博士學位論文

데이터 속성에 따른 초기 클러스터 결정 및 클러스터링

濟州大學校 大學院

電算統計學科

姜 亨 昌

2008年 2月

데이터 속성에 따른 초기 클러스터 결정 및 클러스터링

指導敎授 金 鐵 洙

姜 亨 昌

이 論文을 理學 博士學位 論文으로 提出함

2008年 2月

姜亨昌의 理學 博士學位 論文을 認准함

審 査 委 員 長 _____

副 委 員 長 _____

委 員 _____

委 員 _____

委 員 _____

제주대학교 대학원

2008年 2月

The initial cluster decision and clustering method for the data attributes

Hyung Chang Kang
(Supervised by professor Chul Soo Kim)

A thesis submitted in partial fulfillment of the requirement for the degree of Doctor of Philosophy

Department of Computer Science and Statistics
Graduate School
Cheju National University

February 2008

목 차

List of Figures ····· i
List of Tables ····· ii
Abstract ····· iv
I. 연구 배경과 목적 ····· 1
II. 클러스터링의 개념 ····· 6
1. 클러스터링의 배경 ····· 6
2. 클러스터링의 일반적 과정 ····· 8
3. 클러스터링을 위한 요구사항 ····· 11
III. K-means 기반 클러스터링 ····· 13
1. 수치 데이터에 대한 K-means 기반 알고리즘 ····· 13
1) K-means 알고리즘 ····· 13
2) K-medoids 알고리즘 ····· 15
3) K-means 알고리즘에 대한 초기 클러스터 결정 방법 ····· 19
2. 범주 데이터에 대한 K-means 기반 알고리즘 ····· 24
1) ROCK 알고리즘 ····· 25
2) K-modes 알고리즘 ····· 27
3. 혼합 데이터에 대한 K-means 기반 알고리즘 ····· 30
1) K-prototypes 알고리즘 ····· 30
2) Ahmad와 Dey의 제안 알고리즘 ····· 32
4. 클러스터 ensemble ····· 34
IV. 데이터 속성에 따른 클러스터링 ····· 35
1. 수치 데이터 클러스터링 문제 ····· 35
1) 계층적 클러스터링 문제 ····· 35

2) K-means 알고리즘 문제 ····· 35
2. 범주 데이터 클러스터링 문제 ····· 40
1) 도메인과 속성 ····· 40
2) K-modes 알고리즘 문제 ····· 40
3. 혼합 데이터 클러스터링 ····· 45
1) 혼합 데이터 클러스터 알고리즘 ····· 46
2) 초기 클러스터 결정을 위한 Modified K-means 알고리즘 ····· 46
3) 초기 클러스터 결정을 위한 K-priority 알고리즘 ····· 47
4) 초기 클러스터 결정을 위한 Modified K-modes 알고리즘 ····· 48
V. 실험결과 ····· 51
1. 실험환경 ····· 51
2. 실험 데이터 ····· 51
3. 실험결과 ····· 52
1) 평가방법 ····· 52
2) 혼합 데이터 클러스터링 실험결과 ····· 52
3) 수치 데이터 클러스터링 실험결과 ····· 53
4) 범주 데이터 클러스터링 실험결과 ····· 55
VI. 결론 ····· 59
VII. 참고문헌 ····· 62

List of Figures

Figure 1. K-means algorithm ····· 14
Figure 2. K-medodids algorithm ····· 15
Figure 3. Macqueen approach method ····· 20
Figure 4. Kaufman approach method ····· 21
Figure 5. Max-Min method ····· 22
Figure 6. ROCK algorithm ····· 26
Figure 7. K-modes algorithm ····· 29
Figure 8. K-prototypes algorithm ····· 32
Figure 9. Example of the similarity between object and cluster(true case) ····· 41
Figure 10. Example of the similarity between object and cluster(false case) ····· 41
Figure 11. Overview of CEBMDC algorithm framework ····· 46
Figure 12. Cluster Ensemble Based Mixed Data Clustering ····· 46
Figure 13. Modified K-means algorithm ····· 47
Figure 14. K-priority algorithm ····· 47
Figure 15. Max-Min Method for K-modes algorithm ····· 49
Figure 16. Modified K-modes algorithm ····· 49

