우 N 개의 집단으로 구성된 모델로 N 개의 포인트를 나타내는 경우 log-likelihood는 0으로 최대값을 가지게 된다. 이처럼 필요 이상의 복잡한 모델을써서 발생하는 과대적합을 방지하기 위해 식 (1)의 두 번 째 항인 벌칙항(penalty term)이 필요하다. 이 항은 모델의 복잡도(complexity)를 나타내는 항으로 복잡한 모델에 대해 벌칙을 증가시키며, 따라서 식 (1)은 likelihood와 모델의 복잡도를 고려하여 최적의 모델을찾아낸다. 식 (1)의 BIC는 Rissanen의 MDL(Minimum Description Length)[8]과 동일한 것으로 알려져 있다.

각 집단의 평균과 분산에 대한 Maximum Likelihood Estimation (MLE) 값은 각각 식 (2), (3)과 같이 구할 수 있다.

$$\hat{\mu}_i = \frac{1}{N_i} \sum_{j \in C_i} x_j \quad (2)$$

$$\hat{\Sigma}_i = \frac{1}{N_i - 1} \sum_{j \in C_i} (x_j - \hat{\mu}_i)(x_j - \hat{\mu}_i)^T \quad (3)$$

이 때 N_i는 i 번째 집단에 속하는 데이터의 개수를 나타내고, C_i는 i 번째 집단에 속하는 데이터의 부분집합을 나타낸다. 평균과 분산에 대한 MLE 값을 이용하여 각 데이터 포인트의 확률은 식 (4)와 같이 구할 수 있다.

$$\hat{p}(x_j) = \frac{N_i}{N} \frac{1}{(2\pi)^{d/2} |\hat{\Sigma}_i|^{1/2}} \exp\left(-\frac{1}{2} (x_j - \hat{\mu}_{(j)})^T \hat{\Sigma}_{(j)} (x_j - \hat{\mu}_{(j)}) \right) \quad (4)$$

이 때 $\mu_{(j)}$와 $\Sigma_{(j)}$는 각기 j 번째 데이터 포인트가 속한 집단의 평균과 분산을 나타낸다. 식 (4)를 이용하여 i 번째 집단에 속하는 데이터의 log-likelihood는 식 (5)와 같이 계산된다.

$$\hat{l}(C_i) = \sum_{j \in C_i} \log \hat{p}(x_j)$$

$$= \sum_{j \in C_i} \left[\log \frac{N_i}{N} - \frac{d}{2} log 2\pi - \frac{1}{2} log|\widehat{\Sigma}_i| - \frac{1}{2} trace (x_j - \mu_{(j)})^T \widehat{\Sigma}_i^{-1} (x_j - \mu_{(j)}) \right]$$

$$= N_i \log N_i - N_i \log N - \frac{dN_i}{2} log 2\pi - \frac{N_i}{2} log|\widehat{\Sigma}_i| - \frac{(N_i - 1)d}{2} \quad (5)$$

이 때 모델의 파라미터 개수는 K-1 개의 가우시안 컴포넌트의 확률, K개의 D 차원 평균 벡터, 그리고 K개의 D × D 분산 행렬에 의해 식 (6)과 같이 정의된다.

