이학석사 학위논문

군집 분석 방법에 대한 비교 연구

A Comparative Study of Clustering Analysis Algorithms.

지도교수 김 재 환

2016년 2월

한국해양대학교 대학원

데이터정보학과

양 태 민

본 논문을 양태민의 이학석사 학위논문으로 인준함.

위원장 : 이학박사 박 찬 근 (인)

위 원 : 공학박사 김 재 환 (인)

위 원 : 이학박사 김 익 성 (인)

2015년 11월 24일

한국해양대학교 대학원

A Comparative Study of Clustering Analysis Algorithms.

Yang, Tae Min

Department of Data Information
Graduate School of Korea Maritime and Ocean University

Abstract

Clustering analysis is a widely used in data mining to classify data into categories on the basis of their similarity. Its applications broadly range from pattern recognition to microarray, multimedia, bibliometrics, bioinfomatics, and astronomy. Through the decades, many clustering techniques, such as hierarchi cal and non-hierarchical algorithm have been developed. Recently, fast search by density peaks of clustering algorithm was presented in the science journal. In this thesis, we perform a comparative study of the performance of the fast search and the existing methods on the benchmark data sets in the literature. From computational experiments, we notice that the accuracy of the fast search is more or less sensitive to the value of parameters for the cluster centers.

Keywords : clustering analysis, fast search by density peaks, K-means, K-medoids

목 차

그 림 목 차

표 목 차

제1장 서 론

1.1 연구 배경

군집분석은 범주(class)의 사전 정보가 알려지지 않는 상태(unsupervised learning)에서 특성들을 파악하여, 유사성이 높은 객체를 군집으로 분류하는 방법이다. 군집분석은 마이크로어레이 등의 의학 분야를 비롯하여 다양한 분야에 적용되고 있다[1].

군집분석방법은 크게 계층적 방법(hierarchical method)과 비계층적 방법(non-hierarchical method)으로 크게 분류할 수 있다. 계층적 방법은 군집의 개수를 미리 정하지 않고 분류하는 기법으로서 많은 방법들이 연구되었다[2]. Sorensen [3]이 단일연결법(single linkage method)과 완전연결법(complete linkage method)을 제시하였다. 단일연결법은 객체간의 가장 짧은 거리가 작을수록 두 군집이 유사하다고 평가하는 방법으로서 최단거리법(nearest neighbor method)이라고도 한다. 완전연결법은 이와 반대로 객체간의 가장 먼 거리가 작을수록 두 군집이 더 유사하다고 평가하는 최장거리법(farthest neighbor method)이다. Sokal과 Michener[4]는 두 군집의 객체간의 평균거리를 사용하는 평균연결법(average linkage method)과 중심좌표(centroid)를 이용한 중심연결법(centroid linkage method)을 고안하였다. 이외에도 워드 방법(Ward's method)[5]과 Kaufman과 Rousseeuw[2]가 제안한 다이아나(DIANA) 방법 등이 있다.

비계층적 방법은 군집의 개수를 미리 설정하고 분류하는 기법으로서 K-means 군집방법[6], K-medoids 군집방법[7], DBSCAN(Density-based spatial clustering of applications with noise)([8], [9]), 다중 가우스의 EM 군집방법(Multi-Gaussian with Expectation-Maximization)([10], [11]) 등이 개발되었다.

K-means 군집방법은 각 군집의 중심좌표(centroid)를 고려하는 반면, K-medoids 군집방법에는 각 군집의 대표객체(medoid)를 고려하는 기법이다. 또한, DBSCAN([8], [9])은 밀도를 기반으로 하는 군집방법으로, 다양한 모양과 크기를 가진 군집을 구분하는데 용이하다. EM 군집방법([10], [11])은 로그 가능도 함수(log-likelihood function)를 사용하여 모델의 적합성을 평가했다.

최근 사이언스저널에 새로운 효율적인 군집 분석기법인 fast search[12]가 발표되었다. 따라서 본 논문에서는 이 fast search방법과 기존의 군집분석방법과의 성능 비교를 UC Irvine[13]의 다양한 데이터셋을 이용하여 분석하고자 한다.

1.2 연구 내용

본 논문의 구성은 다음과 같다.

2장에서는 계층적 군집방법, 비계층적 군집방법 등 다양한 군집분석에 대해 상세하게 설명한다. 계층적 방법은 각 객체를 하나의 군집으로 시작하여 결국 모든 객체가 하나의 군집이 되도록 분류 하는 것으로 분류할 군집의 개수를 미리 정하지 않는다. 이 방법에는 연결법과 워드 방법이 있는데, 이 두 개의 군집방법은 시각화 한 트리 형태의 덴드로그램(dendrogram)을 이용하여 적절한 군집의 개수를 정한다. 연결법은 객체 간의 거리행렬로 유사성 척도를 산출하는 반면, 워드 방법은 군집의 제곱합을 활용한다. 비계층적 군집방법은 분류할 군집의 개수를 미리 설정하여 분류하는 방법으로서, K-means 군집과 K-medoids 군집 등의 대표적인 방법이 있다. 또한, DBSCAN, EM(expectation-maximization) 군집방법과 사이언스 저널에 최근 발표된 fast search에 대해서도 언급한다.

3장에서는 위에서 언급한 군집 분석 방법에 대해 성능을 비교 분석하였다. UC Irvine[13]의 대표적인 데이터셋인 Iris를 비롯한 다양한 데이터셋을 이용하여 SVM(suppot vector machine)([14], [15], [16]) 과 함께 그 성능인 분류의 정확도와 속도를 비교 분석하였다.

마지막으로 4장에서는 결론 및 추후연구에 대해 언급한다.

제2장 군집 분석 방법

2.1 군집분석의 개념

빅 데이터 분석 기법 중 하나인 군집분석은 범주의 사전정보가 밝혀지지 않은 상태에서 각 객체들의 속성을 파악하고, 그 속성이 높은 객체들끼리 군집을 만든다. 군집에 속한 객체들과 다른 군집에 속한 객체들 간의 차이점을 분석하는 탐색적 통계분석방법이다.

군집분석의 문제점 중 하나는 군집형태가 매우 다양하다는 점이다. 그림 2.1의 (a)의 경우에는 각 군집이 구형이고, 서로 겹치는 부분이 없이 각각의 군집 간에 명확히 구분된다. (b)의 경우에는 각 군집이 긴 타원형이며 각 객체간의 거리를 유클리드 거리로 측정하면, 객체 B는 C와 같은 군집임에도 불구하고 A와 더 가깝다고 판단하게 된다. (c)의 경우에는 객체 A와 B가 서로 근접하게 위치해 있어서 군집방법에 따라 하나의 군집 또는 두 개의 군집으로 판단할 수 있다.

군집분석의 방법에서의 거리(distance)는 객체에 대한 유사성(similarity) 혹은 비유사성(dissimilarity)을 측정하는 척도인데, 이와 관련되어 다양한 척도들이 존재한다. 일반적으로 속성이 3개 이하인 경우에는 그래프 등을 이용해 시각적으로 군집관계를 파악할 수 있다. 그러나 속성의 수가 많은 경우에는 시각적 관측이 어렵기 때문에 연구자의 분석결과에 대한 주관적 판단이 중요하게 된다. 또한, 각 군집방법에 따라 서로 다른 결과를 나타낼 수 있기 때문에 각 군집방법이 가지는 특성을 잘 이해하는 것이 매우 중요하다.

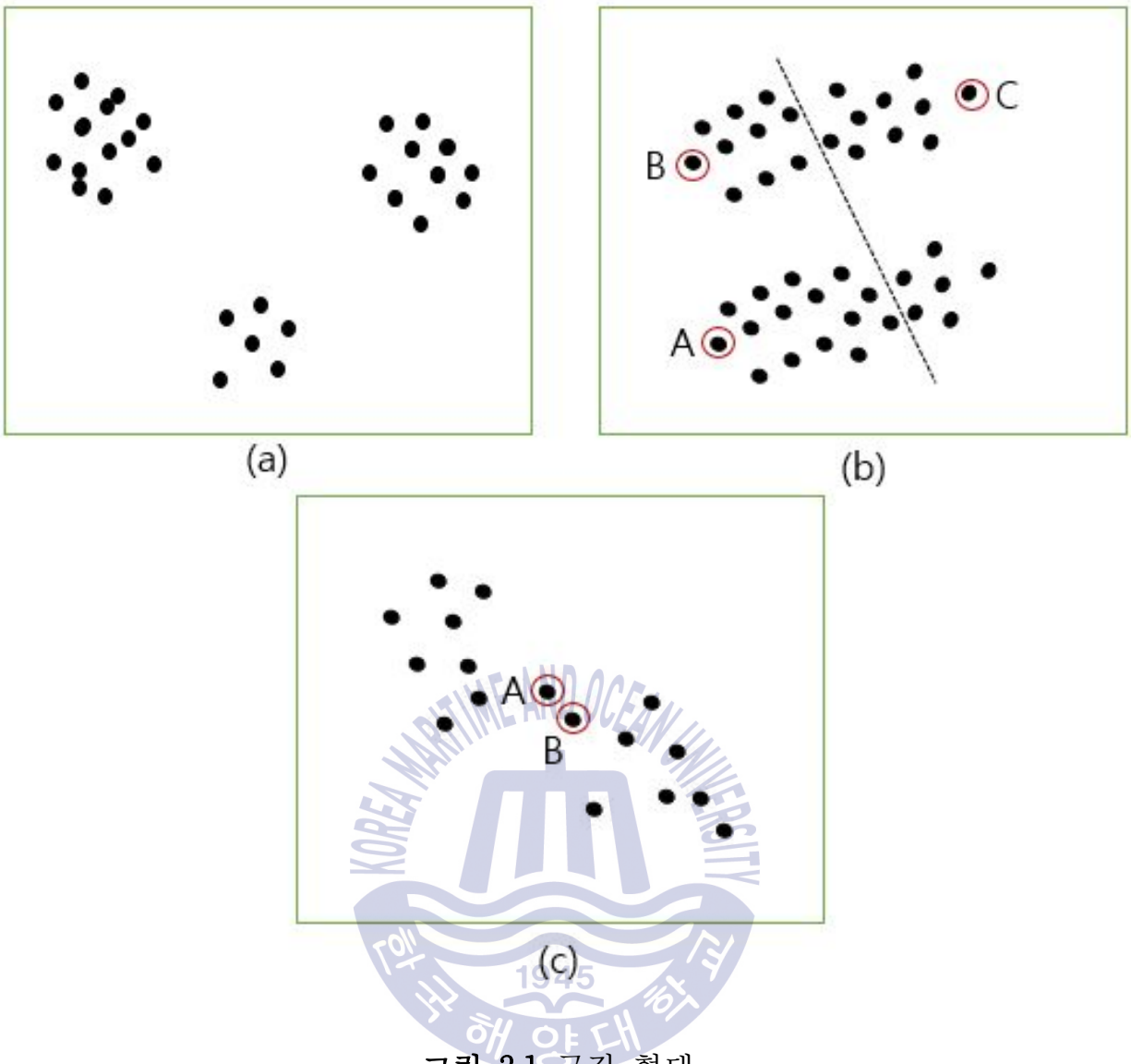


그림 2.1 군집 형태

2.2 계층적 군집방법 (Hierarchical Clustering)

계층적 군집방법은 측정변수, 척도 등을 사용하여 유사성이 가장 높은 객체들을 순차적으로 분류하여 군집을 형성하게 되고, 이러한 과정을 반복하는 단계적 절차를 사용한다. 그러나 객체가 한번 군집에 소속하게 되면 다른 군집으로 이동이 불가능하게 되어 이상치(Outlier)가 제거되지 않는 문제점을 가지고 있다.

비계층적 군집방법(Non-hierarchical Clustering)과의 차이점은 군집분석을 실시할 때, 계층적 군집방법은 초기 군집의 수를 설정 하지 않고 실시하는 반면, 비

계층적 군집방법은 초기 군집수를 설정한다.

계층적 군집 방법에서는 연결법, 워드 방법(Ward's Method), 다이아나(DIANA) 등 다양한 방법이 존재하나, 본 논문에서는 연결법(평균 연결법)과 워드 방법을 fast search와 비교분석 하였다.

연결법을 기술하기 위한 기호는 다음과 같다[17].

C_i : 군집 i

$|C_i|$: 군집 i의 객체 수

$d(u,v) = d(x_u, x_v)$: 객체 u와 객체 v의 거리

$d(C_i, C_j)$: 군집 C_i와 군집 C_j의 거리

(1) 단일 연결법(single linkage method)

$$d(C_i, C_j) = \min_{u \in C_i, v \in C_j} d(u, v) \quad (1)$$

단일연결법은 최단 거리법(nearest neighbor method)이라고 한다. 각 군집간의 유사성 척도로 두 군집간의 모든 객체 쌍과의 거리 중 가장 가까운 거리를 사용하는데, 그 거리가 작을수록 두 군집의 유사성이 높다고 평가하여 군집을 형성하게 된다[17].