List of Tables

Table 1. K-means algorithm for cluster update of the Iris data (all attribute) ····· 36
Table 2. K-means algorithm for cluster update of the Iris data (choice attribute) ····· 36
Table 3. K-means algorithm for cluster update of the Wine data (all attribute) ····· 37
Table 4. K-means algorithm for cluster update of the Wine data (choice attribute) ····· 38
Table 5. An instance of categorical example ····· 40
Table 6. Experimental data ····· 51
Table 7. Cluster recovery result for Heart disease data set with the proposed algorithm(accuracy) ····· 52
Table 8. Cluster recovery result for Iris data set with the proposed algorithm ····· 53
Table 9. Cluster recovery result for Iris data set with the K-means algorithm ····· 53
Table 10. Comparison of different clustering algorithms for Iris data set (best) ····· 54
Table 11. Comparison of different clustering algorithms for Iris data set (average) ····· 54
Table 12. Cluster recovery result for Wine data set with the proposed algorithm ····· 54
Table 13. Cluster recovery result for Wine data set with the K-means

algorithm ····· 55
Table 14. Comparison of different clustering algorithms for Wine data set(best) ····· 55
Table 15. Comparison of different clustering algorithms for Wine data set(average) ····· 55
Table 16. Cluster recovery result for Small Soybean data set with the proposed algorithm ····· 56
Table 17. Cluster recovery result for Small Soybean data set with the K-modes algorithm ····· 56
Table 18. Comparison of different clustering algorithm for Small Soybean data set(best) ····· 56
Table 19. Comparison of different clustering algorithm for Small Soybean data set(average) ····· 56
Table 20. Cluster recovery result for Mushroom data set with the proposed algorithm ····· 57
Table 21. Cluster recovery result for Mushroom data set with the K-modes algorithm ····· 57
Table 22. Comparison of different clustering algorithm for Mushroom data set(best) ····· 57
Table 23. Comparison of different clustering algorithm for Mushroom data set(average) ····· 58
Table 24. Comparison of different clustering algorithm for Mushroom data set(execution time) ····· 58

Abstract

Clustering typically groups data into sets in such a way that the intra cluster similarity is maximized while the inter cluster similarity is minimized.

Most previous clustering algorithms focus on numerical data whose inherent geometric properties can be exploited naturally to define distance functions between data points. However, much of the data existed in the databases is categorical, where attribute values cannot be naturally ordered as numerical values.

The K-means algorithm is best suited for implementing this operation of its efficiency in clustering large data sets. However, working only on numeric values limits its use in data mining because data sets in data mining often contain categorical values.

Pe˜na et al, compared empirically four initialization methods for the K-means algorithm: random, Forgy, Macqueen and Kaufman. Although this algorithm is known for its robustness, it is widely reported in literature that its performance depends upon two key points: initial clustering and instance.

Optimal determination of cluster size has an effect on the result of clustering. In K-means algorithm, the difference of clustering performance is large by initial k. But the initial cluster size is determined by prior knowledge or subjectivity in most clustering process.

This subjective determination may not be optimal. Due to the special properties of categorical attributes, the clustering of categorical data seems more complicated than that of numerical data.

The K-modes algorithm uses a simple matching dissimilarity measure to deal with categorical objects, replaces the means of clusters with modes, and uses a frequency based method to update modes in the clustering process to

minimize the clustering cost function.

The original K-means clustering algorithm is designed to work primarily on numeric data sets. This prohibits the algorithm from being directly applied to categorical data clustering in many data mining applications. However, as is the case with most data clustering algorithms, the algorithm requires a pre-setting or random selection of initial points (modes) of the clusters. The differences on the initial points often lead to considerable distinct cluster results.

The K-prototypes algorithm integrates the K-means and K-modes processes to cluster data with mixed numeric and categorical values. The method is developed to dynamically update the k's prototypes in order to maximize the intra cluster similarity of objects. When applied to numeric data the algorithm is identical to the K-means.

