Journal of Korean
Data & Information Science Society
2003, Vol. 14, No.4 pp. 725~736

A Comparison of Clustering Algorithm in Data Mining

Yung-Seop Lee[1)] · Mi-Young An[2)]

Abstract

To provide the information needed to make a decision, it is important to know the relationship or pattern between variables in database. Grouping objects which have similar characteristics of pattern is called as cluster analysis, one of data mining techniques. In this study, it is compared with several partitioning clustering algorithms, based on the statistical distance or total variance in each cluster.

1. 머리말

컴퓨터와 인터넷이 급격히 발전함에 따라 기업이나 모든 조직들은 데이터를 정보의 인프라로 인식하고 데이터베이스를 구축하게 되었다. 이런 데이터베이스는 간단한 알고리즘을 가진 도구로 찾아내기 어려울 정도로 커졌다. 이러한 방대한 데이터베이스의 특징을 파악함으로써 이제까지 알 수 없었던 새로운 지식을 얻고자 하는 찾아내는 전반적인 과정을 KDD(Knowledge Discovery in Database)과정이라고 하는데, 이중 한 단계가 데이터 마이닝이다. 데이터 마이닝의 여러 분석방법 중 데이터들의 변수들을 비슷한 특징을 가지는 소그룹으로 나누어 그 그룹들의 특징이나 대표성을 찾는 과정을 군집분석이라 한다.

군집 분석은 크게 분리군집방법(partitioning clustering algorithm)과 계층적군집방법(hierarchical clustering algorithm)방법으로 구분할 수 있는데 본 논문에서는 분리군집 방법의 알고리즘들에 대해 다루겠다. 일반적으로 분리군집방법에서 많이 알려져 있는 알고리즘에는 k-평균 알고리즘(k-means algorithm)이 있다. 하지만 k-평균 알고리즘은 중심 계산을 평균으로 하기 때문에 이상치에 민감한 단점이 있다. 이러한 단점을 보완할 수 있는 알고리즘으로 PAM(Partitioning Around Medoids), CLARA(Clustering LARge Applications), CLARANS(Clustering Large Applications based on RANdomized Search)가 있다(Kaufman and Rousseeuw 1978, Kaufman and Rousseeuw 1986, Ng and Han 1994). 이와 같은 방법들은 실제 관찰치인 메도이

1) 서울시 중구 필동 3가 26번지 동국대학교 통계학과 조교수
E-mail : yung@dongguk.edu
2) 서울시 용산구 한강로 2가 삼성경제연구소

드(medoid)를 중심점으로 사용하므로 k-평균 알고리즘 보다 이상치에 덜 민감함을 알 수 있다(Kaufman and Rousseeuw, 1990).

기존의 논문들에서는 위에 언급한 여러 알고리즘들에 대하여 복잡도(complexity)를 계산하고, 대용량 데이터들에 적용시켰을 때의 수행 속도로 비교해 놓았다(Ng and Han, 1994). 그러나 통계학적으로 데이터들이 얼마나 효율적으로 분리되었는가에 대한 연구는 없었다. 따라서 본 논문에서는 여러 가지 군집분석 알고리즘들을 통계학적인 관점에서 비교 분석하여 알고리즘들의 특성이나 패턴을 알아보고자 하는 것이 연구의 목적이다.

2장의 본론 1절에서는 일반적으로 사용하는 k-평균 알고리즘에 대해 설명하고, 2절에서는 메도이드 방식의 PAM 알고리즘과 CLARA 알고리즘 그리고 CLARANS 알고리즘에 대해 설명한다. 3절에서는 앞에서 소개한 알고리즘들을 실제 데이터에 적용시켜 그 결과에 대해 비교하고 평가해보기로 한다.