(2) 완전 연결법(complete linkage method)

완전연결법은 Sorensen[12]에 의해 제안된 것으로 최장 거리법(farthest neighbor method)이라고 한다. 단일연결법과 정반대로 각 군집간의 유사성 척도로 두 군집간의 모든 객체 쌍과의 거리 중 가장 먼 거리를 사용한다. 그 거리가 작을수록 두 군집의 유사성이 높다고 평가하여 군집을 형성하게 된다[17].

$$d(C_i, C_j) = \max_{u \in C_i, v \in C_j} d(u, v) \quad (2)$$

(3) 평균 연결법(average linkage method)

평균연결법은 Sokal과 Michener[4]에 의해 제안된 것으로, 각 군집간의 유사성 척도로 두 군집의 모든 객체 쌍과의 평균 거리를 사용한다[17].

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{u \in C_i, v \in C_j} d(u, v) \quad (3)$$

(4) 중심 연결법(centroid linkage method)

중심연결법은 평균연결법을 제안한 Sokal과 Michener[4]에 의해 알려졌으며, 두 군집간의 유사성 척도로 두 군집을 이루는 객체들의 무게중심에 해당되는 중심좌표(centroid)를 사용한다[17].

$$c_i = (\overline{X}_1^{(i)}, \overline{X}_2^{(i)}, \ldots, \overline{X}_p^{(i)}) \quad \overline{X}_a^{(i)} = \frac{1}{|C_i|} \sum_{j \in C_i} X_{aj} \quad a = 1, \ldots, p \quad (4)$$

$$d(C_i, C_j) = d(c_i, c_j) \quad (5)$$

2.2.1 연결법의 군집 방법

2.2절에서 언급한 연결법은 각각의 군집들과의 유사성 척도 방법이 다를 뿐, 군집 알고리즘은 동일하게 적용된다.

연결법 군집방법 알고리즘 다음과 같다[17].

[초기 단계] 군집분석을 하고자 하는 연결법을 선정하고, 객체들을 각각의 군집으로 가정한다.

[단계 1] 현재의 군집결과에 있는 모든 군집 간의 객체 쌍에 대하여 각 연결법에 해당하는 유사성 척도를 이용하여 새로운 군집을 형성하고, 군집결과를 수정한다.

[단계 2] 전체 군집의 수가 1이 되면 군집분석을 멈춘다. 그렇지 않으면 [단계 1]을 반복한다.

연결법은 각 객체를 하나의 군집으로 시작하고, 모든 객체가 하나의 군집이 될 때 까지 반복된다. 군집 수는 단계별 군집분석결과를 시각적으로 표현한 덴

드로그램(dendrogram)을 사용하여 결정된다.

2.2.2 워드 방법(Ward's method)

워드 방법의 초기 단계와 최종 단계는 연결법과 동일하다. 연결법과의 차이점은 각 군집간의 유사성 척도로 군집의 제곱합을 활용한다는 점이다. 현재 군집결과에 있는 군집들을 통합한다고 할 때 새로운 전체 군집의 제곱합을 산출한 후 그 값이 가장 작을수록 두 군집간의 유사성이 높다고 평가를 하여 서로 군집을 통합하게 된다.

각 군집의 제곱합은 다음과 같은 제곱유클리드 거리로 표현 된다[17].

$$SS(C_i) = \frac{1}{2|C_i|} \sum_{u,v \in C_i} d^2(u,v) \tag{6}$$

전체 군집의 제곱합(SSW)은 다음과 같다[17].

$$SSW = \sum_{i=1}^{k} SS(C_i) \tag{7}$$

워드 군집방법 알고리즘의 단계는 다음과 같다[17].

[초기 단계] 각 객체들을 각각의 군집으로 가정한다.

[단계 1] 현재의 군집결과에 있는 모든 군집에 대하여 전체 군집의 제곱합(SSW)을 산출한다. 그 값이 가장 작은 군집을 새로운 군집으로 형성하여 군집결과를 수정한다.

[단계 2] 전체 군집의 수가 1이 되면 군집분석을 멈춘다. 그렇지 않으면 [단계 1]을 반복한다.

2.3 비계층적 군집방법 (Non-hierarchical Clustering)

2.3.1 K-means 군집방법

K-means 군집방법은 무작위로 초기 군집을 선정한 후 그 군집의 중심좌표(centroid)를 산출하여 각각의 객체들과의 거리가 가장 가까운 군집에 배정하는 반복적 군집방법이다.

K-means 군집방법 알고리즘은 다음과 같다[17].

[초기 단계] 다음과 같은 2개의 규칙에 의해서 초기 군집을 선정하고 그 군집의 중심좌표를 산출한다.

무작위 선정 방법 : 전체 객체들 중 무작위로 초기 군집을 선정한다.

외각 객체 선정 방법: 전체 객체들의 중심좌표에서 가장 멀리 있는 객체를 초기 군집들로 선정한다.

[단계 1] 각각의 객체들에 대해서 위 단계에서 산출한 초기 군집들의 중심좌표와의 거리를 산출한 후 가장 가까운 군집에 그 객체를 배정한다.

[단계 2] 새로 선정된 군집들의 중심좌표를 산출한다.

[단계 3] 새로 산출된 군집중심좌표 값과 이전의 군집 중심 좌표값과 비교하여 변화가 없으면 군집분석을 멈춘다. 그렇지 않으면 [단계 1]을 반복한다.

객체들을 어떤 군집에 배정하는 것이 전체 거리를 최소로 하는지에 대한 최적화 문제로 고려될 수 있다.

최적화 문제로 정식화한 것은 다음과 같다[17].

$$I_{ij} = \begin{cases} 1 & \text{객체} i \text{가 군집} j \text{에 배정될 때} \\ 0 & \text{그밖의 경우} \end{cases}$$

$$Min\, Z = \sum_{i=1}^{n}\sum_{j=1}^{P} d(x_i, c_j) I_{ij} \tag{8}$$

Subject to

$$c_j = \frac{\sum_{i=1}^{n} x_i I_{ij}}{\sum_{i=1}^{n} I_{ij}}, \quad j = 1, \ldots, P \tag{9}$$

$$\sum_{j=1}^{P} I_{ij} = 1, \quad i = 1, \ldots, n \tag{10}$$

$$\sum_{i=1}^{n} I_{ij} \geq 1, \quad j = 1, \ldots, P \tag{11}$$

식 (8)은 각각의 n개의 객체와 그에 해당하는 군집의 중심좌표와의 거리 합을 나타낸 것이다. 식 (9)는 중심좌표 산출하는 식이며, 식 (10)은 각 객체가 P개 군집들 중 어느 하나에 반드시 속해야 한다는 것을 의미한다. 식 (11)은 한 군집에 하나 이상의 객체를 포함되어야 한다는 것을 의미한다.

P개의 군집중심좌표가 산출되면 [단계 1]에서는 식 (8)을 최소로 하는 새로운 I_{ij}을 정하고, [단계 2]에서는 식 (9)를 사용하여 새로운 군집중심좌표를 산출한다. K-means 군집방법은 계산효율이 대체적으로 양호하나 이상치(outlier)가 존재할 때 군집결과가 좋지 않을 수 있는 것으로 알려져 있다.

2.3.2. K-medoids 군집방법

K-medoids 군집방법은 K-means 군집방법과 다르게 대표객체(medoid)를 고려하는 비계층적 군집방법 중 하나이다. 대표객체는 각 군집에 속하는 객체들과

다른 객체들과의 평균(또는 전체)거리가 최소가 되는 객체를 말한다.

K-medoids 군집방법은 초기에 각 객체들을 정해진 군집 수로 분류하는데, 각각의 객체와 군집의 대표객체와의 거리의 총합을 최소로 하는 것이 유사성이 높다고 평가하여 군집에 배정하는 방법이다.

K-medoids 군집방법에 관한 대표적인 알고리즘은 다음과 같다.

1) PAM (Partitioning Around Medoids) 알고리즘

PAM 알고리즘은 Kaufman과 Rousseeuw[2]에 의하여 제안된 것으로, 초기 대표객체를 설정하는 BUILD 단계와 더 좋은 군집 해를 찾아가는 SWAP으로 구성되어 있다[17].

< BUILD >

[초기 단계] 각 객체간의 서로 거리를 구한 후, 거리 합이 가장 작은 객체 하나를 대표객체로 선정한다. 이 때, 선정된 대표객체집합을 C라 한다.

[단계 1] 대표객체로 선정되지 않은 객체 j에 대하여, 대표객체로 선정된 객체들 중 객체 j에 가장 가까운 거리 D_j를 구한다.

$$D_j = \min_{k \in C} d(j,k),\ j \notin C \tag{12}$$

대표객체로 선정되지 않은 두 객체 i, j에 대하여 다음을 산출한다.

$$S_{ji} = \max(D_j - d(j,i), 0) \quad i, j \notin C \tag{13}$$

이는 객체 i가 추가로 대표객체가 된다고 할 때, 객체 j의 입장에서 거리 감소량이다. 만약 S_{ji}의 값이 음수가 나온다면, S_{ji}값은 0이다.

[단계 2] 다음과 같이 거리감소량이 가장 큰 객체 c을 대표객체에 포함시키고,

$$m = \arg max_{i \notin C} \sum_{j \notin C} S_{ji} \tag{14}$$

대표객체집합을 수정한다. ($C \leftarrow C \cup \{c\}$).

[단계 3] 초기 정해진 수의 대표객체가 선정되었으면 군집분석을 멈춘다. 그렇지 않은 경우에는 [단계 1]로 되돌아간다.

< SWAP >

[단계 1] 대표객체로 선정되지 않은 임의의 객체 j(단, $j \neq h$)에 대해 다음과 같이 산출 한다.

S_{jih} = (i와 h를 교환 후 객체 j와 대표객체와의 거리)
- (교환 전 객체 j와 대표객체와의 거리)
($j \notin C, i \in C, h \in C$)

S_{jih}는 다음 4개의 경우에 따라 다르게 산출된다.

(ⅰ) 객체 j로부터 i와 h까지의 거리가 대표객체 중 어느 하나와의 거리보다 더 먼 경우

$$S_{jih} = 0 \tag{15}$$

(ⅱ) 객체 j로부터 h까지의 거리가 j에서 두 번째로 가까운 대표객체까지의 거리보다 더 짧은 경우

$$S_{jih} = d(j,h) - d(j,i) \tag{16}$$

(ⅲ) 객체 j로부터 h까지의 거리가 j에서 두 번째로 가까운 대표객체까지의 거리(E_j)보다 더 먼 경우

$$S_{jih} = E_j - d(j,i) \ (\geq 0) \tag{17}$$

(ⅳ) 객체 j로부터 i까지의 거리가 j로부터 적어도 어느 하나의 대표객체까지의 거리보다는 길고, 객체 j에서 h까지의 거리가 j로부터 다른 어떤 대표객체들과의 거리보다 짧은 경우

$$S_{jih} = d(j,h) - D_j \ (< 0) \tag{18}$$

[단계 2] 대표객체 i를 h로 교환하는 경우 총 변화량은 다음과 같다.

$$T_{ih} = \sum_j S_{jih} \tag{19}$$

이때 T_{ih} 값이 최소가 되는 객체 $\hat{i}$와 $\hat{h}$를 찾아 $T_{\hat{i}\hat{h}} < 0$ 이면 교환한 후에 다시 [단계 1]로 되돌아가고, $T_{\hat{i}\hat{h}} \geq 0$ 이면 군집분석을 멈춘다.

2) CLARA (Clustering LARge Applications) 알고리즘

CLARA 알고리즘은 Kaufman와 Rousseeuw[2]에 의해 제안된 것이다. PAM 알고리즘의 SWAP 부분이 모든 경우의 수를 고려하기 때문에 계산 시간이 길다는 단점이 있는데, 그 단점을 보완한 알고리즘이다. CLARA 알고리즘은 최적의 객체 수를 구한 후, PAM 알고리즘을 적용한 방법이다[17].

[단계 1] 전체 데이터셋의 객체들 중에서 일부 객체를 표본으로 추출한 후 PAM알고리즘을 적용하여 초기에 정한 군집의 대표객체를 산출한다.

[단계 2] 전체 데이터셋의 객체들을 초기에 정한 군집에 포함시킨다.

[단계 3] 현재 군집분석결과에 대한 목적함수 값을 산출한다. 이 값이 이전의 목적함수 값보다 작으면 현재의 대표객체들을 택하고, 그렇지 않으면 이전에 구한 대표객체를 유지한다.

3) CLARANS 알고리즘
(Clustering Large Applications based on RANdomized Search)

CLARANS 알고리즘 Ng와 Han[18]에 의해 제안된 것이다. 초기에 선정된 대표객체집합에서 시작하여 어떠한 인근집합을 고려하여 목적함수 값이 더 좋아지면 대표객체 후보 집합을 교체하고, 그렇지 않으면 다른 임의의 인근집합을 고려하는 과정을 반복하는 알고리즘이다[17].