Ahmad and Dey proposed new cost function and distance measure based on co-occurrence of values. The measures also take into account the significance of an attribute towards the clustering process. In this algorithm presented a modified description of cluster center to overcome the numeric data only limitation of K-mean algorithm and provide a better characterization of clusters.

In this paper, we find another methods that work well for data with mixed numeric and categorical features. We propose a novel divide-and conquer technique to solve this problem.

Cluster ensemble is the method to combine several runs of different clustering algorithms to get a common partition of the original data sets, aiming for consolidation of results from a portfolio of individual clustering results.

First, the original mixed dataset is divided into two sub-data sets: the pure numeric dataset and the pure numeric data sets. Next, existing well established clustering algorithms designed for different types of data sets are

employed to produce corresponding clusters. Last, the clustering results on the numeric and categorical data sets are combined as a categorical data sets, on which the categorical data clustering algorithm is used to get the final clusters.

The goal of this paper is to provide an algorithm framework for the mixed attributes clustering problem, in which existing clustering algorithms can be easily integrated, the capabilities of different kinds of clustering algorithms and characteristics of different types of data sets could be fully exploited. Comparisons with other clustering algorithms on real life data sets illustrate the proposal approach.

I. 연구 배경과 목적

컴퓨터와 인터넷이 발전함에 따라 기업이나 모든 조직들은 정보 인프라로 데이터베이스를 구축하게 되었으며, 데이터베이스는 방대하게 커졌다. 방대한 데이터베이스로부터 새로운 지식을 얻고자 하는 과정을 KDD(Knowledge Discovery in Databases)라 하며 이러한 한 분야가 데이터 마이닝(data mining)이다.

클러스터링(clustering)은 데이터 마이닝의 한 방법으로 데이터를 구성하고 있는 객체(object)들을 비슷한 특징을 갖는 몇 개의 그룹(cluster)으로 구분하여 각 그룹들의 특성을 찾는 탐색적 자료 분석(exploratory data analysis) 과정이다.

탐색적 자료 분석은 대용량 데이터에서 복잡한 관계를 이해하는데 매우 도움이 된다. 대용량 데이터는 개개의 관찰치를 요약하는 것보다는 전체를 유사한 그룹으로 구분한 후, 그룹을 잘 대표하는 클러스터들을 관찰함으로써 전체 데이터에 대한 구조나 특성을 파악하는 것이 의미 있다고 할 수 있다.

대용량 데이터에 있어 '자연스러운 클러스터'를 찾는 일은 매우 중요한 탐색적 방법이다. 자연스러운 클러스터를 정의하는 일은 실질적으로 매우 어려운 일이며 유사도(similarity)를 정의하는 방법에 따라 클러스터 결과가 달라 질 수 있다. 예를 들어 16장의 페이스 카드를 유사한 클러스터로 분류하는 방법은 슈트(suits)별로 묶으면 4개 클러스터가 되고, 검정과 붉은 색상에 따라 묶으면 2개 클러스터, 메이저 또는 마이너에 따라 묶으면 2개 클러스터, 하트 및 퀸 스페이드와 나머지 슈트들로 분류하면 2개 클러스터, 페이스가 같은 것끼리 묶으면 4개 클러스터 등으로 다양한 클러스터 결과가 가능하다.

클러스터링의 많은 응용에 있어서 '좋은 클러스터'와 '나쁜 클러스터'를 알 수 있으나 가장 좋은 클러스터를 고르지 않는 이유는 다음과 같이 생각할 수 있다. 앞의 카드 예에서 2개 클러스터로 나눌 수 있는 가능한 경우의 수는 32,767가지이고, 3개 클러스터로 나눌 수 있는 경우의 수는 7,14,1686가지가 된다. 즉 모든 가능한 클러스터들을 열거하여 이들 중 가장 좋은 클러스터 결과를 얻는 것은 시간상 제약을 받는다. 따라서 최선의 클러스터 결과를 얻기는 어려우며 대신에

좋은 클러스터를 찾을 수 있는 알고리즘이 필요하다.