2. 본 론

2.1 k-평균 알고리즘

k-평균 알고리즘은 Cox(1957)와 Fisher(1958)에 의해 제안되었고, Hartigan(1975)과 MacQueen(1967)에 의해 개발된 후 계속적으로 연구되고 있다. 현재 분리군집방법 중에서 가장 보편적으로 많이 쓰이는 알고리즘의 하나로, 군집 내 유사성은 작게 하고, 군집 간 유사성은 크게 분류를 하는 것이 목적이다. 군집의 유사성은 군집의 중심점인 평균과 객체들간의 거리로 측정한다.

k-평균 알고리즘은 중심점과 주어진 객체의 거리를 계산하여 가장 가까운 중심점에 주어진 객체를 할당하는 방법이다. 거리를 계산하는 방법으로는 유클리디안 거리(Euclidean distance), 맨하탄 거리(Manhattan distance), 민코우스키 거리(Minkowski distance), 마할라노비스 거리(Mahalanobis distance)등이 사용된다. 본 연구에서는 관찰치 간의 거리를 측정할 때 앞에서 언급한 여러 가지 거리 측정 방법 중에서 가장 일반적인 유클리디안 거리를 사용하였다. 한편, k-평균 알고리즘을 사용하기 위해서는 군집의 수인 k 가 미리 결정되어 있어야 한다. 군집 수 k는 사용자에 의해 결정되거나 프로그램에 의하여 선택될 수 있다. k-평균 알고리즘은 〈표 1〉 의 4가지 단계를 거친다.

<표 1> k-평균 알고리즘

【단계1】 자료를 k개의 초기 군집으로 나눈다.
【단계2】 k개로 나누어진 군집의 중심을 평균을 이용하여 구한다.
【단계3】 각 객체와 중심들 사이의 거리를 거리 구하는 식을 이용하여 계산한다. 그리고 객체가 현재 속해있는 군집 중심에 가까우면 현재 군집에 포함되고, 다른 군집의 중심에 가까우면 그 군집으로 재분류한다.
【단계4】 새롭게 할당되는 객체가 없을 때까지 【단계2】와 【단계3】을 반복한다.

k-평균 알고리즘은 중심점을 평균으로 계산하기 때문에, 평균을 구할 수 있는 데이터에서만 사용할 수 있다. 예를 들어, 범주형 자료 같은 경우에는 k-평균 알고리즘을 적용시킬 수 없다.

2.2. 메도이드에 의한 알고리즘

2.2.1 PAM 알고리즘

PAM 알고리즘은 k 메도이드의 한 방법의 한 형태로써 Kaufman and Rousseeuw에 의해 1978년에 제안된 알고리즘이다. k-평균 알고리즘에서는 군집의 중심으로 객체들 간의 평균을 계산한 가상의 점을 사용하는 것과 달리, 군집의 중심으로 실제 객체인 메도이드를 중심으로 사용한다. 여기서는 메도이드란 군집 내에서 객체들간의 평균 비유사성이 가장 작은 객체를 말한다.

이상적인 메도이드를 찾기 위해 반복을 통하여 메도이드들을 변화시켜 나가는데, 이렇게 메도이드들을 변화시킬 때마다 객체들이 가까운 메도이드들을 중심으로 객체를 형성하기 위해 움직이게 된다. 이때 객체들이 변화된 메도이드로 인하여 재분류되었을 때 이동한 객체와 본래의 메도이드, 변화된 메도이드와 거리의 차를 비용이라 한다. 비용함수는 메도이드와 나머지 객체들간의 차이 값으로 계산된다. PAM 알고리즘에서는 객체들이 이동하면서 발생한 비용을 모두 더한 총 비용인 식(2-5)를 이용하여 이상적인 메도이드를 찾아나간다.

$$TC_{ih} = \sum_{j} C_{jih} \tag{2-5}$$

k-평균 알고리즘에서는 거리의 차를 사용하지만, PAM 알고리즘에서는 이것을 비용 함수로 대신한다. PAM 알고리즘의 기본적인 절차는 〈표 2〉와 같고 이 알고리즘도 k-평균 알고리즘처럼 k의 개수를 미리 알고 있어야 한다. PAM 알고리즘은 객체들과 군집의 메도이드들 간의 평균 비유사성(average dissimilarity)으로 군집의 효율성을 측정한다.

PAM 알고리즘은 군집의 질을 평가할 때, 모든 경우의 데이터에 대해 계산을 하므로, 데이터의 크기가 커질수록 계산량이 많아 컴퓨터의 수행 속도가 느려지는 단점이 있다 (Han and Kamber, 2000).