$$n_p = (K-1) + D \cdot K + \frac{D(D+1)}{2} \cdot K \quad (6)$$

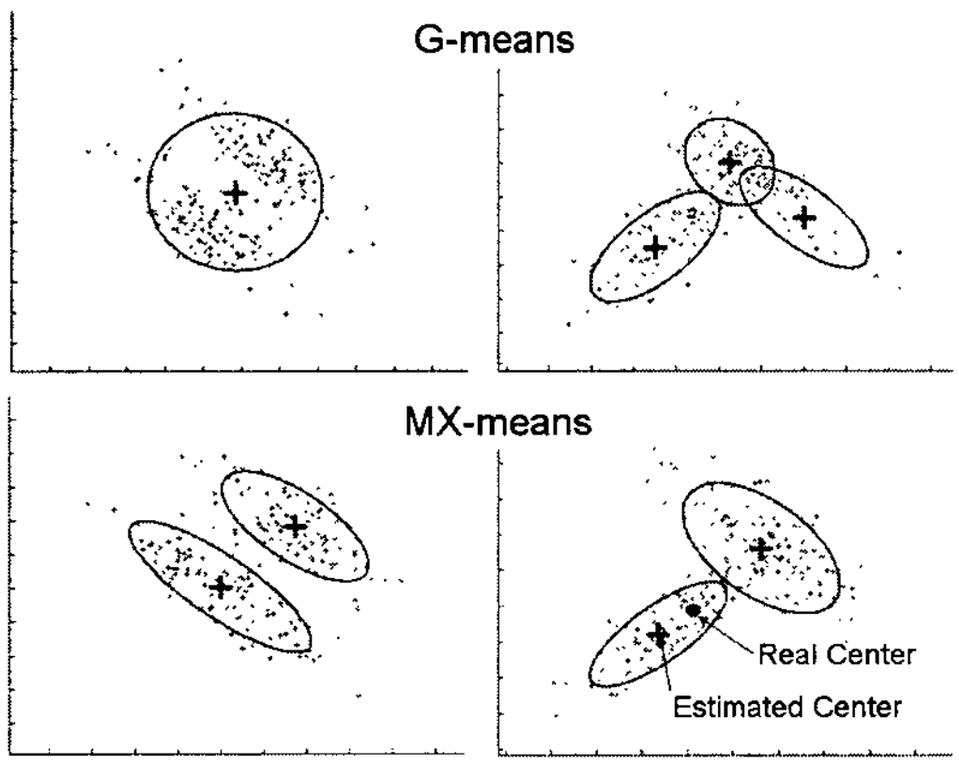

그림 2. G-means와 MX-means의 군집화 결과
Fig. 2. Clustering results of G-means and MX-means

그림 2는 2개의 가우시안 컴포넌트에서 생성된 데이터에 대해 군집화를 한 예를 나타내고 있다. 그림에서 알 수 있듯이 G-means의 경우에는 X-means 보다 과대적합이나 과소적합(under-fitting)이 심하게 발생한다. 특히 통계적 검사의 특성상 데이터 포인트의 수가 적은 경우, 과대적합이 심하게 발생한다. 이에 비해 MX-means는 G-means에 비해 안정적으로 집단을 찾아내는 것을 알 수 있다. 하지만 MX-means의 경우에도 두 집단이 중첩되어 존재하는 경우에 각 집단의 중심이 실제 중심에서 벗어나는 것을 알 수 있다. 이는 MX-means 알고리즘이 K-means 알고리즘을 통해 각 데이터 포인트를 하나의 집단에만 할당하는 hard clustering에 기인한 것으로 G-means의 경우도 마찬가지이다. 이를 해결하기 위해서 다음 장에서는 K-means 알고리즘 대신 EM 알고리즘을 사용하여 하나의 데이터 포인트가 모든 집단에 속할 수 있는 soft clustering을 이용한 GX-means 알고리즘을 제안한다.

Ⅳ. Generalized X-means

Generalized X-means(GX-means)는 기본적으로 그림 1의 MX-means와 동일한 알고리즘 구조를 가지며, 식 (1)과 동일한 BIC 점수를 사용한다. 이 때 log-likelihood $\hat{l}$는 EM 알고리즘의 출력값을 사용하며 파라미터의 개수는 식 (6)과 동일하다.

GX-means가 MX-means와 가장 크게 다른 점으로는 주어진 집단의 개수 K에서 최적의 군집화를 K-means가 아닌 EM 알고리즘을사용한다는 점이다. 그림 2에서 보았듯이 K-means 알고리즘의 hard assignment는 중첩된 집단의 경우 잘못된 집단의 중심을 찾아내는 문제점이 있다. 하지만 EM 알고리즘의 soft assignment는 이러한 문제점이 없다. 그림 3은 3개의 실제 집단에서 생성된 데이터를 군집화 한 예로 GX-means가 MX-means보다 실제 중심을 보다 정확하게 찾아내고 있음을 알 수 있다.

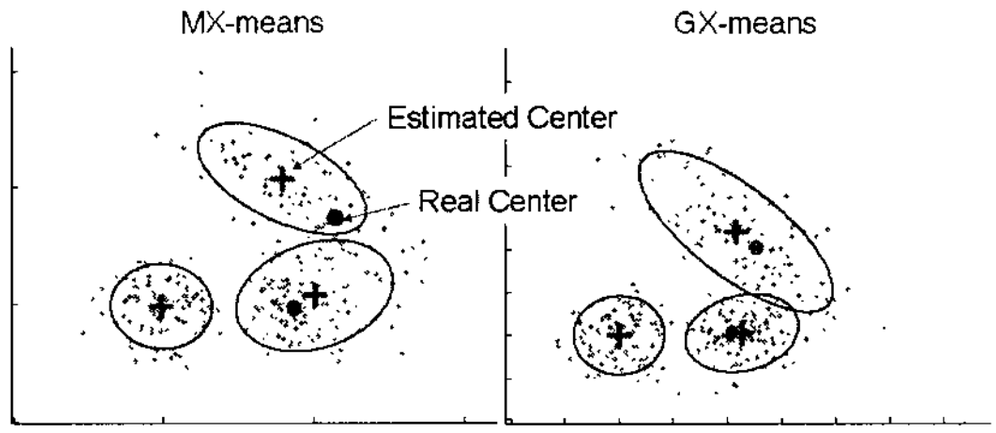

그림 3. MX-means와 GX-means의 군집화 결과
Fig. 3. Clustering results of MX-means and GX-means