가능한 모든 클러스터를 검토하는 일은 거의 불가능하다. 이러한 문제 때문에 가능한 모든 클러스터를 구하지 않고도 합리적인 클러스터들을 찾을 수 있는 클러스터링 알고리즘이 연구되었다.

클러스터링 알고리즘을 선택하는 기준은 데이터를 구성하고 있는 객체 속성(attribute)과 클러스터링 응용에 따라 달라진다(Han과 Kamber 2001, A.K. JAIN 등 1999, Rui Xu 등 2005).

계층적(hierarchical) 클러스터링은 객체들을 연속적으로 병합하거나(agglomerative) 분리하는(divisive) 과정을 거침으로서 진행된다.

병합적 방법은 개별적인 객체로부터 시작한다. 따라서 맨 처음에는 객체만큼의 클러스터가 있게 되고, 이러한 초기 클러스터들은 그들의 유사도에 따라 병합되어 하나의 클러스터로 묶여지게 된다. 새로운 클러스터가 형성됨에 따라 클러스터간의 유사도는 점차적으로 줄어들며 최종적으로 모든 부분 클러스터들이 묶이어 한 개의 단일 클러스터를 만들게 된다. 병합적 방법은 bottom-up 접근 방법이라 할 수 있다.

분리적 방법은 병합적 방법과 정반대로 진행된다. 모든 객체들이 단일 클러스터에서 시작하여 맨 처음에는 두 개의 부분 클러스터로 분리되는데 이때 두 부분 클러스터는 가능한 서로 멀리 떨어지도록 분리된다. 즉 두 부분 클러스터간의 거리(distance)가 최대가 되는 클러스터들로 분리된다. 그 다음에는 같은 방법으로 각 부분 클러스터들이 다시 비유사한 부분 클러스터들로 분리되는 과정을 반복하게 되고, 최종적으로는 객체 수만큼의 부분 클러스터로 세분화되어 하나의 클러스터에 하나의 데이터 객체가 할당될 때까지 반복되거나 또는 종료 조건에 도달할 때까지 반복된다. 분리적 방법은 top-down 접근 방법이라 할 수 있다.

계층적 클러스터링은 어떤 객체가 하나의 클러스터에 포함이 되면 다른 클러스터로는 이동하지 못하는 특성이 있다.

분할적(partitioning) 클러스터링은 객체들을 몇 개의 클러스터로 분할하는 방법이며, 기준함수를 최적화하는 클러스터를 찾는다.

클러스터를 구성하는 과정에서 클러스터에 포함되어 있는 객체들은 (재)할당이

반복적으로 발생하기 때문에 초기(initial)에 어떤 객체가 부적절하게 클러스터에 할당된다 하더라도 다른 클러스터에 포함될 수 있는 특성이 있다.

분할적 클러스터링은 n개의 객체로 이루어져 있는 데이터로부터 $k(k \leq n)$개의 분할된 클러스터를 구성하고, 다음 사항들을 만족하여야 한다.

- 각각의 분할은 적어도 하나 이상의 객체를 포함하고 있어야 한다.
- 하나의 객체는 하나의 클러스터에만 속해야 한다.

k개의 초기 분할로 구분된 클러스터들은 (재)할당을 반복하여 클러스터를 갱신(update)한다. 클러스터의 갱신이란 같은 클러스터에 속한 데이터 객체들 사이의 유사도는 커지고, 다른 클러스터에 속한 데이터 객체와는 유사도가 작아짐을 의미한다. 클러스터 수 k는 미리 규정되거나 클러스터 절차의 한 부분으로 결정될 수 있다.

분할적 클러스터링은 클러스터 단계마다 유사도를 다시 구하거나 또는 컴퓨터 작업 수행 중에 기본 데이터를 저장할 필요가 없기 때문에 대용량 데이터에 적합하다.