<표 2> PAM 알고리즘

【단계1】 임의로 K개의 초기치를 선택한다.
【단계2】 K개의 객체를 데이터에서 임의로 추출하여, 현재 메도이드인 O_i로 설정한다.
【단계3】 현재 메도이드인 O_i에 가까이 있는 객체들로 군집을 분류한 다.
【단계4】 분류된 K개의 군집 내에서 비유사성을 계산하여 가장 좋은 메도이드 O_h를 찾는다.
【단계5】 현재 O_i와 새로운 메도이드 O_h 사이의 $\min_{O_i, O_h} TC_{ih}$를 가지는 메도이드를 찾는다. 만약 $\min_{O_i, O_h} TC_{ih}$가 음수이면, 현재 메도이드 O_i를 새로운 메도이드 O_h로 바꾸고, 【단계3】으로 되돌아 간다.
【단계6】 음수가 아닐 경우가 발생할 때까지 【단계3】, 【단계4】, 【단계5】를 반복하여 평균 비유사성이 가장 낮은 군집을 찾는다.

2.2.2 CLARA 알고리즘

CLARA 알고리즘은 Kaufman and Rousseeuw에 의해 1986년 제안된 알고리즘으로 대용량 데이터를 효율적으로 다루기 위한 방법이다. 군집을 나눌 때, 전체 데이터 대신에 데이터의 표본을 랜덤 추출한다. 추출한 랜덤 표본에 PAM 알고리즘을 적용시켜 표본에서 k개의 최적의 메도이드를 구한다. 표본 추출과 메도이드를 찾는 과정을 반복하여 최적의 메도이드를 찾는다. 표본을 충분히 랜덤추출 하였다면, 그 표본의 메도이드들은 전체 데이터에서 구한 메도이드와 비슷하게 근사할 것이다. CLARA 알고리즘은 데이터에서 반복적으로 표본을 추출하고, PAM 알고리즘을 적용시켜 메도이드를 찾는 과정을 반복하여 최적 군집을 찾는 것이다. 하지만 군집의 정확성을 측정할 경우에는 표본에서 구하여진 메도이드들과 표본의 객체들간의 평균 비유사성을 구하는 것이 아니라, 표본에서 구하여진 메도이드들과 전체 데이터의 모든 객체들의 평균 비유사성을 계산한다. CLARA 알고리즘의 기본적인 절차는 〈표 3〉과 같고 이 알고리즘도 k-평균 알고리즘처럼 군집수를 미리 알고 있어야 한다.

<표 3> CLARA 알고리즘

【단계1】 전체 데이터에서 임의로 표본을 추출하여 그 표본에 PAM 알고리즘을 적용시켜 k개의 메도이드를 찾는다.
【단계2】 단계 1에서 구한 k개의 메도이드를 중심으로 전체 데이터를 이용해 메도이드에 가까운 객체들로 군집을 형성한다.
【단계3】 전체 데이터로 형성된 군집을 이용해 평균 비유사성을 계산한다. 만약 계산된 값이 현재 값보다 작다면, 계산된 값을 현재 값으로 바꾼다.
【단계4】 메도이드가 수렴할 때까지 【단계1】, 【단계2】, 【단계3】을 반복한다.

데이터의 크기가 커지게 되면 PAM 알고리즘과 비교해 볼 때 CLARA 알고리즘이

더 효과적으로 군집을 형성할 수 있는 알고리즘임을 알 수 있다. CLARA 알고리즘은 표본 크기에 의존된다. PAM 알고리즘은 전체 데이터에서 메도이드를 추출하지만, CLARA는 전체 데이터에서 추출된 표본에서 최적의 메도이드를 찾는 것이다. 만약 추출된 표본에 좋은 메도이드가 없다면, CLARA는 최적의 군집을 찾지 못할 수도 있다.

2.2.3 CLARANS 알고리즘

CLARANS 알고리즘은 Ng and Han에 의해 1994년 제안된 알고리즘으로 CLARANS 알고리즘은 그래프의 개념을 이용한다.