[초기 단계] $i \leftarrow 1$, $m \leftarrow \infty$ (m: 최소 목적함수)

[단계 1] 임의의 대표객체집합을 선정하여 C라 하고, C의 인근집합을 구한다.

[단계 2] $j \leftarrow 1$

[단계 3] C의 임의의 인근집합에 대해서 목적함수를 산출한다. 목적함수 값이 음수이면, 인근집합을 C로 두고 [단계 3]으로 되돌아가고, 양수이면 [단계 5]로 넘어간다.

[단계 4] $j \leftarrow j+1$
$j \leq M$(M : 인근집합 수) 이면 [단계 4]반복
$j >$ M 일 때, 현 목적함수 값이 m보다 작으면, $m \leftarrow$ 현 목적함수 값으로 대체한다.

[단계 5] $i \leftarrow i+1$
$i \leq n$ 이면 [단계 2]반복(n : 초기집합 수)
$i >$ n이면 알고리즘을 중지하고, 현재의 C가 최종해가 된다.

4) K-means-like 알고리즘

K-means-like 알고리즘은 Park과 Jun[19]에 의해 제안된 것으로, CLARA 알고리즘과 동일하게 PAM 알고리즘이 계산속도가 느리다는 단점을 보완하기 위한 알고리즘이다. 대표객체의 교체를 반복하는데, K-means 군집방법의 군집 분석 방법을 이용한 군집방법이다[17].

[단계 1] 초기대표객체를 선정하여, 각 객체들을 가장 가까운 대표객체에 배정하여 초기 군집 해를 얻는다.

[단계 2] 현재 군집결과에 의해 새로 배정된 객체들의 대표객체를 구하여 새로운 대표객체로 선정한다.

[단계 3] 각 객체들을 가장 가까운 대표객체에 배정하여 새로운 군집 해를 얻는다. 새롭게 산출된 대표객체들(또는 군집해)이 이전과 동일하면 알고리즘을 멈추고, 그렇지 않으면 [단계 2]를 반복한다.

2.3.3. 퍼지 K-means 군집방법 (Fuzzy K-means Algorithm)

퍼지 K-means 군집방법은 K-means 군집방법과 유사하나, 한 개의 객체가 여러 군집에 속할 가능성을 허용하는 확률 또는 퍼지(fuzzy)상수를 도입한 것이다. P_{ij}를 객체 i가 군집 j에 속할 확률이라 할 때, 아래와 같이 최적화 문제로 정식화 할 수 있다[17].

$$Min\ Z = \sum_{i=1}^{n}\sum_{j=1}^{K} P_{ij}^{m} d(x_i, c_j) \tag{20}$$

Subject to

$$\sum_{j=1}^{K} P_{ij} = 1, \quad i = 1, \ldots, n \tag{21}$$

$$\sum_{i=1}^{n} P_{ij} > 0, \qquad i = 1, \ldots, K \tag{22}$$

$$P_{ij} \in [0,1], \qquad i = 1, \ldots, n;\ j = 1, \ldots, K \tag{23}$$

m은 퍼지상수(fuzziness index)로써, m이 1에 가까울수록 K-means 군집방법과 비슷하게 군집분석이 되며, 반대로 큰 값을 가질수록 각 객체가 동일한 확률로 군집에 배정되게 된다. 일반적으로 m=2가 사용되고 있다. 한편, 중심좌표를 산출하는데 K-means 군집방법과 달리 P_{ij}^{m}을 가중치로 사용하여 산출된다.

$$c_j = \frac{\sum_{i=1}^{n} P_{ij}^{m} x_i}{\sum_{i=1}^{n} P_{ij}^{m}} \tag{24}$$

P_{ij}는 아래와 같이 산출 할 수 있다.

$$P_{ij} = \frac{1}{\sum_{a=1}^{k} \left(\frac{d(x_i, c_j)}{d(x_i, c_a)} \right)^{1/(m-1)}} = \frac{\left[d(x_i, c_j) \right]^{-1/(m-1)}}{\sum_{a=1}^{k} \left[d(x_i, c_a) \right]^{-1/(m-1)}} \tag{25}$$

퍼지 K-means 군집 알고리즘은 다음과 같다.

[초기 단계] P, m을 정한다. 초기 P개의 군집을 임의로 정한다.

$$P_{ij} = \begin{cases} 1 & \text{객체 } i\text{가 군집 } j\text{에 속하면} \\ 0 & \text{아니면} \end{cases}$$

[단계 1] 각 군집들의 중심좌표를 식 (24)를 이용하여 산출한다.

[단계 2] P_{ij}을 식 (25)을 이용하여 새로 산출한다.

[단계 3] 각 객체들에 관해서 P_{ij} 값이 가장 큰 군집에 배정하여 새로운 군집결과를 얻는다. 이전의 군집결과와 비교하여 동일하면 군집분석을 멈추고 , 그렇지 않으면 [단계 1]로 되돌아간다.

제3장 다양한 군집 분석 방법

3.1. DBSCAN 군집방법
(Density-Based Spatial Clustering of Application with Noise)

DBSCAN 군집방법은 Ester[8] 등에 의해 제안된 밀도를 기반으로 하는 군집 분석 방법이다. 이 군집방법은 큰 공간데이터(spatial data)에서 다양한 모양과 크기를 가진 군집들을 구분하는데 용이하며, 이상치(outlier)인 잡음(noise)을 포함한 군집을 밀도를 기준으로 구분한다.

DBSCAN[20]의 5개의 정의를 다음과 같이 정의한다.

[정의 1] ε-이웃해(ε-neighborhood)

한 점이 군집에 포함되기 위해서는 ε에 가까운 또 다른 한 점이 필요하다. p의 ε-이웃해는 p로부터 ε-반경 내에 있는 이웃들의 집합이다.

$$N_{\varepsilon}(p) = \{q \in D | dist(p,q) \leq \varepsilon\} \quad (26)$$

D : 데이터셋

[정의 2] 밀도-접근성(density-reachability)

p가 q로부터 '밀도-접근성' 은 서로 다른 두 점 p, q로부터 다음 두 조건 만족하는 것이다.

$$1)\ p \in N_{\varepsilon}(q) \quad (27)$$

$$2)\ |N_{\varepsilon}(q)| \geq MinPts \quad (28)$$

$MinPts$: 최소객체수

[정의 3] 밀도-연결성(density-connected)

p가 q로부터 '밀도-연결성' 은 p와 q로부터 '밀도-접근성' 한 r이 존재 하는 것이다.

[정의 4] 군집(cluster)

군집을 C라고 가정하면,

1) 모든 p, q에 대해 $p \in C$ 이고, p가 q로부터 '밀도-접근성' 하면 $q \in C$ 이다.

2) 모든 p, q, $p \in C$: p는 q에 '밀도-연결성' 한다.

[정의 5] 잡음(noise)

잡음 N은 전체 데이터셋의 어떤 군집에도 속하지 않는 점을 말한다.

$$N = \{p \in D \mid \forall i : p \notin C_i\} \tag{29}$$

DBSCAN 군집 알고리즘은 다음과 같다.

[초기 단계] 임의의 초기 객체 q를 선정한다.

[단계 1] 초기 객체 q에 대해서 ε-반경내에 $MinPts$를 만족하는 충분한 객체가 있으면 군집으로 배정시킨다. ('밀도-접근성')

[단계 2] 그렇지 않으면, 잡음으로 분류한다.

[단계 3] ε-반경 안에 있는 객체에서 같은 방법으로 ε-반경 안에 $MinPts$를 만족한다면 군집을 확장하고, 만족하지 못하면 잡음으로 정의한다.

3.2. 다중 가우스함수의 EM 군집방법
(Multi-Gaussian with Expectation-Maximization)

EM 군집방법(Expectation-Maximization algorithm)은 Hartley에 의해서 처음 제안되었고, Dempster[11]에 의해서 체계화되었다. 이 군집방법[20]은 매개변수에 관한 추정값으로 로그 가능도(log likelihood)의 기대치를 계산하는 기대치 단계(E-Step)와 이 기대치를 최대화하는 변수 값을 구하는 최대화 단계(M-Step)를 번갈아가면서 적용한다. 반복 과정을 통해서 각 객체들이 혼합 모형(mixture model)에 속할 가능성을 조정하여 최적의 모형을 생성해 간다. K-means는 거리 기반 군집 방법인 것에 비하여 EM 군집방법은 확률 기반 군집방법(probability-based clustering)이다.

3.2.1 혼합 모형 (Mixture Model)

EM 군집방법에서는 각 객체들은 여러 개의 확률 분포 모형(군집)에 속하는 가중치를 가지고 배정한다. 관찰할 수 있는 객체 $x_i\,(i=1,\dots,N\,;x_i\in R^M)$ 에 대해 $P_k(x_i|\theta_k)\ \ (k=1,\dots,K)$를 θ_k로 표현되는 k번째 군집으로부터 객체 x_i가 생성될 확률밀도함수로 정의하면, 군집 분석 모형은 다음과 같은 두 가지 방향으로 나타낼 수 있다.

1) 분류 가능도(classification likelihood) 군집 분석 모형

$$L_{CL}(\theta_1,\dots,\theta_K\,;\alpha_1,\dots,\alpha_K)=\prod_{i=1}^{N}p_{\alpha_i}(x_i\,|\theta_{\alpha_i}) \qquad (30)$$

위 식을 최대화하는 모수의 추정치를 구한다. 여기서, x_i가 k번째 확률밀도함수로부터 생성된다면, $\alpha_i=k$ 이다.
이것은 각각의 객체가 한 개의 확률밀도함수로부터 생성된다고 가정한 모형이다.

2) 혼합 가능도(mixture likelihood) 군집 분석 모형

$$L_{ML}(\theta_1,\ldots,\theta_K;\alpha_1,\ldots,\alpha_K \mid x)=\prod_{i=1}^{N}\sum_{k=1}^{K}\alpha_k p_k(x_i\mid\theta_k) \qquad (31)$$

위 식을 최대화하는 모수의 추정치를 구한다. 이는 혼합 모형으로써, 객체가 여러 확률밀도함수의 '혼합'으로부터 생성된다고 가정한 모형이다. 여기서 '혼합'은 여러 개의 확률 분포가 혼합된 것을 의미한다. 그러나 $\alpha_k(\alpha_k \geq 0, \sum_{k=1}^{K}\alpha_k = 1)$는 객체와 독립적인 변수로 K개의 밀도함수 중 k번째 밀도함수가 선택될 확률이다.

3.2.2 군집 분석 모형

EM 군집방법은 군집에 대한 사전 정보가 알려져 있지 않은 데이터셋에 대해 주어진 가능도를 최대화하기 위한 방법이다. 각 객체가 어떤 집단에 속하는지 모두 알려져 있는 완전 데이터의 경우 다음과 같이 정의한다.

$$c_{ik}=\begin{cases}1 & \text{if } x_i \in z_k\\ 0 & otherwise\end{cases} \qquad (32)$$

가능도는 다음과 같다.

$$L(\theta_1,\ldots,\theta_K;\alpha_1,\ldots,\alpha_K;c_{i1},\ldots,c_{iK} \mid x)=\prod_{i=1}^{N}\sum_{k=1}^{K}(\alpha_k p_k(x_i\mid\theta_k))^{c_{ik}} \qquad (33)$$

(33)은 (30)과 (31)가 결합된 식이라고 볼 수 있다. 즉, 객체가 여러 개의 확률밀도함수의 '혼합'으로부터 생성하되 어떤 확률밀도함수로부터 생성되었는지가 c_{ik}이다.

양변에 로그를 취한 로그 가능도는 다음과 같다.

$$L_{LOG}(\theta_1,\ldots,\theta_K;\alpha_1,\ldots,\alpha_K;c_{i1},\ldots,c_{iK} \mid x)=\prod_{i=1}^{N}\sum_{k=1}^{K}c_{ik}[\log(\alpha_k p_k(x_i\mid\theta_k))] \qquad (34)$$

주어지는 데이터셋은 군집수를 알 수 없으므로, c_{ik} 또한 알 수는 없다. 그러므로 위 식을 바로 최대화할 수는 없고, 위 식의 c_{ik}에 대한 기대치를 최대화한다.

위 식의 기대치를 최대화하는 방향으로 θ_k를 갱신해나가는 것이 바로 위 식을 최대화하는 것과 같다.

그러므로, 객체 x_i와 θ_k가 주어졌을 때 c_{ik}의 평균값인 $\hat{c_{ik}}$를 반복적으로 산출

$$\hat{c_{ik}} = E[c_{ik} | x_i, \theta_1, \ldots, \theta_K] \quad (35)$$

하고, 특정 시점에서 객체 x_i는 K개의 확률밀도함수 중 $\hat{c_{ik}}$가 최대가 되는 (이때 $\hat{c_{ik}}$의 값을 c_{ik}^*라고 하면,) 함수로부터 생성되었다고 정한다. 즉 k번째 군집으로 할당하여 그러한 k에 대해 $c_{ik} = 1$, 나머지 k에 대해 $c_{ik} = 0$ 으로 한다. 이렇게 할당할 수 있는 이유는, 객체 x_i에 대해 $1 - \max_k c_{ik}^*$는 x_i가 현재 할당된 군집에 대해 이 할당의 불확실도(measure of uncertainty)이기 때문이다. 또한 $\hat{c_{ik}}$가 계산되면 위의 가능도를 증가시키는 방향으로 θ_k와 α_k를 갱신한다. 이러한 모형 학습 알고리즘을 EM이라고 한다. EM 군집방법에서 E-Step은 $\hat{c_{ik}}$를 계산하는 과정, 즉, 각 데이터를 적절한 군집으로 할당하는 과정이고, M-Step은 모형을 갱신하는 과정이다.