분할적 클러스터링은 속성들을 분할하여 초기 클러스터를 만들거나 또는 클러스터의 중심(nucleus)을 형성하게 될 초기 클러스터를 결정하는 것으로부터 출발한다. 클러스터링 절차에서 훌륭한 출발점이 되기 위해서는 명백한 편의로부터 영향을 받지 않도록 해야 한다. 그 중 한 가지는 초기 클러스터를 속성들로부터 임의로 선택하거나 또는 속성들을 랜덤하게 나누어 초기 클러스터를 형성하는 방법 등이 있다.

분할적 클러스터링 방법은 두 데이터 객체 사이의 유사도를 기반으로 클러스터를 구성하기 때문에 구(spherical) 형태의 클러스터만을 찾을 수 있다. 임의의 모양을 하고 있는 클러스터는 찾기 어렵게 된다.

밀도 기반(density based) 클러스터링은 이웃(neighbor)의 밀도가 기준치 이상이 되면 클러스터를 확장시켜 나가는 방법이다. 밀도 기반 클러스터링 방법은 임의의 모양을 갖춘 클러스터를 찾아낼 수 있으며 잡음(noise)을 제거할 수 있다.

격자 기반(grid based) 클러스터링 방법은 데이터 객체의 수에는 독립적이고, 단체 공간을 격자 구조의 유한한 칸으로 양자화 한다. 모든 클러스터링 동작은 이러한 격자 구조의 격자 칸(cell) 수에만 영향을 받게 된다.

모델 기반(model based) 클러스터링 방법은 각각의 클러스터별로 모델을 가정하고 해당 모델에 가장 잘 맞도록 클러스터를 구성한다.

대용량 데이터에서 객체들에 대한 간단한 클러스터 구조를 만들어 내기 위해서는 근접성(closeness) 또는 유사도 측도가 필요하다. 유사도 측도는 속성의 성질(이산, 연속, 이항 등) 또는 측정척도(명목, 순서, 구간, 비율 등) 등을 고려해야 한다. 데이터 객체들이 클러스터링 될 때 근접도는 일종의 거리 개념에 의해 표현된다.

클러스터링은 객체들이 취하는 속성에 따라 클러스터 결과가 달라지므로 데이터를 구성하고 있는 객체들의 속성을 먼저 측정하여야 한다.

데이터를 구성하고 있는 객체의 속성은 크게 세 가지로 나눌 수 있다. 첫째, 수치적인(numerical) 데이터는 데이터를 구성하고 있는 객체의 속성들이 모두 수치로 측정된 데이터를 의미한다.

등간척도(interval measure), 비율척도(ratio measure) 등이 수치 데이터에 속한다. 이러한 수치 데이터는 다음 2장에서 설명되는 유사도를 기준으로 클러스터링 한다. 수치 데이터는 각 속성의 측정 단위가 서로 다르면 클러스터 중요도가 달라지므로 각 속성들을 표준화한 후에 측정한다.

둘째, 범주적인(categorical) 데이터는 데이터를 구성하고 있는 객체의 속성들이 모두 범주로 측정된 데이터를 의미한다.

명목척도(nominal measure), 이항척도(binary measure) 등이 범주 데이터에 속한다. 이러한 범주 데이터는 다음 2장에서 설명되는 유사도를 기준으로 클러스터링 한다.

셋째, 혼합된(mixed) 데이터는 데이터를 구성하고 있는 객체의 속성들이 수치 속성과 범주 속성이 혼합되어 측정된 데이터를 의미하며, 유사도 또는 상이도를 기준으로 클러스터링 한다.

분할적 클러스터링은 초기 클러스터 결정 문제를 해결해야 하며, 클러스터링 대상이 되는 객체의 속성에 따라 클러스터링 하기 위한 유사도를 계산하여야 한다.

따라서 본 논문에서는 데이터 속성에 따른 유사도를 정의한 후 클러스터링 하는 방법에 대해 설명한다. 특히 데이터 객체들이 혼합된 속성을 구성하고 있는 혼합 데이터에 대한 초기 클러스터 결정 문제와 클러스터 결과를 분할적 클러스터링 알고리즘의 하나인 K-means paradigm을 이용한 알고리즘을 제안하고, 실험을 통하여 결과를 보이고자 한다.