또 한 가지 다른 점은 MX-means의 경우 집단의 분할을 국부적으로(locally), 즉 분할하고자 하는 집단에 속하는 데이터 포인트들만을 대상으로 해서 결정하는 반면, GX-means는 전역적으로(globally) 결정한다는 점이다. GX-means에서는 하나의 포인트가 모든 집단에 속할 수 있으므로, 특히 중첩된 집단들을 분리하는 경우에는 국

부적인 연산만으로는 정확한 집단들을 찾아낼 수가 없으므로 전역적인 계산을 필요로 한다.

또 한 가지 MX-means와 다른 점은 중심을 분할하는 경우 새롭게 생성되는 집단의 파라미터를 설정하는 방법이다. MX-means에서는 K-means 알고리즘을 사용하므로 중심 좌표만을 설정하면 되고 이는 그림 1의 line 13과 같이 PCA를 통해 결정하였다. 하지만 EM을 이용하기 위해서는 중심과 더불어 분산 및 각 집단의 사전 확률을 할당해줄 필요가 있다. 이 논문에서는, 하나의 집단을 두 개로 분할한 경우 분산 행렬은 분할하기 전 집단의 분산 행렬로 동일하게 지정하였고, 집단의 사전 확률은 분할하기 이전 집단 값의 1/2로 두 집단에 동일하게 설정하였다.

GX-means 알고리즘이 MX-means 알고리즘보다 그 성능에서 우수하며, 알려진 하향식 알고리즘 중에서도 가장 나은 성능을 보이지만, 실행 속도에 문제가 있다. GX-means가 사용하는 EM 알고리즘은 그 실행 시간이 K-means에 비해 오래 걸릴 뿐만이 아니라, MX-means는 국부적으로 분할을 판단하는 반면 GX-means는 전역적으로 판단하기 때문에 많은 연산을 필요로 한다. GX-means 알고리즘은 알려진 하향식 알고리즘 중 가장 많은 연산을 필요로 하며, BK-means에 비해서도 3-4배 느리다. 따라서 이 논문의 실험을 위해서 다른 알고리즘들은 Matlab으로 구현하였지만, GX-means 알고리즘의 EM 알고리즘 부분은 C로 구현하여 실험하였다.

GX-means의 실행 속도 개선을 위한 방안으로는, MX-means에서와 같이 집단의 분할을 국부적으로 판단하고 분할이 결정된 이후에 전역적으로 파라미터를 갱신하는 방법이 있을 수 있다. 이 때 고려되어야 할 점은 soft assignment 상황에서 특정 집단에 속하는 데이터의 부분집합을 어떻게 결정할 것인가 하는 점이다. EM 알고리즘의 결과로 군집화를 하기 위해서는, 각 데이터 포인트가 주어졌을 때 그 데이터 포인트를 생성할 확률이 가장 높은 집단에 각 데이터 포인트를 할당하는 것이 일반적이지만, 여기에 특이점(outlier)들을 고려하여 일정 확률 이상으로 임계값을 정해 데이터의 포인트들을 군집화하는 방안을 연구 중에 있다. 이처럼 분할을 결정하기 위해서 지역적으로 판단을 내리는 경우에는 전역적으로 판단을 내리는 경우에 비해 정확성이 떨어질 수 있지만 실행 속도 향상에 도움이 되며, 근사값을 구한다는 점에서 임계값 설정을 통해 대상이 되는 데이터 포인트를 더욱 제한하는 것이 가능해진다. MX-means의 hard assignment는 이러한 부분 집합 결정 문제가 발생하지 않는다.

또 한 가지는 중심의 분할 방향을 결정하는 문제이다. 현재 중심의 분할 시도 횟수는 데이터의 차원과 동일하게 설정되어 있지만, 고차원 데이터의 경우 실험적으로 min(4, dimension) 값에서도 거의 동일한 결과를 보임을 알 수 있었으며, 이에 대한 연구 역시 진행 중이다.

V. 실험 결과

이 논문에서 제안한 두 알고리즘, MX-means와 GX-means의 성능을 기존의 G-means와 BK-means 알고리즘과 비교하기 위해 일련의 실험 데이터에 대한 실험을 수행하였다.

그림 4는 평행하게 존재하는 두 개의 집단에 대한 군집화를 나타낸 것으로, 무작위로 생성된 100개의 데이터 집합에 대하여 각 알고리즘이 찾아낸 집단 개수를 그림 5에 나타내었다.