그래프 $G_{n,k}$는 n개의 객체, k개의 군집을 가지는 데이터에서 각각의 노드들의 집합을 말한다. 노드란 메도이드들의 집합인 $\{O_{m_1},\cdots,O_{m_k}\}$을 말한다. 만약 두 개의 노드에서 한 개의 메도이드만 다르고 다른 메도이드들은 동일하다면 이 두 노드를 이웃(neighbor)이라고 한다. 즉 노드 $S_1=\{O_{m_1},\cdots,O_{m_k}\}$과 노드 $S_2=\{O_{w_1},\cdots,O_{w_k}\}$는 $|S_1\bigcap S_2|=k-1$로 두 개의 노드에 $k-1$개의 공통 메도이드가 있는 것이다. 각각의 노드들은 $k(n-k)$개의 이웃들을 가진다.

현재 메도이드 O_i가 새로운 메도이드 O_h로 움직일 경우, 이때 발생하는 비용은 PAM 알고리즘에서 정의한 식(2-5)의 TC_{ih}를 이용하여 계산을 한다. 메도이드를 변형시킬 때, 메도이드 O_i는 S_1에 속하는 객체이고, 새로운 메도이드 O_h는 S_2에 속하는 객체이다. $O_i, O_h \notin S_1\bigcap S_2$로 메도이드 O_i와 O_h는 노드의 교집합에 속하지는 않지만, $O_i\in S_1$, $O_h\in S_2$으로 노드에 속하는 유일하게 다른 하나의 메도이드로 새로운 메도이드로 설정한다. CLARANS 알고리즘의 단계는 〈표 4〉와 같다.

< 표 4 > CLARANS 알고리즘

【단계1】 CLARANS 알고리즘에 이용할 총 반복 회수와 평가할 이웃들의 수를 설정한다.
【단계2】 반복 회수인 i 의 값을 1로 초기화한다.
【단계3】 그래프 $G_{n,k}$에서 현재 사용할 노드를 뽑아 초기 노드로 사용한다.
【단계4】 현재 노드에서 평가할 이웃들이 값의 개수만큼 이웃의 표본을 뽑아 이웃들 사이의 비용을 계산한다. 비용 계산식은 식(2-5)를 사용한다.
【단계5】 만약 표본으로 사용한 이웃들간의 비용이 더 작다면 현재 비용을 작은 비용으로 갱신하고 【단계 3】으로 돌아가 이웃들 사이의 비용을 다시 계산한다. 이 과정을 표본의 이웃들의 수만큼 반복한다.
【단계6】 반복한 후 최소 비용을 현재 최소 비용으로 저장하고, 수렴할 때까지 【단계 3】과 【단계 4】를 반복한다.
【단계7】 반복 회수가 총 반복 회수가 될 때까지 전체의 과정을 반복한다.

CLARA 알고리즘처럼 CLARANS 알고리즘은 노드의 모든 이웃들을 평가하지는 않

는다. CLARANS 알고리즘은 노드의 이웃의 표본을 추출하여 평가하는 것이다. 즉 CLARA 알고리즘은 반복을 할때마다 정해진 표본의 개수만큼 표본을 추출하는 것이고, CLARANS 알고리즘은 단계를 거칠 때 마다 노드의 이웃의 표본을 뽑는 것이다. 그러므로 CLARANS 알고리즘도 CLARA 알고리즘처럼 표본 추출에 의존된다(Han and Kamber, 2000).

3. 예제를 통한 비교분석

3.1 분석 방법

본 논문에서 사용되는 데이터는 입력변수만 있는 데이터로 목표변수가 있을 경우 제외하고 군집 분석을 하였다. 데이터는 아래의 〈표 5〉의 데이터를 사용해 평가한다. k-평균 알고리즘과 메도이드에 의한 알고리즘은 연속형 변수들로만 군집분석을 할 수 있으므로, 입력 변수들은 모두 연속형인 데이터를 사용하였다.

군집이 잘 분류되었는가에 대한 평가 기준은 평균 거리(average distance)와 총 분산(total variance)을 사용하였다. 평균 거리는 각 군집들 각각의 평균 거리를 합하여 총 군집 수로 나눈 값으로, 다시 말해서 각 군집의 평균거리의 평균이고, 앞으로 평균 거리는 이것을 나타내는 말로 사용하겠다. 총 분산은 분산 공분산 행렬에서 대각 원소들을 더한 값이다. 각각의 데이터에 k-평균, PAM, CLARA, CLARANS의 네 가지 알고리즘을 적용시키고, 군집의 수 k는 2개와 3개, 4개로 고정하여 평가하겠다. vehicle 데이터와 segmentation 데이터는 이상치가 존재하므로 이상치가 있는 경우와 없는 경우 모두 실험을 해 보았다.