1과 0 중에서 랜덤하게 c_{ik}를 초기화하고, $\hat{c_{ik}}$ 또한 초기화 한다.

< 반복 >

M-step : (34)식의 c_{ik}에 대한 기대치를 최대화 한다.

$$\hat{\tau_k} \leftarrow \frac{\sum_{i=1}^{N} \hat{c_{ik}}}{N}$$

θ_k 와 $\hat{\theta_k}$ 를 갱신한다. ..... (A)

E-step : M-step으로부터 얻어진 추정치로 $\hat{c_{ik}}$ 를 계산한다.

$$\hat{c_{ik}} \leftarrow \frac{\hat{\tau_k} p_k(x_i | \hat{\theta_k})}{\sum_{k'=1}^{K} \hat{\tau_{k'}} p_{k'}(x_i | \hat{\theta_{k'}})}$$

기준치 수렴에 만족하거나, 최대값을 가질 때까지 반복한다.

그림 3.1 혼합 모형에 의한 EM 군집방법 알고리즘[22]

3.2.3 다중 가우스함수의 EM 군집방법

혼합 모형을 통한 군집분석에서 θ_k는 평균과 분산으로 표현될 수 있다. 일반적으로 확률밀도함수 $p_k(x_i|\theta_k)$를 다변량 정규분포로 가정하여 식 (34) $p_k(x_i|\theta_k)$를 식 (36)과 같이 나타내고 식 (40)을 최대화한다. μ_k와 $\sum_k$은 각각 모형의 평균벡터와 공분산행렬이다.

$$p_k(x_i|\theta_k) = p_k(x_i|\mu_k, \sum_k) = \frac{\exp\left\{-\frac{1}{2}(x_i-\mu_k)^T\sum_k^{-1}(x_i-\mu_k)\right\}}{(2\pi)^{\frac{M}{2}}\sqrt{|\sum_k|}} \tag{36}$$

식 (41)의 의미는, 어떤 확률밀도함수 식 (35)으로부터 객체가 생성될 때, 각 객체는 해당 군집의 평균과 분산에 따라 정규분포로 생성된다는 의미이다. 그러면 그림 3.1의 (A)를 다음과 같이 쓸 수 있다.

$$\widehat{\mu_k} \leftarrow \frac{\sum_{i=1}^{N}\widehat{c_{ik}}x_i}{\sum_{i=1}^{N}\widehat{c_{ik}}} \tag{37}$$

$$\widehat{\sum_k} : \text{모형 의존도} \tag{38}$$

이 때, 각 가우스 함수(Gaussian component)의 공분산 행렬이 $\sum_k = \sigma_k^2 I$(I는 '$M\times M$' 행렬, M은 객체에 대한 속성의 수라고 가정하면 확률밀도함수 식 (34)는 식 (39)와 같이 쓸 수 있고 식 (37), 식 (38)은 식 (40), 식(41)과 같이 쓸 수 있다.

$$p_k(x_i|\mu_k, \sigma_k) = \frac{\exp\left\{\frac{-\|x_i-\mu_k\|^2}{2\sigma_j^2}\right\}}{(2\pi\sigma_j^2)^{\frac{M}{2}}} \tag{39}$$

$$\widehat{\mu_k} \leftarrow \frac{\sum_{i=1}^{N}\widehat{c_{ik}}x_i}{\sum_{i=1}^{N}\widehat{c_{ik}}} \tag{40}$$

$$\widehat{\sigma_k} \leftarrow \frac{\sum_{i=1}^{N} \widehat{c_{ik}} \parallel x_i - \widehat{\mu_k} \parallel}{\sum_{i=1}^{N} M \bullet \widehat{c_{ik}}} \tag{41}$$

3.3. Fast Search

Fast search 군집방법은 Rodriguez[12]등에 의해 제안되었다. [12]는 DBSCAN과 다르게 군집의 중심객체에 초점을 두고 있는데, 높은 밀도를 가진 군집 중심객체와 그 군집 중심객체와의 거리가 멀리 위치해 있는 객체와의 관계를 계산한다. 밀도 ρ는 다음 식으로 산정한다.

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{42}$$

$$\chi(x) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & otherwise \end{cases}$$

i : 객체 i

ρ_i : 지역 밀도

d_{ij} : 객체 i와 객체 j와의 거리

d_c : 임계 거리

기본적으로, ρ_i 값은 객체 i에 대한 d_c값에 가까이 있는 객체의 수와 동일하다. 또한, 이 군집 분석 방법은 큰 데이터셋 일수록 다른 객체간의 ρ_i값의 관계에 민감하게 반응하며, d_c값의 선택에 관해서는 연구가 진행중이다.

거리 δ는 다음 식으로 산정한다.

$$\delta_i = \min_{j:\rho_j > \rho_i}(d_{ij}) \tag{43}$$

δ_i : 높은 밀도에 속한 객체로 부터의 거리

δ_i는 객체 i와 높은 밀도의 또 다른 객체(객체 j)간의 최소거리를 계산한다.

만약, 객체 i가 객체 j보다 높은 밀도를 가지고 있다면,

$$\delta_i = \max_j(d_{ij}) \tag{44}$$

위의 식으로 계산한다.

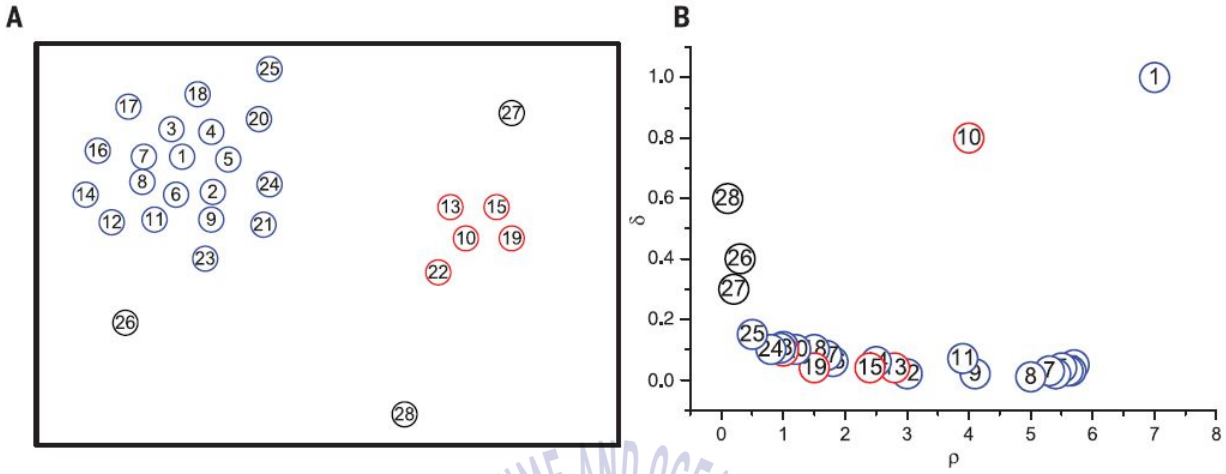


그림 3.2 Fast search[12]

위 그림의 A는 객체들의 분포도를 나타낸 것이며, B는 fast search의 의사결정 그래프이다. B를 보면 10과 11, 9번의 객체가 동일한 ρ값을 가지고 있으나, δ에는 큰 차이가 있음을 알 수 있다. 그러므로 ρ값, δ값 둘 다 높은 값을 가지고 있어야 군집의 중심점이 된다.

3.3.1 Fast Search의 문제점[23]

Fast search[12]는 군집의 중심점들을 찾는데 매우 빠르다. 그러나 이 정확도가 d_c 값에 의존한다는 것을 알 수가 있다. 또한 d_c 값을 선택할 수 있는 구체적인 방안이 나와 있지 않다.

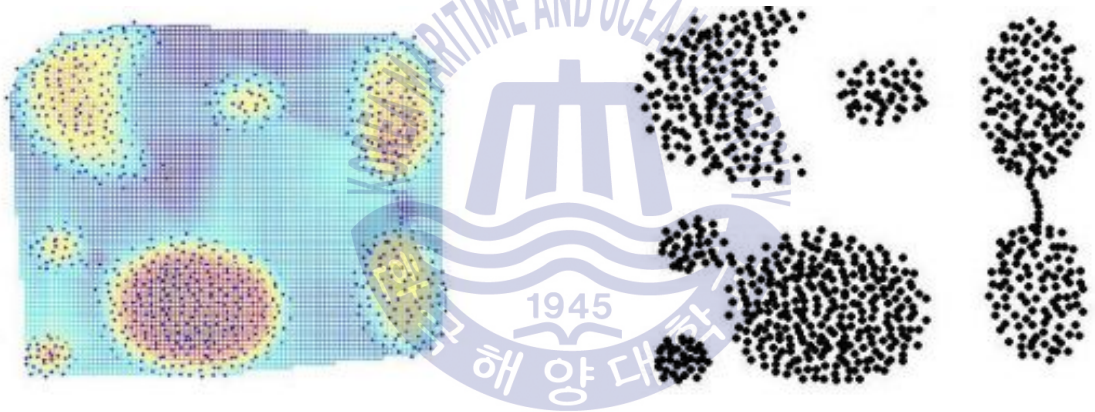
최근에 Shuliang Wang[23]등에 의해 d_c값 산출에 대한 새로운 방안이 제시되었다.

먼저, 가우스 함수를 이용하여 밀도를 산출한다.

$$\phi(x) = \sum_{i=1}^{n} \left(e^{-\left(\frac{\| x - x_i \|}{\sigma} \right)^2} \right) \tag{45}$$

$$\{x_1, x_2, \ldots, x_n\} \in D$$

식 (45)은 fast search[12]에서 밀도를 계산하는 식과 매우 유사하다는 것을 알 수 있다. 그림 3.3의 (a)는 데이터 필드의 잠재성의 분포를 보여 주는 것이며, 대해서는 검은색 지역은 큰 잠재성을 가진다고 본다. (b)는 원래 데이터셋의 밀도 분포를 나타낸 것이다. 그림 3.3의 (a)와 (b)를 보면 유사하다는 것을 알 수 있으므로, 데이터 필드의 잠재성과 데이터셋의 밀도는 유사한 효과를 가진다고 볼 수 있다. 그러므로 임계 거리인 d_c값을 데이터 필드의 영향요소(impact factor) σ의 최적화하는 것과 유사하다고 볼 수 있다.



(a) (b)

그림 3.3 데이터 필드의 잠재성 분포[23]

또한 불확실한 데이터에서는 주로 엔트로피(entropy)로 나타낼 수 있기 때문에, 엔트로피를 이용하여 영향요소(impact factor) σ의 최적값을 구한다.

$$H = -\sum_{i=1}^{n} \frac{\phi_i}{Z} log\left(\frac{\phi_i}{Z} \right) \tag{46}$$

$$\{x_1, x_2, \ldots, x_n\} \in D$$

모든 객체의 잠재성 : $\{\phi_1, \phi_2, \ldots, \phi_n\}$

정규화 인자 : $Z = \sum_{i=1}^{n} \phi_i$

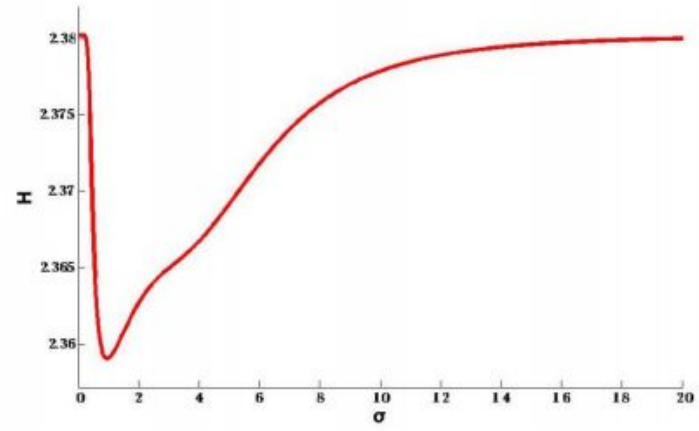


그림 3.4 σ값에 대한 엔트로피의 변화[23]

그림 3.4를 보면 σ는 0에서 ∞까지 나아가고 있으며, 먼저 엔트로피(entropy)의 값이 급격히 감소가 시작하는 곳과 나중에 천천히 증가하는 부분과 같은 레벨인 것을 알 수 있다. 또한, 엔트로피가 가장 작은 곳의 값이 σ=0.9531 임을 알 수 있다.