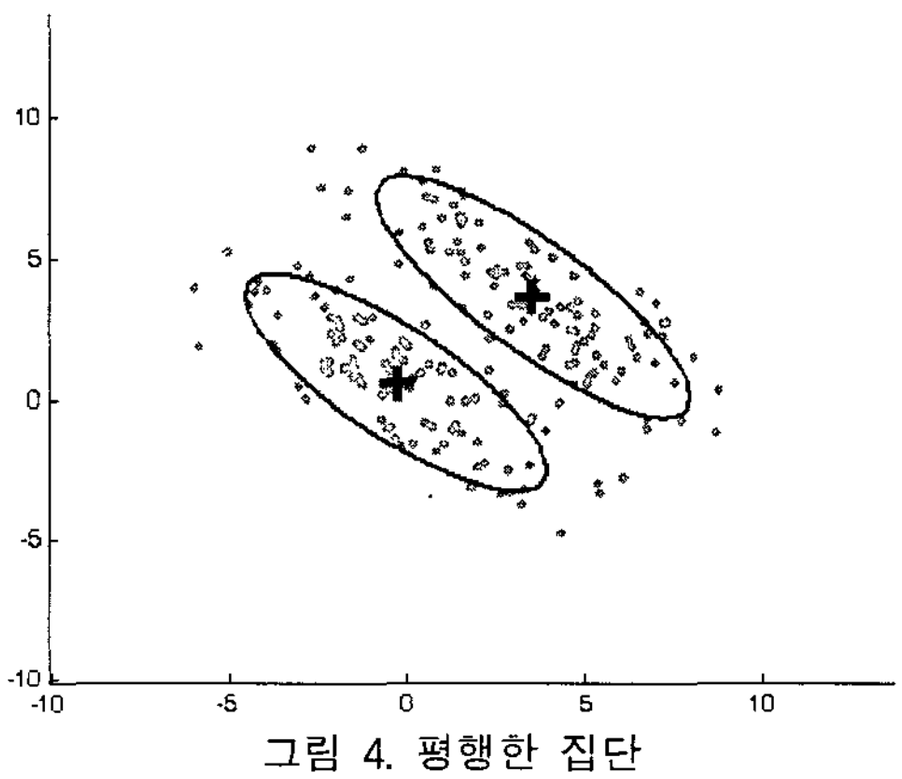

그림 4. 평행한 집단
Fig. 4. Parallel clusters

그림 5에서 x 축은 두 집단 사이의 거리를 나타낸 것으로 두 집단이 가깝게 존재할 경우 하나의 집단만을 찾아낼 확률이 높아진다. 그림 5의 수직 막대는 찾아낸 집단 개수의 분산을 나타낸다. 그림 5에서 알 수 있듯이 제안한 두 알고리즘은 기존의 G-means와 BK-means에 비해 보다 안정적으로 두 개의 집단을 찾아내고 있으며 이는 두 집단이 가깝게 존재할 경우 확연하다. 표 1은 그림

5의 실험 결과를 요약한 것으로 그림 4와 같이 평행한 집단의 경우에는 MX-means가 가장 우수함을 알 수 있다.