<표 5> 예제 데이터

데이터 이름	객체 수	입력변수
vehicle	846	18
segmentation	1000	18
glass	214	9
ionosphere	351	34
letter	1100	16
sonar	208	60
wine	178	13

3.2 분석 결과

각각의 데이터에 군집의 수를 달리하며 네 가지 알고리즘에 적용시킨 결과는 다음과 같다. 각각의 표들에 진하게 칠한 부분이 각 군집 수 별 평균 거리와 분산이 가장 작은 두 개의 값을 각각 표시한 것이다. vehicle 데이터의 실험 결과는 〈표 6〉에서 제시하고 있다. 여기에서 각 알고리즘들간의 평균 거리와 분산을 비교해 보면, 군집

수에 따라 차이가 있지만 대체적으로 PAM 알고리즘과 CLARANS 알고리즘의 평균 거리와 분산이 나머지 두 알고리즘보다 대체적으로 작게 측정되었음을 알 수 있다.

segmentation 데이터의 실험 결과는 〈표 7〉에서 제시하고 있다. 여기에서 각 알고리즘들간의 평균 거리와 분산을 비교해 보면, 군집 수에 따라 차이가 있지만 대체적으로 k-평균 알고리즘과 CLARANS 알고리즘의 평균 거리와 분산이 나머지 두 알고리즘보다 대체적으로 작게 측정되었음을 알 수 있다.

<표 6> vehicle 데이터의 비교 결과

알고리즘 / 군집수	k-평균		PAM		CLARA		CLARANS	
	평균 거리	분산	평균 거리	분산	평균 거리	분산	평균 거리	분산
k=2	76.52362	9591.499	80.77152	9559.135	82.4734	9670.147	79.794	9605.12
k=3	71.32229	6534.245	64.60056	6203.892	68.56543	6227.993	64.1952	6237.519
k=4	59.91864	4613.635	56.11855	4456.854	58.29263	4539.312	56.4237	4509.967

<표 7> segmentation 데이터의 비교 결과

알고리즘 / 군집수	k-평균		PAM		CLARA		CLARANS	
	평균 거리	분산	평균 거리	분산	평균 거리	분산	평균 거리	분산
k=2	97.77234	9839.139	91.7184	9477.385	91.03559	8877.833	90.18463	8785.946
k=3	81.29118	7353.75	81.92808	7404.528	86.37084	8076.916	84.06005	8037.3573
k=4	75.39002	5566.634	72.18646	6057.746	77.1717	6215.627	76.57865	6164.1201

<표 8> glass 데이터의 비교 결과

알고리즘 / 군집수	k-평균		PAM		CLARA		CLARANS	
	평균 거리	분산	평균 거리	분산	평균 거리	분산	평균 거리	분산
k=2	1.724858	5.5604975	1.959022	6.36511	1.995837	5.7712765	1.859465	5.52754
k=3	1.653649	7.6859283	1.770405	5.0054593	1.804097	5.0520976	1.65498	4.90248
k=4	1.248056	5.960817	1.527075	3.9122325	1.563677	3.977652	1.498321	3.90087

<표 9> sonar 데이터의 비교 결과

알고리즘 / 군집수	k-평균		PAM		CLARA		CLARANS	
	평균 거리	분산	평균 거리	분산	평균 거리	분산	평균 거리	분산
k=2	2.277142	7.237232	2.341692	7.172479	2.371954	7.051941	2.218137	6.957284
k=3	2.126071	6.7851653	2.142651	6.1984623	2.174158	6.3543526	2.058717	6.28468
k=4	2.032764	6.0497797	1.994861	5.667519	2.032012	5.6729402	2.03097	5.684320

<표 10> letter 데이터의 비교 결과

알고리즘 / 군집수	k-평균		PAM		CLARA		CLARANS	
	평균 거리	분산	평균 거리	분산	평균 거리	분산	평균 거리	분산
k=2	8.159776	71.215005	8.66553	71.710905	8.735498	72.228145	8.69155	70.28164
k=3	7.710129	64.11755	8.198726	65.47579	8.395326	65.40661	8.22453	61.84893
k=4	7.496921	61.496447	7.856442	60.073695	8.306868	62.772815	8.01667	59.98132