본 논문은 다음과 같이 구성된다. 먼저 2장에서는 클러스터링의 일반적 과정 및 요구사항을 살펴보고, 3장에서는 그 동안 연구되어 왔던 K-means 기반 클러스터링 방법들에 대해 살펴보고, 4장에서는 본 논문에서 제안하는 데이터 속성에 따른 클러스터링 알고리즘에 대해 논한다. 그리고 5장에서는 실험을 통해 제안한 알고리즘의 성능을 평가하고, 6장에서 결론을 맺는다.

II. 클러스터링의 개념

1. 클러스터링의 배경

클러스터링의 기본 목적은 속성(또는 변수)들의 자연스러운 클러스터를 찾아내는 것이다. 자연스러운 클러스터를 찾아내기 위해서는 객체들 간의 연관성(유사도 또는 상이도)을 측정할 수 있도록 양적인 척도 또는 질적인 척도가 필요하며, 분류[1]와는 달리 클러스터의 수나 그 구조에 대해 아무런 가정을 하지 않는다.

대용량 데이터는 개개의 관찰치를 요약하는 것보다는 전체를 유사한 집단으로 구분한 후, 집단을 잘 대표하는 클러스터들을 관찰함으로써 전체 데이터에 대한 의미 있는 결과를 얻어내는 것이 의미 있다고 할 수 있다.

클러스터링은 특별한 정보나 배경지식을 필요로 하지 않고 데이터들 간의 주어진 척도를 이용하여 결과를 이끌어내므로 비교사 학습(unsupervised learning)에 속하는 패턴 분류 방법이다.

클러스터링 기법은 거리 기반(distance based) 클러스터링을 중심으로 오랫동안 연구되어 왔다. 클러스터링에 대한 대부분의 연구는 점(point)들 간의 거리로 데이터 사이의 거리가 명확하게 정의되는 수치 데이터를 다루거나 데이터의 거리가 정의되지 않는 범주 데이터를 구분하여 다루어 왔다. 하지만 현실에는 수치 데이터와 범주 데이터가 혼합되어 이루어진 데이터 집합이 많이 존재한다. 이러한 혼합 데이터에 대한 클러스터링 연구는 최근에 연구되기 시작하였다.

K-means 알고리즘(Macqueen, 1967)은 수치 데이터에 대한 분할적 클러스터링 방법으로 가장 많이 사용되는 기법 중 하나이다. 이 알고리즘은 각 객체를 유사한 특성을 갖는 k개의 클러스터로 분할하는 방법으로, 각 클러스터에 속하는 객체들의 평균을 중심으로 근접한 거리에 있는 객체들을 묶어서 분할한다. 이 알고

1) 분류(classification)는 미리 정의된 그룹으로 데이터를 구분하는 것에 반해, 클러스터링은 데이터의 속성 값에 따라 그룹을 구분하는 기법으로 데이터가 갖는 속성 값에 따라 각 그룹의 범위나 성격이 다르게 정의된다.

리즘은 평균을 클러스터의 중심으로 사용하기 때문에 이상값(outlier)에 민감하다.

이를 해결하기 위한 알고리즘으로 K-medoids 알고리즘 등이 연구되어 왔다.

K-medoids 알고리즘은 클러스터의 중심으로 평균을 사용하지 않고, 클러스터내의 중심에 위치한 중앙값(메도이드)을 사용한다. 클러스터내의 중앙값을 중심으로 사용하기 때문에 K-means 알고리즘에 비해 이상값에 덜 민감하다고 알려졌다(kaufman과 Rousseeuw 1990).

이 알고리즘은 이상값에는 덜 민감하지만, 클러스터내의 중심인 메도이드를 찾기 위해 모든 가능한 객체들의 쌍을 비교하면서 반복적으로 최선의 메도이드를 선택하기 때문에 대용량 데이터에 비효율적이다.

대용량 수치 데이터를 클러스터링 하기 위해 K-means 알고리즘이 K-medoids 알고리즘에 비해 효율적이기는 하지만, 초기 클러스터 결정 문제가 발생한다. 이 문제를 해결하기 위해 초기 클러스터 결정에 관한 연구가 진행되어 왔다(Pe˜na 등 1999, Khan과 Ahmad 2004, Bae와 Roh 2005).