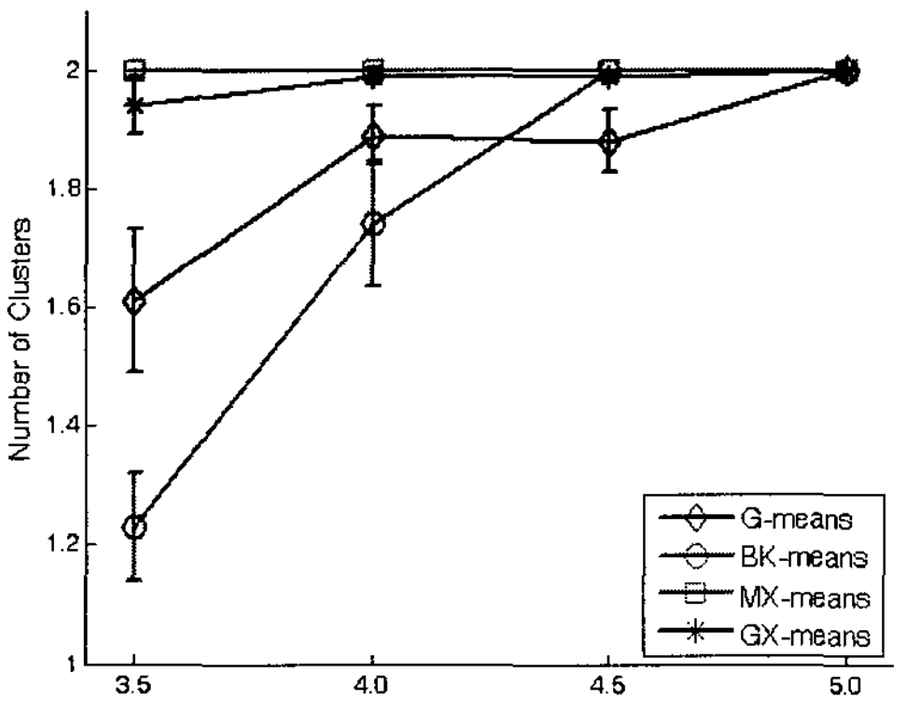

그림 5. 군집화 결과
Fig. 5. Clustering results

표 1.평행한 두 군집에 대한 군집화 결과
Table 1. Clustering results of parallel clusters

기법	3.5		4.0		4.5		5.0	
	평균	분산	평균	분산	평균	분산	평균	분산
G	1.61	0.24	1.89	0.10	1.88	0.11	2.00	0.00
BK	1.23	0.18	1.74	0.21	2.00	0.00	2.00	0.00
MX	2.00	0.00	2.00	0.00	2.00	0.00	2.00	0.00
GX	1.94	0.10	1.99	0.03	1.99	0.01	2.00	0.00

그림 6은 경계가 인접한 3개의 집단에 대한 군집화를 나타낸 것으로, 그림 7에 그 결과를 나타내었다. 이 때 그래프의 중심값은 100개의 데이터 집합에 대해 찾아진 집단 개수의 평균값을 나타내고 수직 막대는 분산을 나타낸다. 그림 7에서 알 수 있듯이, 인접한 집단들에 대한 군집화의 경우 G-means 알고리즘은 평균 4개 이상의 집단을 찾아내어 과대적합이 심하게 발생함을 알 수 있다. MX-means 알고리즘은 G-means 정도는 아니지만 3개 이상의 집단을 찾아내는 경우가 있지만, GX-means는 안정적으로 3개의 집단을 찾아내고 있으며 이는 BK-means도 마찬가지다.