<표 11> sonar 데이터의 비교 결과

알고리즘 / 군집수	k-평균		PAM		CLARA		CLARANS	
	평균 거리	분산	평균 거리	분산	평균 거리	분산	평균 거리	분산
k=2	1.147331	1.4036435	1.241276	1.5958155	1.271457	1.6309325	1.178196	1.58912
k=3	1.04023	1.2183316	1.173633	1.2381266	1.191687	1.2358753	1.068845	1.0069172
k=4	0.995379	1.1286134	1.119825	1.1592071	1.117632	1.1229765	0.984535	0.987382

<표 12> wine 데이터의 비교 결과

알고리즘 / 군집수	k-평균		PAM		CLARA		CLARANS	
	평균 거리	분산	평균 거리	분산	평균 거리	분산	평균 거리	분산
k=2	135.4235	28698.915	139.4664	29887.675	140.0105	29887.675	137.5197	27946.837
k=3	104.3782	16275.72	96.61018	15187.0723	96.647	15416.161	89.1297	15279.359
k=4	73.51851	8900.1815	73.13004	9078.45975	73.54931	9078.4597	72.91547	8973.5948

glass 데이터의 실험 결과는 〈표 8〉에서 제시하고 있다. 여기에서 각 알고리즘들간의 평균 거리와 분산을 비교해 보면, 군집 수에 따라 차이가 있지만 대체적으로 k-평균 알고리즘과 CLARANS 알고리즘의 평균 거리와 분산이 나머지 두 알고리즘보다 대체적으로 작게 측정되었음을 알 수 있다.

ionosphere 데이터의 실험 결과는 〈표 9〉에서 제시하고 있다. 여기에서 각 알고리즘들간의 평균 거리와 분산을 비교해 보면, 군집 수에 따라 차이가 있지만 대체적으로 PAM 알고리즘과 CLARANS 알고리즘의 평균 거리와 분산이 나머지 두 알고리즘보다 대체적으로 작게 측정되었음을 알 수 있다.

letter 데이터의 실험 결과는 〈표 10〉에서 제시하고 있다. 여기에서 각 알고리즘들간의 평균 거리와 분산을 비교해 보면, 군집 수에 따라 차이가 있지만 대체적으로 k-평균 알고리즘과 PAM 알고리즘의 평균 거리와 분산이 나머지 두 알고리즘보다 대체적으로 작게 측정되었음을 알 수 있다.

sonar 데이터의 실험 결과는 〈표 11〉에서 제시하고 있다. 여기에서 각 알고리즘들간의 평균 거리와 분산을 비교해 보면, 군집 수에 따라 차이가 있지만 대체적으로 k-평균 알고리즘과 CLARANS 알고리즘의 평균 거리와 분산이 나머지 두 알고리즘보다 대체적으로 작게 측정되었음을 알 수 있다.

wine 데이터의 실험 결과는 〈표 12〉에서 제시하고 있다. 여기에서 각 알고리즘들간의 평균 거리와 분산을 비교해 보면, 군집 수에 따라 차이가 있지만 대체적으로 k-

평균 알고리즘과 CLARANS 알고리즘의 평균 거리와 분산이 나머지 두 알고리즘보다 대체적으로 작게 측정되었음을 알 수 있다.

데이터에 알고리즘을 적용하여 나온 결과를 종합해 보면 아래의 <표 13>과 같다. 여기에서 알고리즘은 네 개의 알고리즘 중 평균 거리와 분산이 낮게 나온 두 알고리즘이다. 7개의 데이터에서 CLARANS 알고리즘이 공통적으로 작게 나온 것을 알 수 있다.