그림 3.3에서 언급하였듯이 데이터 필드의 분포는 객체들의 분포를 잘 반영함으로, 엔트로피가 가장 낮을 때 σ값을 선정해야한다.

그러나, 가우시안 분포의 3B rule(3B rule of Gaussian distribution)[24]에서는 모든 객체들의 영향 반경(influence radius)을 $\frac{3}{\sqrt{2}}\sigma$로 선정한다.

제4장 전산 실험 결과

4.1 Iris 데이터 분석 결과

본 논문의 서론에서 언급하였듯이, 여러 군집 방법들을 다양한 데이터셋을 이용하여 성능을 비교하고자 한다. SVM 알고리즘은 군집의 사전 정보가 알려져 있는 상태(supervised learning)에서 특성을 파악하기 때문에 군집분석과 큰 차이점을 보인다. 현재 군집의 정보가 알려져 있는 데이터셋(supervised learning dataset)을 사용하여 성능 비교를 하였기 때문에, SVM 알고리즘 또한 포함시켰다. UC Irvine Machine Learning Repository의 대표적인 데이터셋인 Iris로 각각의 군집방법들의 결과는 다음과 같다.

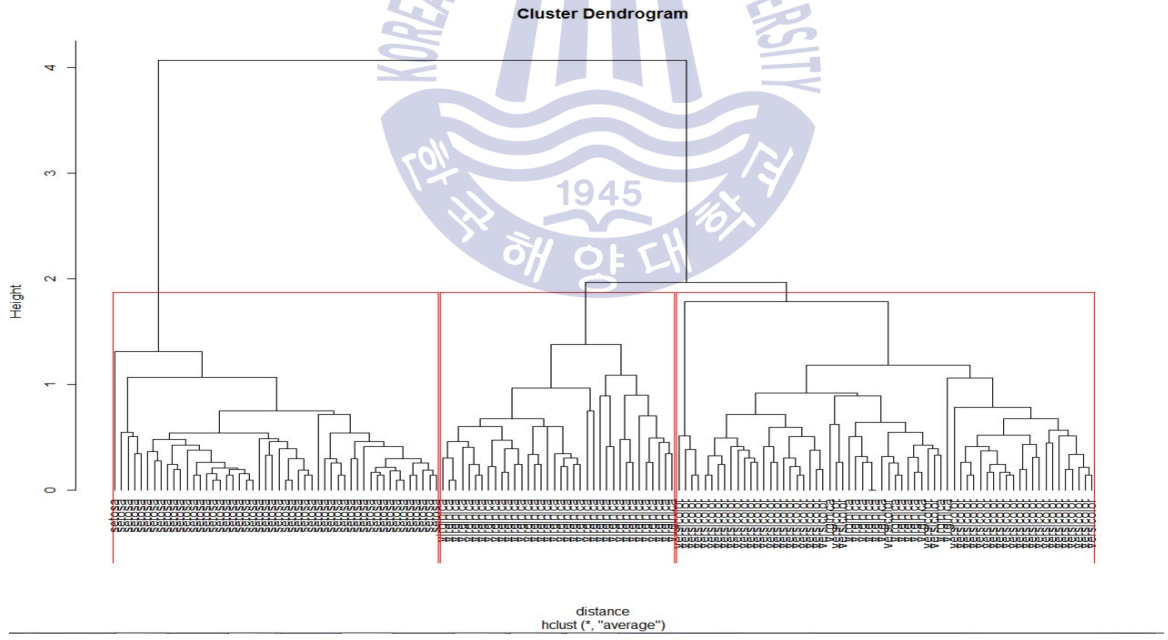


그림 4.1 iris 데이터셋을 계층적 군집방법(평균 연결법)으로 분석한 덴드로그램

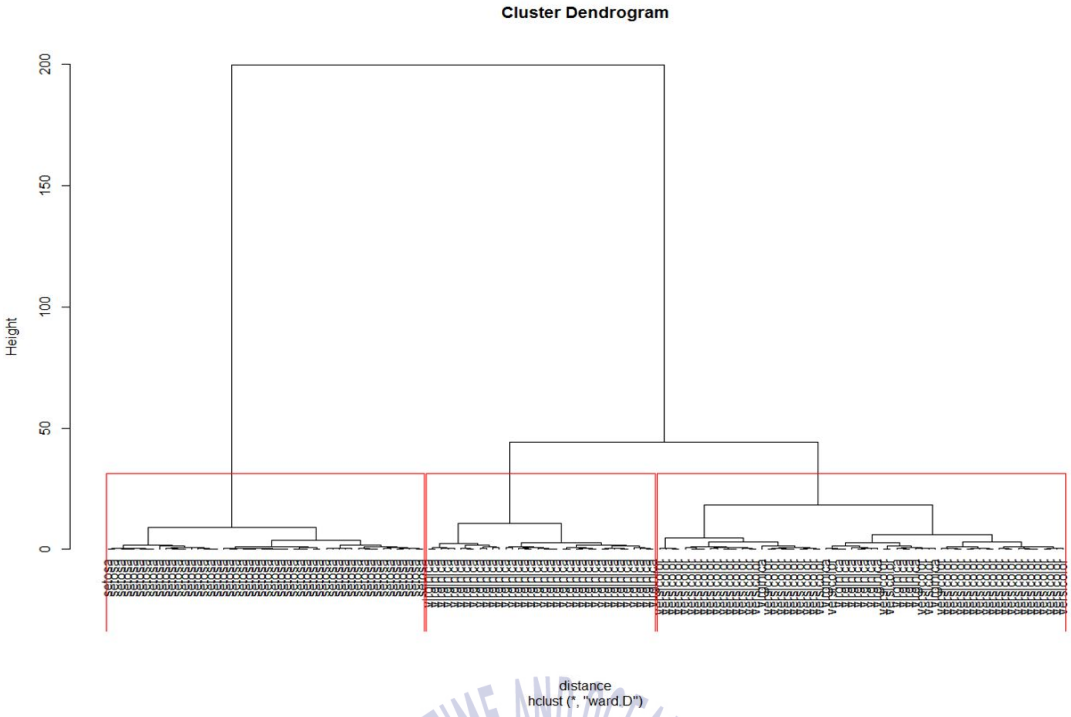


그림 4.2 iris 데이터셋을 워드 방법으로 분석한 덴드로그램

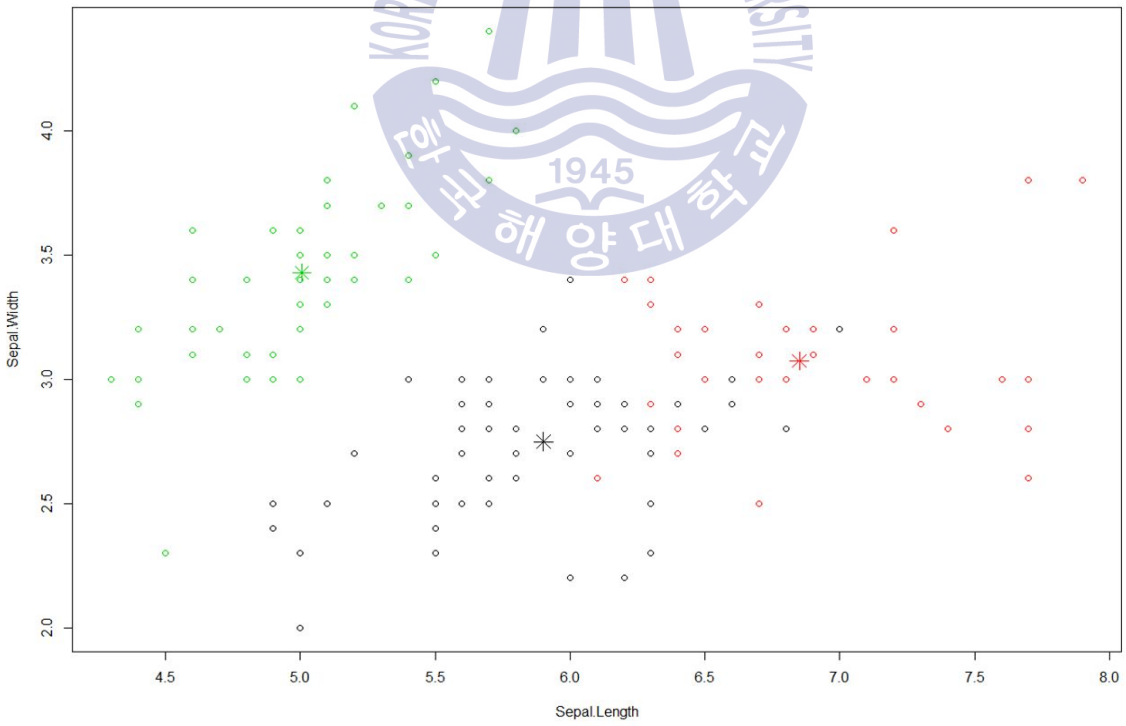


그림 4.3 iris 데이터셋을 K-means 군집방법으로 분석한 결과

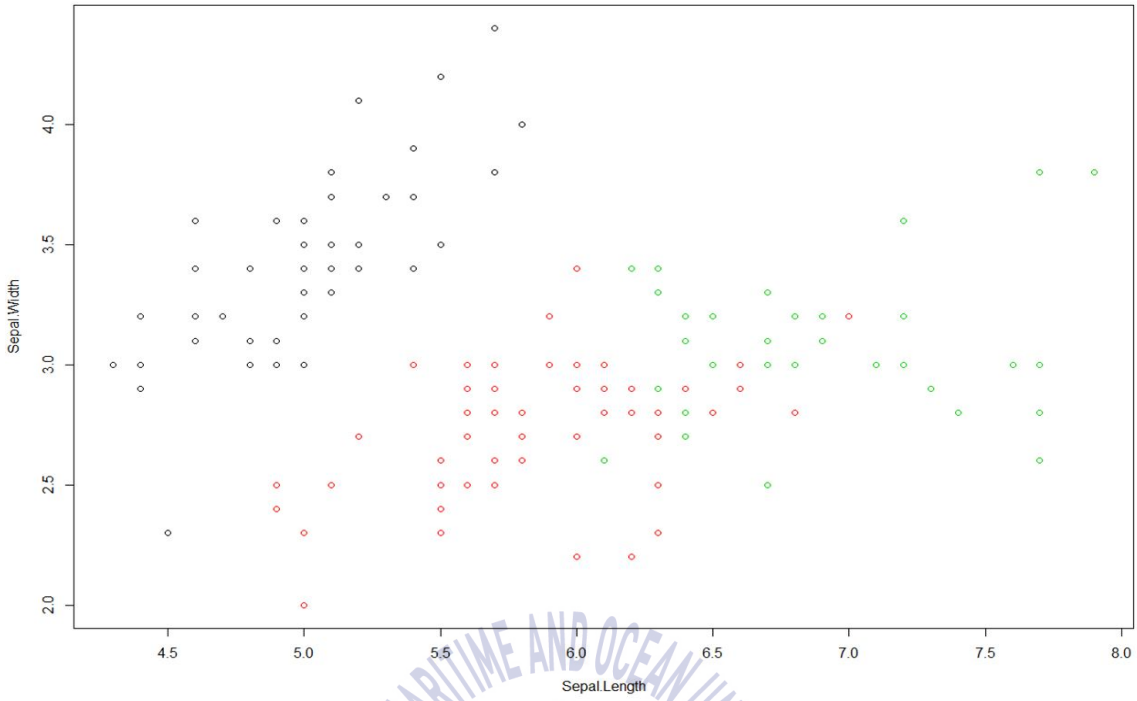


그림 4.4 iris 데이터셋을 K-medoids(PAM algorithm)군집방법으로 분석한 결과

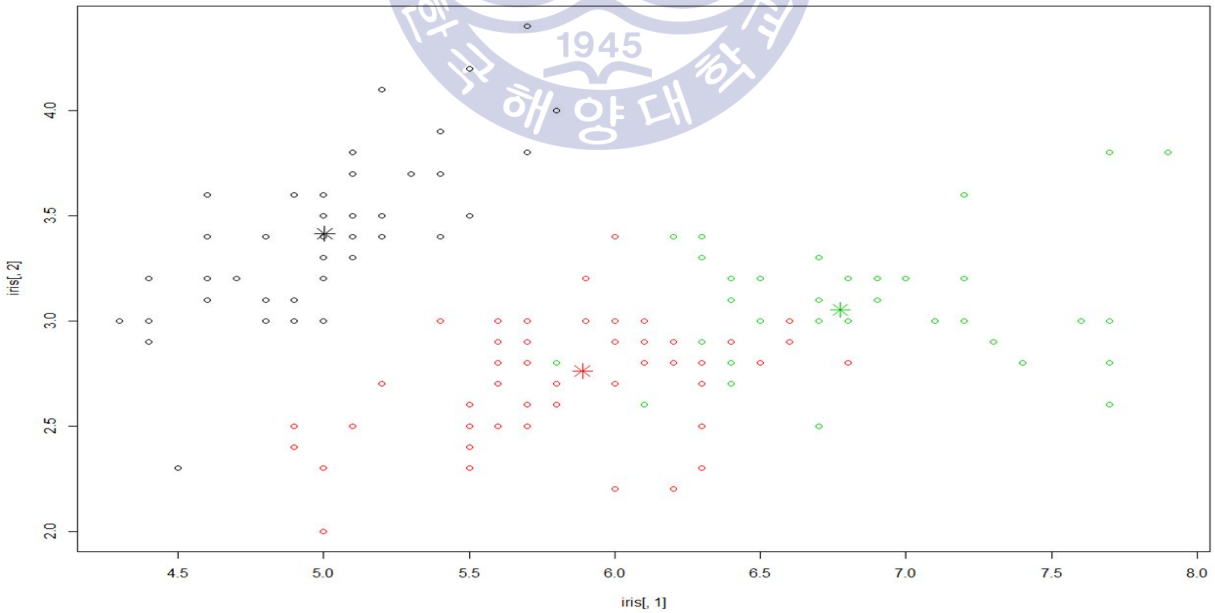


그림 4.5 iris 데이터셋을 퍼지 K-means 군집방법으로 분석한 결과