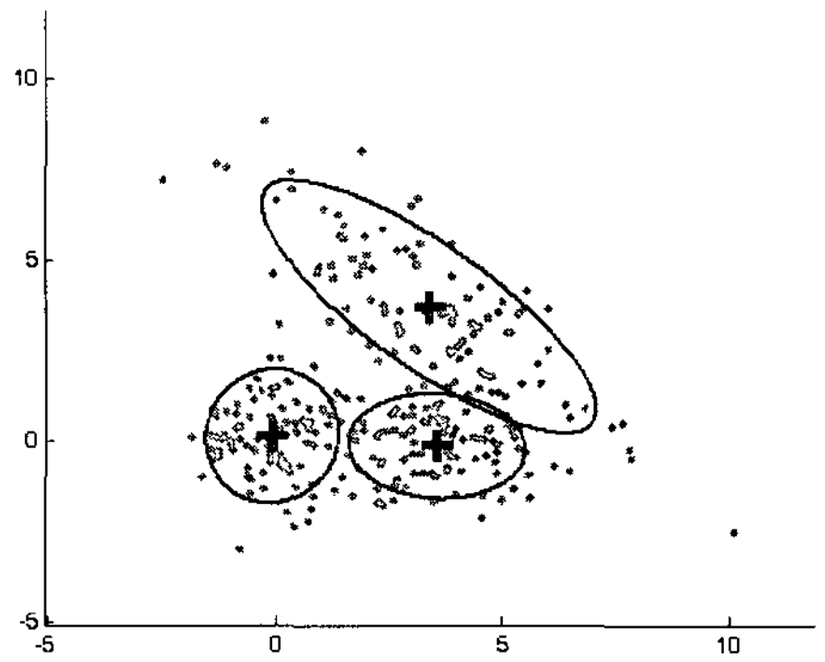

그림 6. 인접한 집단
Fig. 6. Touching clusters

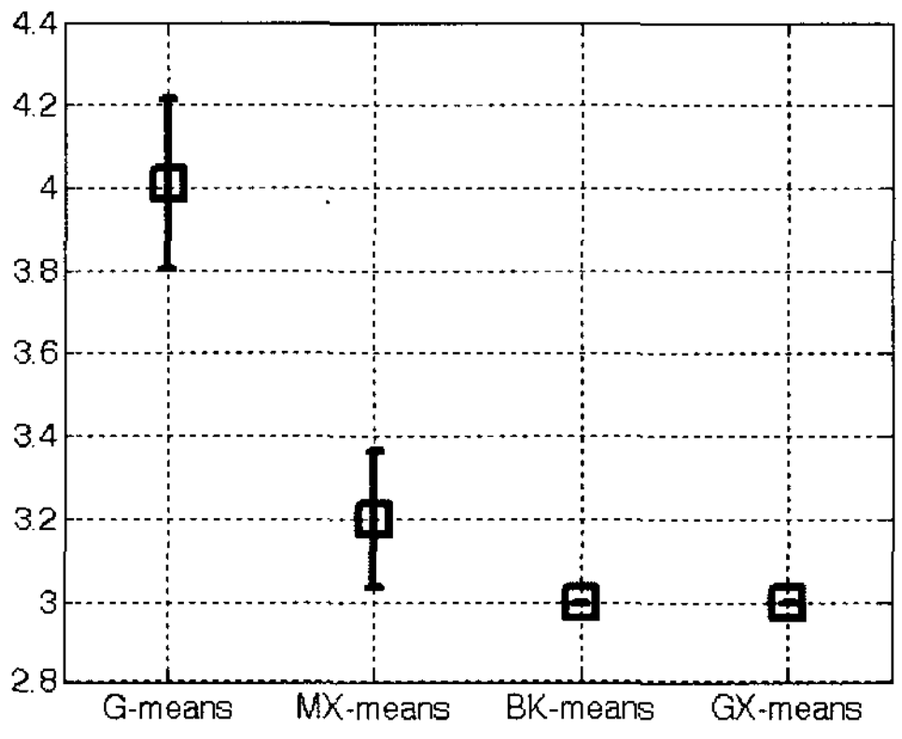

그림 7. 군집화 결과
Fig. 7. Clustering results

표 2는 그림 7에 대한 결과를 요약한 것으로, 이와 더불어 BK-means와 GX-means가 찾아낸 중심과 실제 집단 중심 사이의 오차를 비교하고 있다. 표 2에서 알 수 있듯이 GX-means는 BK-means에 비해 중심 위치의 오차와 분산 면에서 모두 우수함을 알 수 있다.

표 2. 인접한 세 집단에 대한 군집화 결과
Table 2. Clustering results of touching clusters

기법	집단 개수		중심위치의 오차					
			중심 1		중심 2		중심 3	
	평균	분산	평균	분산	평균	분산	평균	분산
G	4.01	0.41	-	-	-	-	-	-
X	3.20	0.32	-	-	-	-	-	-
BK	3.00	0.00	0.13	0.01	0.21	0.03	0.39	0.13
GX	3.00	0.00	0.14	0.00	0.16	0.01	0.28	0.03

그림 8은 일부 겹쳐진 두 집단에 대한 군집화를 나타낸 것으로, 그림 9에 그 결과를 나타내었다. 이 때 중심값은 찾아낸 평균 집단의 개수를, 수직 막대는 분산값을 나타낸다. 그림 7에서와 마찬가지로 G-means 알고리즘은 다른 알고리즘에 비해 과대적합이 심하게 발생함을 알 수 있다. 다른 3개의 알고리즘들은 안정적으로 2개의 집단을 찾아내고 있다.