<표 13> 예제를 통한 결과 비교

데이터 이름	알고리즘	
vehicle	PAM	CLARANS
segmentation	k-평균	CLARANS
glass	k-평균	CLARANS
ionosphere	PAM	CLARANS
letter	k-평균	PAM
sonar	k-평균	CLARANS
wine	k-평균	CLARANS

다음은 이상치가 있는 경우에 대하여 분석한 결과이다. vehicle 데이터와 segmentation 데이터에 이상치가 존재하므로 두 데이터에 대하여 네 가지 알고리즘을 적용하여 군집 분석을 해보았다. 〈표 14〉는 이상치가 있는 vehicle 데이터에 알고리즘들을 적용하여 군집분석을 해본 결과이고, 〈표 15〉는 이상치가 있는 segmentation 데이터의 결과이다. 먼저 이상치가 있는 데이터는 분산값이 크게 될 것이라고 예상할 수 있다. 그러나 각 알고리즘의 비교를 위하여 편의상 분산값이 큼에도 불구하고 그대로 사용하였다. 따라서 이상치가 있는 데이터의 경우 주로 평균거리 측정방법을 중심으로 비교하였다.

<표 14> 이상치가 있는 vehicle 데이터의 비교 결과

알고리즘 / 군집수	k-평균		PAM		CLARA		CLARANS	
	평균 거리	분산	평균 거리	분산	평균 거리	분산	평균 거리	분산
k=2	180.6117	29516.48	119.942	340237.3	120.8943	334753.7	119.521	3538.266
k=3	179.4243	911556.9	96.84613	240770	97.53197	240237.7	96.2734	240491.3
k=4	90.16701	14590.76	60.67178	22634.93	207.4478	393083	61.7523	25396.97

<표 15> 이상치가 있는 segmentation 데이터의 비교 결과

알고리즘 / 군집수	k-평균		PAM		CLARA		CLARANS	
	평균 거리	분산	평균 거리	분산	평균 거리	분산	평균 거리	분산
k=2	118.7719	11511.46	92.8614	14091.46	92.3847	13431.76	92.01764	13622.97
k=3	100.0708	10841.56	82.65367	12665.15	87.52656	13218.67	86.0455	13300.92
k=4	81.80669	9376.754	73.46778	13106.38	78.36584	12840.74	76.54689	12979.93

<표 14> 에서 각 알고리즘들의 평균 거리와 분산을 비교해보자. 평균을 이용한 k-평균 알고리즘의 평균 거리보다 메도이드를 이용한 PAM 알고리즘의 평균 거리가 군집 수에 관계없이 작아지는 경향이 나타났다. 또한, 메도이디드를 이용한 CLARA 알고리즘, CLARANS 알고리즘도 k-평균 알고리즘과 비교해 보았을 때, 평균 거리가 모두 작아졌다. 이상치가 있는 vehicle 데이터에서 평균 거리가 가장 낮은 알고리즘은 CLARANS 알고리즘임을 알 수 있다. 〈표 15〉 에서도, 평균을 이용한 k-평균 알고리즘보다 메도이드를 이용한 나머지 세 가지 알고리즘의 평균 거리가 군집 수에 관계없이 대부분 작아졌고, 그중 CLARANS 알고리즘의 수치들이 나머지 PAM과 CLARA 알고리즘보다 작게 측정된 것을 알 수 있다.

4. 결 론

알고리즘들의 특징을 알기 위해 평균 거리와 분산을 측정해본 결과 몇 가지 결론을 제시할 수 있다. 여러 데이터에 알고리즘들을 적용시켰을 때, 메도이드를 이용한 알고리즘에서는 CLARANS 알고리즘의 평균 거리와 분산 모두가 PAM과 CLARA 알고리즘보다 거의 대부분 평균 거리와 분산보다 작게 나왔다. 그 다음으로는 CLARA 알고리즘의 평균 거리와 분산이 작았고, PAM 알고리즘이 두 알고리즘보다 약간 높은 평균 거리와 분산이 나왔다. 평균을 이용한 k-평균 알고리즘도 vehicle 데이터와 ionosphere 데이터 두 개를 제외한 모든 데이터에서 평균 거리와 분산이 작게 나왔다. 분석 결과를 종합해 볼 때 메도이드를 이용한 CLARANS 알고리즘이 나머지 알고리즘보다 평균 거리가 더 짧고, 분산도 작아지는 경향이 나타났다. 결과적으로, k-평균 알고리즘과 CLARANS 알고리즘이 데이터나 군집의 수에 관계없이 좋은 결과가 나오는 경향이 나타났다.