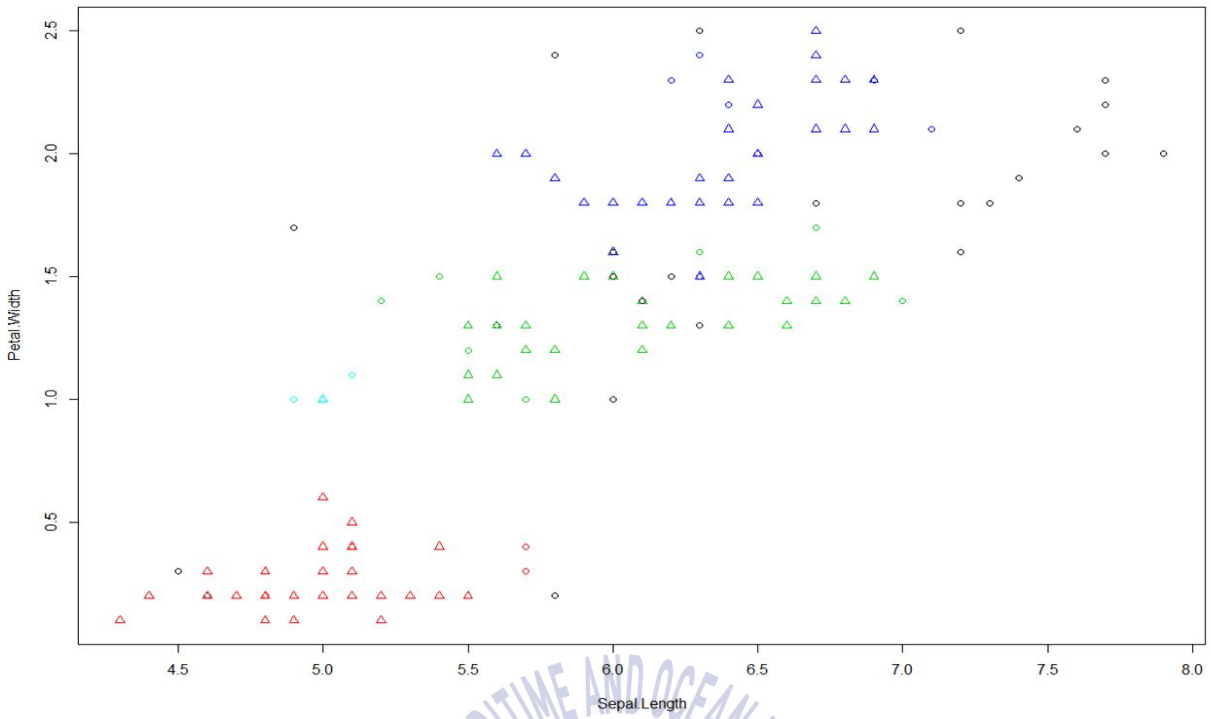


그림 4.6 iris 데이터셋을 DBSCAN으로 분석한 결과

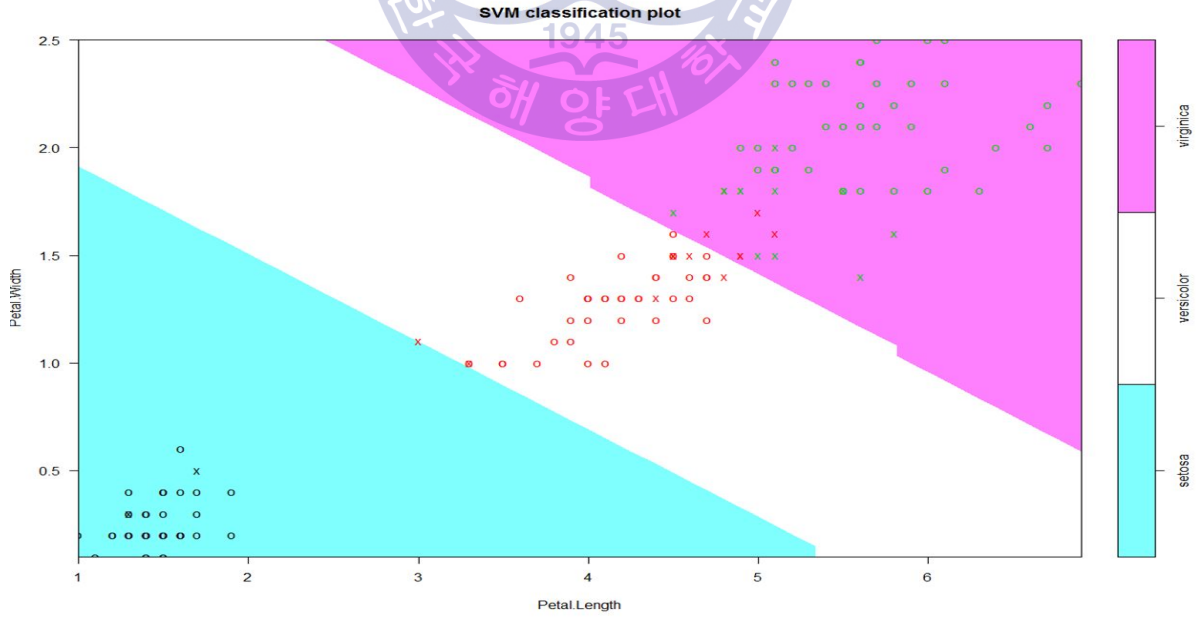


그림 4.7 iris 데이터셋을 SVM으로 분석한 결과

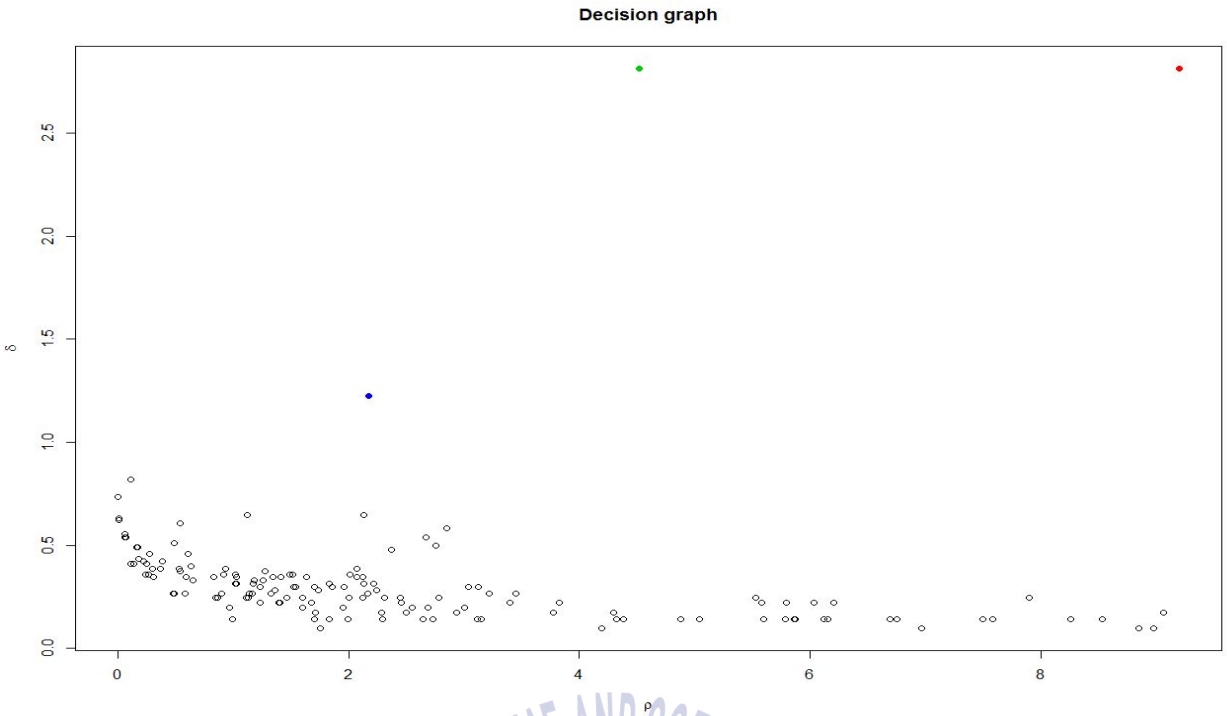


그림 4.8 iris 데이터셋을 fast search[1]의 의사 결정 그래프로 나타낸 결과

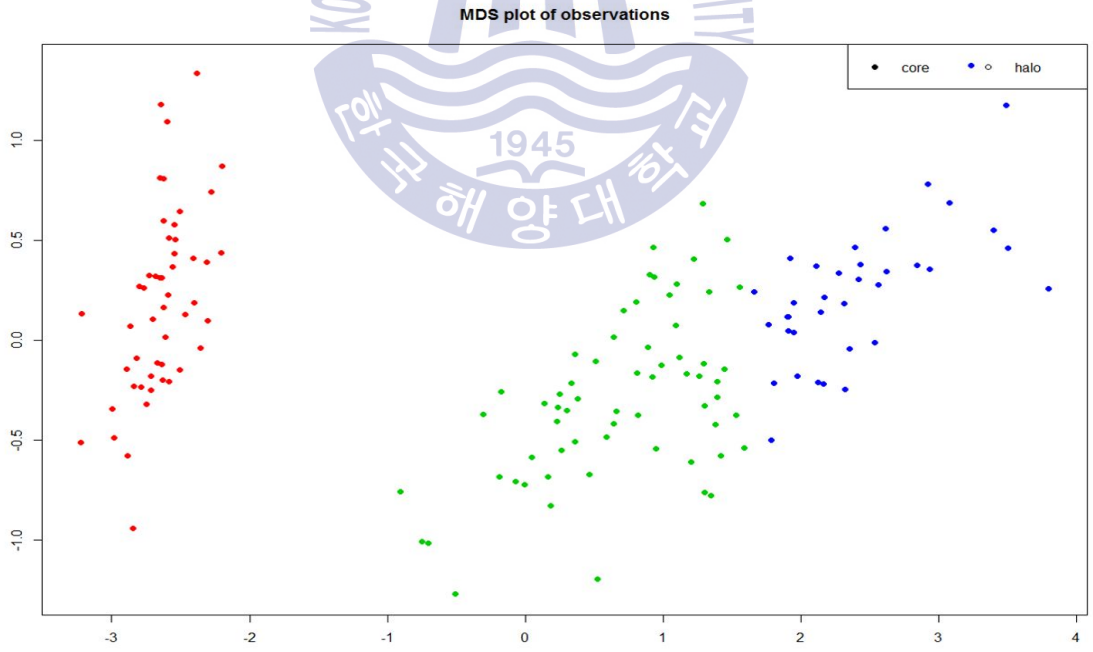


그림 4.9 iris 데이터셋을 fast search으로 분석한 결과

표 4.1 Iris 데이터셋에 대한 군집 분석 방법의 정확도와 계산속도

	HA	WM	KM	KMP	FKM	DB	EM	FS	SVM
정확도	90.66	90.66	89.33	89.33	89.33	78.66	96.66	90.66	96.66
속도	0.40	0.32	0.14	0.18	0.12	0.12	0.32	0.08	0.41

< 군집방법 >

HA : Hierarchical Algorithm

WM : Ward's method

KM : K-means Algorithm

KMP : K-medoids (PAM Algorithm)

FKM : Fuzzy K-means(C-means) Algorithm

DB : DBSCAN

EM : Multi-Gaussian with Expectation-Maximization Algorithm

FS : Fast search

표 4.1를 보면 96.66(%)로 EM 군집방법이 제일 정확도가 높게 나왔으나 속도에서는 0.32(sec)로 느리게 나왔다. 그리고 정확도가 90.66(%)로 fast search와 계층적 군집방법, 워드 방법이 두 번째로 높게 나왔다. 속도면에서는 fast search가 0.08(sec)로 가장 빠른 것을 관측하였고, 군집수를 정하지 않는 단점이 있는 계층적 군집방법과 워드방법이 속도가 매우 느린 것을 알 수 있었다.