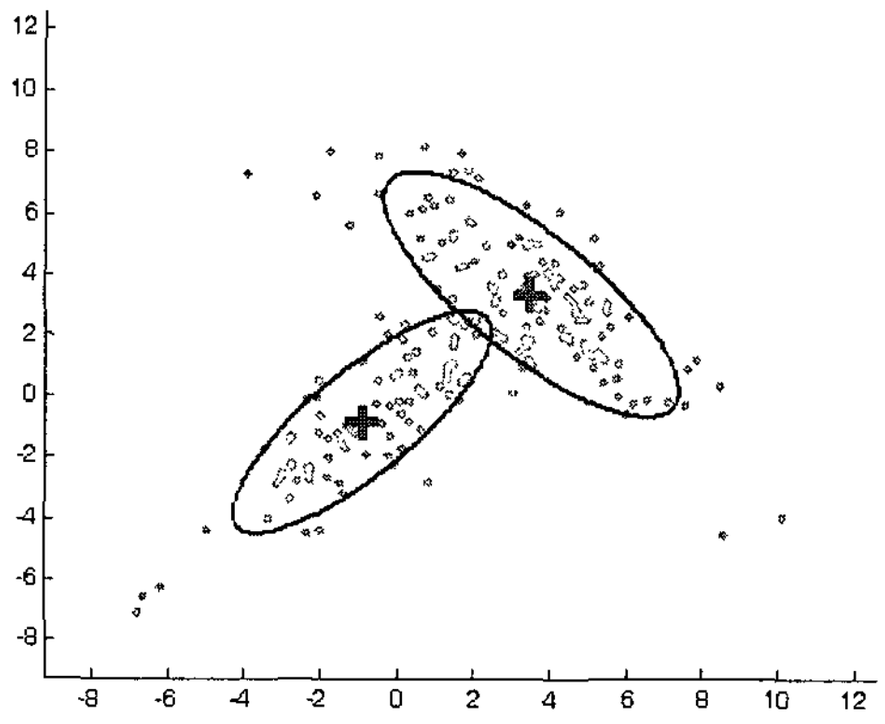

그림 8. 겹쳐진 집단
Fig. 8. Overlapping clusters

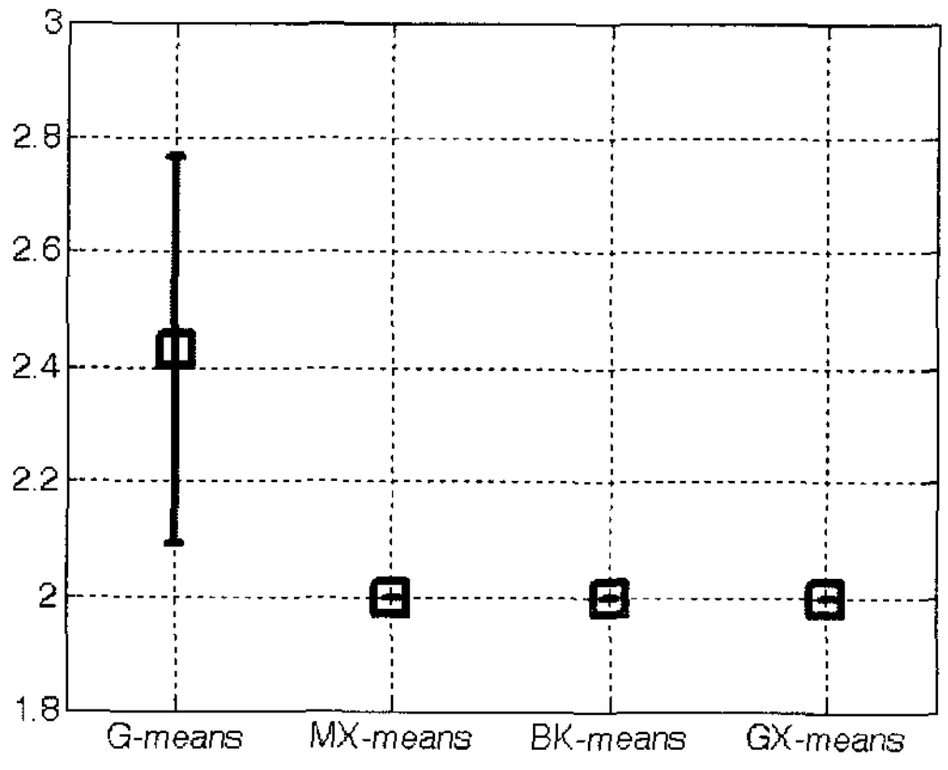

그림 9. 군집화 결과
Fig. 9. Clustering results

표 3은 그림 9를 요약한 것으로, 2개의 집단을 안정적으로 찾아내는 알고리즘들에 대한 중심 위치들을 비교하고 있다. 표 3에서 알 수 있듯이 GX-means는 MX-means나 BK-means에 비해 안정적인 결과를 나타내고 있으며 MX-means는 K-means의 hard clustering으로 인해 다른 두 알고리즘에 비해 오차가 크게 발생함을 알 수 있다.

표 3. 겹쳐진 두 집단에 대한 군집화 결과
Table 3. Clustering results of overlapping clusters

기법	집단 개수		중심위치의 오차			
			중심 1		중심 2	
	평균	분산	평균	분산	평균	분산
G	2.43	0.67	-	-	-	-
MX	2.00	0.00	1.08	0.12	0.45	0.03
BK	2.00	0.00	0.57	0.11	0.27	0.02
GX	2.00	0.00	0.35	0.05	0.27	0.03