K-modes 알고리즘(Huang, 1997a)은 범주 데이터를 대상으로 K-means 알고리즘을 확장하여 적용한 알고리즘이다. 이 알고리즘도 대용량 데이터에 적합하지만, 분할적 클러스터링 방법이기 때문에 초기 클러스터 결정 문제가 발생한다. 이 문제를 해결하기 위한 초기 클러스터 결정에 관한 연구로는 Sun 등(2002), 양순철 등(2007)이 있다.

K-prototypes 알고리즘(Huang 1997b)은 혼합 데이터를 대상으로 K-means paradigm을 이용하여 수치 데이터와 범주 데이터를 나눈 후 비용함수를 계산하는 알고리즘을 제시하였으며, Ahmad와 Dey(2007)는 혼합 데이터를 대상으로 K-means paradigm을 이용하여 수치 데이터와 범주 데이터를 나눈 후, 유사도 측도를 객체 간 속성 값의 동시발생(co-occurrences)값으로 측정하는 새로운 비용함수를 제안하였다. 두 알고리즘은 분할적 클러스터링의 K-means paradigm을 이용하며, 초기 클러스터를 랜덤하게 선택한다.

2. 클러스터링의 일반적 과정

1) 속성 측정(attribute measurement)

클러스터링은 데이터를 구성하는 객체들이 취하는 속성 값에 따라 몇 개의 그룹으로 구분하여, 데이터 전체의 구조에 대한 이해를 돕고자 하는 방법이다. 따라서 클러스터링은 객체들이 취하는 속성에 따라 클러스터 결과가 달라지므로 데이터를 구성하고 있는 객체들의 속성을 먼저 측정하여야 한다. 즉, n개의 객체에 대하여 p개의 속성을 측정한다.

2) 유사도 측정(similarity measurement)

대용량 데이터에서 객체들에 대한 간단한 클러스터 구조를 만들어 내기 위해서는 근접성(closeness) 또는 유사도 측도가 필요하다.

측정된 속성들을 이용하여 모든 객체들 간의 유사도를 계산한다. 유사도 값은 클수록 두 객체가 가까운 것을 의미하고, 작을수록 두 개체가 먼 것을 의미한다.

데이터 객체들이 클러스터링 될 때 근접도는 일종의 거리 개념에 의해 표현된다.

속성들이 의미 있는 p차원 측정값으로 표현된 데이터 객체들의 속성 쌍에 대한 유사도 계산은 다음과 같다.

p 차원 상의 두 객체 $X=(x_1, x_2, ..., x_p)$와 $Y=(y_1, y_2, ..., y_p)$간의 유클리디안 거리(euclidean distance)는

$$d(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_p - y_p)^2} = \sum_{i=1}^{p} |x_i - y_i|^2$$

로 정의된다.

분할적 클러스터링에서 유사도 측정 기준은 n개의 데이터 객체들을 k개의 클러스터로 구분하는 것으로 다음의 두 가지 측정에서 높은 값을 가지도록 하는 것이 좋다

- Homogeneity: 같은 클러스터에 속한 객체들 사이의 가까운 정도
- Separation: 다른 클러스터에 속한 객체들 사이의 먼 정도

수치 속성에 대한 유사도(거리)는 다음 조건을 만족한다.

a) $d(x_i, x_j) = d(x_j, x_i)$

b) $d(x_i, x_j) \geq 0$

c) $d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$

d) 만약 $x_i = x_j$ 이면 $d(x_i, x_j)$

속성들이 의미 있는 측정값으로 표현되지 않는 성질을 갖고 있는 데이터 객체들의 속성에 대한 유사도는 속성들이 어떤 특정한 성질을 갖고 있는지 여부에 따라 속성들의 쌍을 비교하게 된다.