4.2 UC Irvine의 데이터 분석 결과

군집방법의 비교를 위한 다양한 UC Irvine 데이터셋의 약어는 다음과 같다.

< 데이터셋 >[13]

CW : Chickweight dataset

LD : Liver disorders dataset

TF : Blood trasfusion service center dataset

SD : Seeds dataset

LC : Lung cancer dataset

HM : Haberman's survival dataset

MM : Mammographic mass dataset

PM : Pima indians diabetes dataset

표 4.2 여러 군집 분석 방법의 정확도

(%)	HA	WM	KM	KMP	FKM	DB	EM	FS	SVM
Iris	90.66	90.66	89.33	89.33	89.33	78.66	96.66	90.66	96.66
CW	40.00	37.72	32.35	29.76	31.14	36.16	25.78	39.79	100
LD	55.65	55.65	55.36	53.04	52.46	57.68	51.59	42.90	71.59
TF	76.74	57.35	73.93	71.12	70.72	76.87	57.89	76.34	76.20
SD	95.24	96.66	89.52	89.05	89.52	56.19	85.24	85.52	95.24
LC	55.55	70.37	59.26	55.55	70.37	44.44	44.44	29.63	100
HM	73.53	55.88	51.96	54.25	50.98	73.53	66.66	73.86	73.20
MM	67.59	67.59	68.55	68.19	68.55	51.45	59.88	66.02	82.77
PM	64.97	60.94	66.02	59.90	65.89	64.45	52.73	65.89	77.34

표 4.3 여러 군집 분석 방법의 계산속도

(sec)	HA	WM	KM	KMP	FKM	DB	EM	FS	SVM
Iris	0.40	0.32	0.14	0.18	0.12	0.12	0.32	0.08	0.41
CW	0.49	0.43	0.17	0.33	0.16	0.36	1.00	0.61	0.51
LD	0.21	0.20	0.16	0.19	0.04	0.19	0.53	0.17	0.14
TF	0.68	0.67	0.29	0.43	0.15	0.62	21.16	1.16	0.38
SD	0.50	0.44	0.26	0.34	0.11	0.33	4.34	0.40	0.24
LC	0.30	0.28	0.22	0.28	0.21	0.24	0.32	0.24	0.27
HM	0.50	0.49	0.28	0.34	0.12	0.35	0.38	0.39	0.27
MM	2.96	2.74	2.60	2.95	2.45	2.78	4.32	3.99	2.67
PM	0.65	0.64	0.30	0.46	0.13	0.57	1.16	1.68	0.37

표 4.4 여러 군집 분석 방법의 평균 정확도, 계산속도

	HA	WM	KM	KMP	FKM	DB	EM	FS	SVM
정확도	68.88	65.87	65.14	63.35	65.42	59.94	60.10	63.40	85.89
속도	0.34	0.34	0.23	0.30	0.19	0.30	1.95	0.29	0.38

표 4.2에서 보면 계층적 군집분석은 정확도와 비계층적 군집분석 정확도는 유사하게 나왔다. 이 중에서는 SVM을 제외하고 계층적 군집방법이 비교적 정확도가 높게 나왔고, 그다음은 K-means, K-medoids 군집방법과 fast Search가 높은 것을 알 수 있다.

그러나, fast search와 같은 경우는 d_c값을 어떻게 주느냐에 따라 정확도의 차이가 크다(표 4.5 참조). DBSCAN의 경우도 ε과 $MinPts$값을 어떻게 주느냐에 따라 군집의 개수와 형태의 차이점이 현저하게 달라진다는 것을 알 수 있다(그림 4.10, 4.11 참조).

계층적 군집방법은 정확도는 비교적 높으나 이상치를 제외시키지 않는 단점이 있다. 이를 보완하기 위해 본 논문에서는 계층적 군집방법에 대해 이상치를 제거하여 비교하였다.

표 4.5 d_c값에 따른 fast search 결과

	Iris		CW	
d_c	0.2596	0.28	6.3336	7.03
정확도	79.33	90.66	39.10	39.79

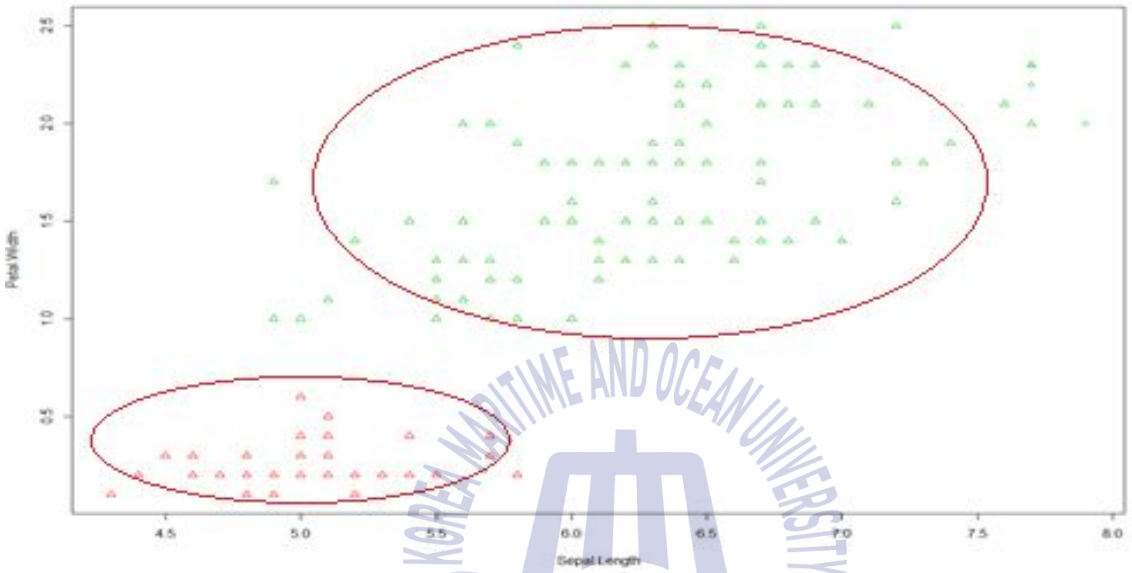


그림 4.10 iris 데이터셋을 DBSCAN으로 분석한 결과(ε=0.9, $MinPts$=6)

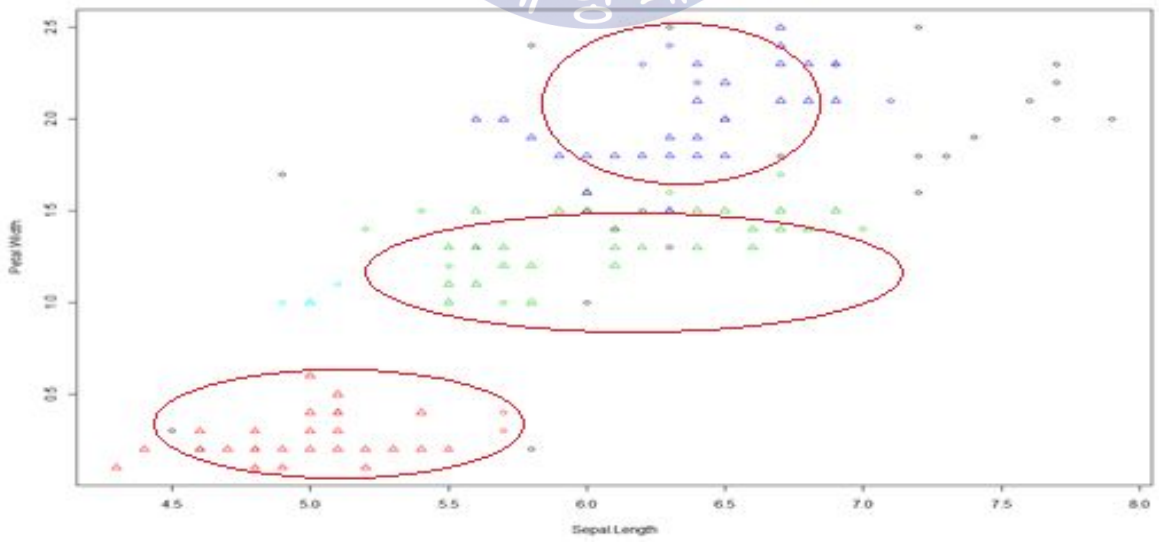


그림 4.11 iris 데이터셋을 DBSCAN으로 분석한 결과(ε=0.4, $MinPts$=4)

제5장 결론 및 추후 연구

5.1 결론

본 논문에서는 사이언스 저널에 발표된 fast search와 계층적, 비계층적 군집방법과 성능비교를 하였다. 계층적 군집방법에서는 연결법과 워드 방법, 비계층적 군집방법은 K-means 군집방법, K-medoids 군집방법, 퍼지 K-means 군집방법, 그리고 DBSCAN, EM 군집방법이 있다. 표 4.1과 같이 정확도에서는 DBSCAN이 제일 낮고, 계층적 군집방법이 제일 높게 관측되었다. Fast search는 d_c값에 따라서 정확도에 차이를 보이며, DBSCAN도 ε과 $MinPts$값에 따라 정확도에 큰 차이를 보이는 것을 알 수 있었다.

Fast search는 군집의 중심점을 구하여 분류함으로써, 속도에서는 빠르다는 것을 알 수 있었다. 계층적 군집방법은 덴드로그램을 이용하여 주관적으로 군집수를 정해야 된다는 단점이 있다. 비계층적 군집방법은 미리 군집수를 정해야 하기 때문에 초기해와 초기 군집수를 정해야 한다는 단점이 있다.

5.2. 추후 연구

본 논문의 비교결과, fast search의 d_c값을 주관적으로 조정하여 객체의 군집을 분류하는 단점이 있다. 그러므로 fast search의 d_c값의 추정에 대한 Wang등[23]과 같은 추후 연구가 필요하다. 또한 결론에서 언급했듯이 각 군집분석방법의 단점을 보완한 효율적인 군집 알고리즘의 개발이 시급하고, 의학 및 IT분야에의 응용에 관한 추후 연구도 진행되어야 한다.

참고 문헌

[1] Bolon.C. 2014, “A review of microarray datasets and applied feature selection methods”, Information Sciences vol. 282, pp. 111-135.

[2] Kaufman.L., Rousseeuw.P.J. 1990, “Finding Groups in Data : An Introd duction to Cluster Analysis”, Wiley, New York.

[3] Sorensen.T. 1948, “A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons“, Biologiske Skrifter.

[4] Robert.R.S., Charles.D.M. 1958, “A Statistical Method for Evaluating Sys tematic Relationships“, Univ Kans Sci Bull vol. 38, pp. 1409-1438.

[5] Ward.J.H. 1963, “Hierarchical Grouping to Optimize an Objective Funct ion”, Journal of the American Statistical Association vol. 58, pp. 236-244.

[6] MacQueen.J. 1967, “Some methods for classification and analysis of m ultivariate observations”, Proceedings of the 5th Berkeley Symposium o n Mathematical Statistics and Probability vol. 1, pp. 281-297.

[7] Vinod.H. 1969, “Integer programming and the theory of grouping”, Jo urnal of the American Statistical Association vol. 64, pp. 506-517.

[8] Martin.E., Hans-Peter.K., Jörg.S., Xiaowei.X. 1980, “A Density-Based Al gorithm for Discovering Clusters in Large Spatial Databases with Noise ”, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining vol. 96, pp. 226-231.

[9] Henrik.B., Anders.H., Niklas.N. 2011, “DBSCAN A Density-Based Spatial Clustering of Application with Noise”, Linköpings Unive rsitet.

[10] Christopher.M.B. 2006, “Mixture models and the EM algorithm”, Micro soft Research, Cambridge.

[11] Joshua.H.P. 2013, “The EM Algorithm in Multivariate Gaussian Mixture Models using Anderson Acceleration”, Worcester Polytechnic Institute.

[12] Alex.R., Alessandro.L. 2014, “Clustering by fast search and find of de nsity peaks”, Science vol. 344, pp. 1492-1496.

[13] UC Irvine Machine Learning Repository <http://archive.ics.uci.edu/ml/>.

[14] Burges.C.J.C. 1998, “A tutorial on support vector machines for patter n recognition”, Data Mining and Knowledge Discovery vol. 2 pp. 121 -167.

[15] Vapnik.V. 1995, “The Nature of Statistical Learning Theory”, Springe r, NewYork.

[16] Vapnik.V. 1998, “Statistical Learning Theory”, Wiley. NewYork.

[17] 전치혁. 2012, “데이터마이닝 기법과 응용”, 한나래출판사.

[18] Ng.R.T., Han.J. 1994, “Efficient and effective clustering methods for spatial data mining”, Proceedings of the 20th VLDB Conference, Santi ago, Chile, pp. 144-155.

[19] Park.H.S., Jun.C.H. 2009, “A simple and fast algorithm for K-medoids clustering”, Expert Systems With Applications vol. 36, pp. 3336-3341.