VI. 결 론

이 논문에서는 BIC 점수를 이용하여 군집의 개수를 효율적으로 결정하는 X-means 알고리즘을 개선한 두 가지 알고리즘, MX-means와 GX-means 알고리즘을 제안하였다. 제안한 알고리즘들은 하나의 집단에서 시작하여 BIC 점수를 증가시키는 집단들을 반복적으로 분할해 나가는 하향식 wrapper 알고리즘으로, 주어진 집단의 수 k에서 최적의 집단들을 찾아내기 위해 MX-means는 K-means 알고리즘을 사용하여 hard clustering을 수행하고 GX-means는 EM 알고리즘을 통해 soft clustering을 수행하는 점에서 다르다. 실험 결과에 나타나 있듯이 MX-means 알고리즘은 K-means의 wrapper 알고리즘들 중 가장 우수한 성능을 보였으며, GX-means는 비교 대상이 되는 모든 알고리즘 중에서 가장 우수한 성능을 보였다. 또한 제안한 알고리즘들은 여타의 파라미터를 필요로 하지 않는 장점이 있다. 하지만 GX-means는 EM 알고리즘을 반복적으로 수행하는 구조로 인해 실행 속도가 느린 단점이 있으며, 비교 대상 알고리즘들 중에서 가장 느리다. C로 구현한 GX-means 알고리즘은 충분히 큰 데이터 집합이나 고차원의 데이터를 처리하는 데 문제가 없지만, 다른 알고리즘과 동일하게 Matlab 만으로 구현한 경우에는 속도에 문제가 있다. 실험 결과에서 알 수 있듯이, 집단들이 중첩되지 않게 분포하는 경우에는 MX-means 알고리즘을 사용하면 빠른 속도로 정확한 집단 구성이 가능하며, 집단들이 중첩되어 분포하는 경우에는 GX-means 알고리즘을 통해 정확한 집단 구성이 가능하다.

GX-means의 가장 큰 문제점은 그 실행 속도에 있으며 이를 개선하기 위해 연구가 진행 중에 있다. 실행 속도 개선을 위해서는 먼저 MX-means에서의 국부적인 분할 결정을 도입하는 방안이 있을 수 있다. GX-means는 하나의 집단을 분할하기 위해서 전역적으로 분할을 평가하고 있다. 하지만 MX-means의 실험 결과에서 알 수 있듯이 국부적인 결정으로도 일정 수준 이상의 정확도는 얻어낼 수 있으므로 국부적인 분할 결정을 통해 상당 부분 연산량을 줄일 수 있을 것으로 기대된다. 또한 중심을 분할하는 경우 분할을 시도하는 회수를 제한하는 방법도 가능하다. 현재 하나의 집단을 데이터의 차원만큼 분할해 보고 그 중 최고의 점수를 가지는 분할을 선택하지만, 고차원 데이터의 경우 실험적으로 min(4, dimension) 값으로 횟수를 제한하는 경우에도 비슷한 결과를 얻을 수 있었다. 이는 Dasgupta[9]의 무작위 투영 결과와 어느 정도 일치하지만 PCA를 이용하는 경우에는 더 낮은 수준으로 제한하는 방법이 가능할 것으로 생각되며 이에 대한 연구 역시 진행 중이다.

참고문헌

[1] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006

[2] Dan Pelleg and Andrew Moore, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," *Proceedings of the 17th International Conference on Machine Learning*, pp. 727-734, 2000

[3] Greg Hamerly and Charles Elkan, "Learning the k in k-means," *Proceedings of the 17th Annual Conference on Neural Information Processing Systems(NIPS-2003)*, pp. 281-288, 2003

[4] Yu Feng and Greg Hamerly, "PG-means: learning the number of clusters in data," *Proceedings of the 20th Annual Conference on Neural Information Processing Systems(NIPS-2006)*, pp. 393-400, 2006

[5] Max Welling and Kenichi Kurihara, "Bayesian k-means as a 'maximization- expectation' algorithm," *Proceedings of the 6th SIAM Conference on Data Mining*, pp. 472-476, 2006

[6] Robert E. Kass and Larry Wasserman, "A Reference Bayesian Test for Nested Hypotheses and Its Relationship to the Schwarz Criterion," *Journal of the American Statistical Association*, Vol.90, No.431, pp. 928-934, 1995

[7] Gideon Schwarz, "Estimating the Dimension of a Model," *The Annals of Statistics*, Vol.6, No.2, pp. 461-464, 1978

[8] J. Rissanen, "Modeling by shortest data description," *Automatica*, Vol.14, No.5, pp. 454-471, 1978

[9] Sanjoy Dasgupta, "Experiments with Random Projection," *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, pp. 143-151, 2000

저자소개

허 경 용(Gyeongyong Heo)

1994년 2월 연세대학교 전자공학과 (공학사)
1996년 8월 연세대학교 본대학원 전자공학과(공학석사)
2004년 9월 ~ 현재 Dept. of Computer and Information Science and Engineering, University of Florida
※관심분야 : 영상처리, Machine Learning, Bayesian Network

우 영 운(Young Woon Woo)

1989년 2월 연세대학교 전자공학과 (공학사)
1991년 8월 연세대학교 본대학원 전자공학과(공학석사)
1997년 8월 연세대학교 본대학원 전자공학과(공학박사)
1997년 9월 ~ 현재 동의대학교 멀티미디어공학과 교수
2007년 ~ 현재 한국해양정보통신학회 국제이사
※관심분야 : 인공지능, 영상처리, 의료정보