이상치가 있는 데이터들에서는 메도이드를 이용한 알고리즘의 평균 거리가 평균을 이용한 k-평균 알고리즘의 평균 거리보다 대부분 작았으며, PAM 알고리즘보다는 CLARA 알고리즘의 평균 거리가 대부분 작게 나왔고, CLARA 알고리즘보다는 CLARANS 알고리즘의 평균 거리가 대부분 작게 나왔다. 메도이드를 이용한 알고리즘을 적용시켰을 때, k-평균 알고리즘보다 분산이 대부분 커졌다. 하지만 평균 거리가 작아지는 것이 데이터들의 분포가 메도이드에 가까이 위치하는 것이므로 분산이 약간 커지더라도 메도이드를 이용한 알고리즘이 이상치가 있는 데이터에서는 군집분석의 결과에 더 좋다.

이상치는 나머지 데이터들과는 차별적인 성격을 가지면서 동떨어져 있는 데이터이

다. 경우에 따라서는 분석에서 제외하지만, 이상치가 어떤 과학적인 정보를 가지고 있는 경우도 많으므로, 이러한 경우에는 제외할 수 없다. 그러므로 이상치가 존재하지만 제거할 수 없는 데이터에서는 이상치를 제외하고 군집분석을 적용하기보다는 메도이드를 이용한 알고리즘을 적용하는 것이 더 효과적이다.

여러 알고리즘에 대한 특성을 데이터에 군집 수만을 달리하여 총 분산과 평균거리로 평가하여 일반화시킨다는 것은 어려움이 따른다. 따라서 다양한 특징을 가지는 데이터들을 사용하여 거리 공식이나 표본추출 방법 또는 평가 방법 등을 달리하여 여러 각도로 비교하여 평가해 보아야 하겠다.

참고문헌

1. Chu. S., Roddick. J. and Pan. J. S. (2002), An Efficient K-Medoids-Based Algorithm Using Previous Medoid Index, Triangular Inequality Elimination Criteria, and Partial Distance Search. : DaWak , LNCS 2454, pp. 63 - 72.
2. Cox. D. R. (1957), *Note on Grouping*, Journal of the American Statistical Association, 52, 543-547.
3. Dillon. W. and Goldstein. M. (1984), *Multivariate Analysis Methods and Applications*, John. Wiley & Sons, Inc.
4. Ester. M., Kriegel. H.-P. and Xu. X. (1995), *Knowledge discovery in large spatial databases : Focusing techniques for efficient class identification*. In Proc. 4th Int. Symp. Large Spatial Database(SSD' 95), pages 67-82, Portland, ME, Aug.
5. Fisher, W. (1958), *On grouping for maximum homogeneity.* Journal of the American Statistical Association. 53: 789-798.
6. Hartigan. J. A. (1975), *Clustering Algorithms*, New York : Wiley.
7. Han. J. and Kamber. M. (2000), *Data Mining : Concepts and Techniques*, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufman Publishers.
8. Huang. Z. (1997), *Clustering Large Data Sets with Mixed numeric and Categorical Values*, Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp.21-34.
9. Huang. Z. (1997b), *A Fast Clustering Algorithm to Cluster Very Large Catgorical Data Sets in Data Mining*, Workshop on Research Issues on Data Mining and Knowledge Discovery.
10. Johnson. R. and Wichern. D. (2002), *Applied Multivariate Statistical Analysis*, Prentice Hall, Inc.
11. Kaufman L. and Rousseeuw. P. J. (1990), *Find Groups in Data : an Introduction to Cluster Analysis*, John Wiley & Sons.
12. MacQueen. J. (1967), *Some methods for classification and analysis of multivariate observation*. Proc. 5th Berkeley Symp. Math. Statist. Prob.,

1:128 - 297.
13. Ng. R. and Han. J. (1994), *Efficient and Effective clustering Methods for Spatial Data Mining.* In Proc. 1994 Int. Conf. Very Large Data Bases (VLDB'94), pages 144-155, Santiago. Chile, Sept.
14. Ng. R. and Han. J. (2001), *CLARANS : A Method for Clustering Objects for Spatial Data Mining*

[2003년 3월 접수, 2003년 10월 채택]