[20] 윤애란. 2004, “DBSCAN 알고리즘을 이용한 유전자 발현 데이터 마이닝 시스템의 설계 및 구현”, 이화여자대학교 과학기술대학원 석사학위 청구논문.

[21] 김완섭. 2009, “대용량 데이터를 처리하기 위한 EM Survey”.

[22] 신형주. 2001, “Latent Variable을 이용한 확률적 클러스터링 모델 (A Probailistic clustering model with Latent Variables)”, 서울대학교 대학원 석사학위청구논문.

[23] Shuliang.W., Dakui.W., Caoyuan.L., Yan.L. 2015, “ Comment on Clustering by fast search and find of density peaks”

[24] Bárány.I., Vu.V. 2007, “Central limit theorems for Gaussian polytopes”, Annals of Probability(Institute of Mathematical Statistics) vol. 35, pp. 1593-1621.

[25] Alex.R., Alessandro.L. 2014, “Supplementary Materials for Clustering by fast search and find of density peaks”, Science vol. 344, pp. 1492-1496.

[26] Mohamed.N.A., Sameh M.Y., Nevin.M., Aly.A.F., Thomas.M. 2002, “A Modified Fuzzy C-Means Algorithm for Bias Field Estimation and Segmentation of MRI Data”, IEEE Transactions on medical imaging vol. 21 pp. 193-199.

[27] James.C.B. 1984, “FCM: The Fuzzy c-means clustering algorithm”, Computers & Geosciences vol. 10, pp. 191-203.

[28] Lee.G., Clayton.S. 2010, “EM algorithms for multivariate Gaussian mixture models with truncated and censored data”, Computational Statistics & Data Analysis vol. 56, pp. 2816-2829.

[29] “The EM Algorithm for Gaussian Mixture”, Probabilistic Learning: Theory and Algorithms.

[30] 박연복., 이규민., 강상진. 2011, “군집분석을 이용한 수준설정 방법과 타당성 연구”, Journal of Educational Evaluation 2011 vol. 24, pp. 645-664.

부 록

< Iris >

```
# Hierarchical Clustering
library(fpc)
distance <- dist(iris[,-5], method="euclidean")
cluster <- hclust(distance, method="average")
plot(cluster, hang=-1, label=iris$Species)
r<-rect.hclust(cluster , k=3, border="red")
a<-vector(length=150)
a[r[[1]]]<-1
a[r[[2]]]<-2
a[r[[3]]]<-3
table(a,iris$Species)

# Ward Hierarchical Clustering
library(fpc)
distance <- dist(iris[,-5], method="euclidean")
cluster <- hclust(distance, method="ward.D")
plot(cluster, hang=-1, label=iris$Species)
r<-rect.hclust(cluster , k=3, border="red")
a<-vector(length=150)
a[r[[1]]]<-1
a[r[[2]]]<-2
a[r[[3]]]<-3
table(a,iris$Species)

# K-means Algorithm
library(e1071)
newiris <- iris
newiris$Species <- NULL
(kc <- kmeans(newiris, 3))
table(iris$Species, kc$cluster)
plot(newiris[c("Sepal.Length","Sepal.Width")], col=kc$cluster)
points(kc$centers[,c("Sepal.Length","Sepal.Width")], col=1:3, pch=8, cex=2)

# K-medoids PAM Algorithm
library(cluster)
newiris <- iris
newiris$Species <- NULL
(result <- pam(newiris [1:4],3,FALSE,"euclidean"))
summary(result)
table(iris$Species,result$cluster)
plot(newiris[c("Sepal.Length","Sepal.Width")], col=result$cluster)
points(result$centers[,c("Sepal.Length",
```

```
"Sepal.Width")], col=1:3, pch=8,cex=2)

# Fuzzy C-Means Algorithm
library(e1071)
result <- cmeans(iris[,-5], 3, 100,m=2, method="cmeans")
plot(iris[,1], iris[,2], col=result$cluster)
points(result$centers[,c(1,2)], col=1:3, pc h=8, cex=2)
result$membership[1:3,]
table(iris$Species, result$cluster)

# Density-based Cluster
library(fpc)
cluster <- dbscan(iris [,-5], eps=0.9,Mi nPts=6)
plot(cluster, iris[,c(1,4)])
table(cluster$cluster, iris $Species)

# Multi-Gaussian with Expectation-M aximization
library(mclust)
mc <- Mclust(iris[,1:4], 3)
plot(mc, data=iris[,1:4], what=c('classif ication'),dimens=c(3,4))
table(iris$Species, mc$classification)

# Clustering by fast search
library(densityClust)
irisDist <- dist(iris[,1:4])
irisClust <- densityClust(irisDist, gauss ian=TRUE)
plot(irisClust)
irisClust <- findClusters(irisClust, rho= 1,delta=1)
plotMDS(irisClust)
table(iris$Species,irisClust $cluster)

# Support vector machine
data(iris)
attach(iris)
N<-nrow(iris)
y<-iris[,5]
m2<-svm(Species~.,data=iris,kernel="lin ear")
summary(m2)
pred<-predict(m2,iris)
table(pred,y)

< Lung cancer >

# hierarchical clustering
library(fpc)
lungcancer<-read.csv("C:/Users/Yang/ Desktop/lungcancer.csv")
d<-NULL
for(i in 1:nrow(lungcancer)){
        if(any(lungcancer[i,]=="?"))next
```

```
        d<-rbind(d,lungcancer[i,])
}
for(i in 1:ncol(d)) d[,i]<-as.integer(d[,i])
d$y<-as.factor(d$y)
nrow(d)
distance <- dist(d,method="euclidean")
cluster<-hclust(distance, method="average")
plot(cluster, hang=-1, label=d$y)
r<-rect.hclust(cluster,k=3, border="red")
a<-vector(length=27)
a[r[[1]]]<-1
a[r[[2]]]<-2
a[r[[3]]]<-3
table(a,d$y)
```

```
# ward's method
library(fpc)
lungcancer<-read.csv("C:/Users/Yang/Desktop/lungcancer.csv")
d<-NULL
for(i in 1:nrow(lungcancer)){
        if(any(lungcancer[i,]=="?"))next
        d<-rbind(d,lungcancer[i,])
}
for(i in 1:ncol(d)) d[,i]<-as.integer(d[,i])
d$y<-as.factor(d$y)
nrow(d)
distance<- dist(d, method="euclidean")
cluster<-hclust(distance, method="ward.D")
plot(cluster, hang=-1, label=d$y)
r<-rect.hclust(cluster,k=3, border="red")
a<-vector(length=27)
a[r[[1]]]<-1
a[r[[2]]]<-2
a[r[[3]]]<-3
table(a,d$y)
```

```
# K-means algorithm
library(e1071)
lungcancer<-read.csv("C:/Users/Yang/Desktop/lungcancer.csv")
d<-NULL
for(i in 1:nrow(lungcancer)){
        if(any(lungcancer[i,]=="?"))next
        d<-rbind(d,lungcancer[i,])
}
for(i in 1:ncol(d)) d[,i]<-as.integer(d[,i])
d$y<-as.factor(d$y)
lungcancer<-d
nrow(d)
```

```
newlungcancer <- lungcancer
newlungcancer$y <- NULL
(kc <- kmeans(newlungcancer , 3))
table(lungcancer$y,kc$cluster)

# K-medoids algorithm
library(cluster)
lungcancer<-read.csv("C:/Users/Yang/
Desktop/lungcancer.csv")
d<-NULL
for(i in 1:nrow(lungcancer)){
        if(any(lungcancer[i,]=="?"))next
        d<-rbind(d,lungcancer[i,])
}
for(i in 1:ncol(d)) d[,i]<-as.integer(d[,i])
d$y<-as.factor(d$y)
lungcancer<-d
nrow(d)
newlungcancer <- lungcancer
newlungcancer $y <- NULL
(result<-pam(newlungcancer [1:56],3,F
ALSE,"euclidean"))
table(lungcancer$y,result$cluster)
g<-result$cluster
h<-list(which(g==1),which(g==2),which(
g==3))
a<-vector(length=27)
a[h[[1]]]<-1
a[h[[2]]]<-2
a[h[[3]]]<-3
table(a,lungcancer$y)

# Fuzzy C-means
library(e1071)
lungcancer<-read.csv("C:/Users/Yang/
Desktop/lungcancer.csv")
d<-NULL
for(i in 1:nrow(lungcancer)){
        if(any(lungcancer[i,]=="?"))next
        d<-rbind(d,lungcancer[i,])
}
for(i in 1:ncol(d)) d[,i]<-as.integer(d[,i])
d$y<-as.factor(d$y)
lungcancer<-d
nrow(d)
result <- cmeans(lungcancer[,-57], 3,
100, m=2, method="cmeans")
table(lungcancer$y, result$cluster)

# DBSCAN
library(fpc)
lungcancer<-read.csv("C:/Users/Yang/
Desktop/lungcancer.csv")
d<-NULL
for(i in 1:nrow(lungcancer)){
        if(any(lungcancer[i,]=="?"))next
        d<-rbind(d,lungcancer[i,])
}
```

```
for(i in 1:ncol(d)) d[,i]<-as.integer(d[,i])
d$y<-as.factor(d$y)
lungcancer<-d
nrow(d)
newlungcancer <- lungcancer
newlungcancer $y <- NULL
cluster<-dbscan(lungcancer[,1:56],eps=
4.5,MinPts=4)
table(cluster$cluster, lungcancer$y)

# Multi-Gaussian with Expectation-M
aximization
library(mclust)
lungcancer<-read.csv("C:/Users/Yang/
Desktop/lungcancer.csv")
d<-NULL
for(i in 1:nrow(lungcancer)){
        if(any(lungcancer[i,]=="?"))next
        d<-rbind(d,lungcancer[i,])
}
for(i in 1:ncol(d)) d[,i]<-as.integer(d[,i])
d$y<-as.factor(d$y)
lungcancer<-d
nrow(d)
newlungcancer <- lungcancer
newlungcancer $y <- NULL
mc <- Mclust(newlungcancer[,1:56], 3)
table(lungcancer$y, mc$classification)

# fast search
library(densityClust)
lungcancer<-read.csv("C:/Users/Yang/
Desktop/lungcancer.csv")
d<-NULL
for(i in 1:nrow(lungcancer)){
        if(any(lungcancer[i,]=="?"))next
        d<-rbind(d,lungcancer[i,])
}
for(i in 1:ncol(d)) d[,i]<-as.integer(d[,i])
d$y<-as.factor(d$y)
lungcancer<-d
nrow(d)
lungcancerDist<-dist(lungcancer[,1:56])
lungcancerClust<-densityClust(lungcan
cerDist, gaussian=TRUE)
plot(lungcancerClust)
lungcancerClust<-findClusters(lungcanc
erClust,rho=4,delta=5.5)
table(lungcancer$y,lungcancerClust$clu
ster)
plotMDS(lungcancerClust)

# support vector machine
library(e1071)
lungcancer<-read.csv("C:/Users/Yang/
Desktop/lungcancer.csv")
d<-NULL
for(i in 1:nrow(lungcancer)){
```

```
        if(any(lungcancer[i,]==“?“))next
        d<-rbind(d,lungcancer[i,])
}
for(i in 1:ncol(d)) d[,i]<-as.integer(d[,i])
d$y<-as.factor(d$y)
lungcancer<-d
nrow(d)
y<-lungcancer[,57]
m2<-svm(y~.,data=lungcancer,kernel=“l
inear“)
pred<-predict(m2,lungcancer)
table(pred,y)
```

감사의 말

석사 과정을 마무리하며 지난 시간들을 돌이켜보니 아쉬움과 후회가 남습니다. 항상 주변에서 저에게 힘을 주시고 방향을 잡아주셨던 많은 분들께 감사의 말씀을 전하고자 합니다.

먼저 본 논문을 비롯하여 석사 과정 동안 연구에 매진할 수 있도록 세심한 지도와 많은 격려로 이끌어 주신 김재환 교수님께 진심으로 감사드립니다. 또한 논문 심사를 맡아주시며 충고와 조언을 해주셨던 박찬근 교수님, 김익성 교수님께 감사드리며, 장길웅 교수님, 배재국 교수님, 홍정희 교수님, 손미정 교수님께도 감사드립니다.

또한 같은 고민을 하며 고생한 대학원 식구들인 호연이와 정태, 세영, 동호, 재익이에게도 심심한 감사의 인사를 드립니다.

항상 부족한 자식을 믿어주시고 지원을 아끼지 않으신 부모님께 감사의 말씀을 드리며, 항상 좋은 생각을 가지고 힘이 되어주는 동생 예민이에게 대학원 생활의 결실인 이 논문을 바칩니다. 앞으로 이러한 가족들의 은혜에 조금이나마 보답하며, 장남으로써 큰 버팀목이 될 수 있도록 노력하겠습니다.

마지막으로 일일이 언급을 하지 못했지만 그 동안 저를 아끼고 사랑해주신 모든 분들게 다시 한번 진심으로 감사 드립니다.