유사 항목 쌍은 그렇지 않은 항목 쌍에 비해 공통된 성질들을 더 많이 갖게 된다. 어떤 특징을 갖는지 여부는 이항변수(특징을 가지면 1, 그렇지 않으면 0)로 표현할 수 있다.

x_{ij}를 i 번째 항목에 대한 j번째 이항변수 값(1 또는 0)이라 하고, x_{kj}는 k번째 항목에 대한 j번째 이항변수 값이라 하면, 유사도 계산은 다음과 같다.

$$(x_{ij} - x_{kj})^2 = \begin{cases} \delta(x_{ij}, x_{kj}) = 0, & x_{ij} = x_{kj} \\ \delta(x_{ij}, x_{kj}) = 1, & x_{ij} \neq x_{kj} \end{cases}$$

유클리디안 거리 $\sum_{j=1}^{p}(x_{ij} - x_{kj})^2$은 일치하지 않은 짝(mismatch)의 개수를 나

타낸다. 거리가 크면 일치하지 않은 짝, 즉 유사도가 낮은 것(상이도는 높음)을 나타내고, 거리가 작으면 유사도가 높은 것(상이도는 낮음)을 나타낸다.

범주 속성에 대한 유사도(상이도)는 다음 조건을 만족한다.

a) $\delta(x_i, x_j) = \delta(x_j, x_i)$

b) $0 \leq \delta(x_i, x_j) \leq 1$

c) $\delta(x_i, x_j)\delta(x_j, x_k) \leq [\delta(x_i, x_j) + \delta(x_j, x_k)]\delta(x_i, x_k)$

d) 만약 $x_i = x_j$ 이면 $\delta(x_i, x_j) = 1$ 그렇지 않으면 $\delta(x_i, x_j) = 0$

3) 클러스터링

각 객체들의 쌍을 대상으로 유사도를 측정한 후에 각 객체들을 클러스터링 한다. 클러스터링 방법은 객체의 속성 또는 클러스터 결과의 응용이나 다음 3절에서 설명될 요구사항에 따라 고려된다.

4) 각 클러스터의 성격이나 상호관계를 파악한다.

클러스터링을 통해 형성된 클러스터에 속한 데이터들은 하나의 그룹으로 간주되어 응용 프로그램에서 사용된다.

클러스터링을 사용하는 응용 프로그램에는 기업에서 고객의 구매 정보를 사용하여 고객을 구분하거나, 생물학 등에서 식물과 동물을 분류학적 개념을 사용하여 구분하거나 또는 정보 추출(information discovery) 측면에서 웹(web) 상에 존재하는 여러 문서들을 분류하는데 사용되기도 한다.

3. 클러스터링을 위한 요구사항

각각의 응용마다 특별한 요구사항들이 있기 때문에 클러스터링 알고리즘은 개별 응용의 요구를 충족하기 위하여 여러 가지 항목들에 대하여 고려해야 한다. 다음은 클러스터링 알고리즘에서 나타나는 요구사항들이다.

1) 확장성

수백만 이상의 객체를 갖는 대용량의 데이터베이스에서도 처리 가능한 복잡도(complexity)가 선형적으로 증가할 수 있는 확장 가능한 클러스터링 알고리즘

2) 다른 종류의 속성을 다룰 수 있는 능력

수치 데이터, 범주 데이터 또는 혼합된 데이터를 포함하고 있는 객체에 대한 클러스터링 알고리즘

3) 임의 모양의 클러스터 찾기

대부분의 거리 기반 클러스터링은 구형의 클러스터를 발견한다. 현실적으로 클러스터의 모양은 임의의 형태로 존재하기 때문에 임의의 모양을 찾을 수 있는 클러스터링 알고리즘

4) 입력 매개변수의 최소 요구치

클러스터 결과 값은 초기 입력 매개변수 값에 따라 크게 달라지고 그 값을 정하는 것은 어렵다. 대부분의 알려진 클러스터링 알고리즘은 사용자로 하여금 초기 매개변수 값을 입력하도록 되어 있기 때문에 어려움이 있다.

5) 잡음이 섞인 데이터 취급 능력

현실 데이터에는 잡음(오류 또는 불분명한 데이터) 이 있는 경우가 있다. 잡음이 섞인 데이터가 있는 경우 클러스터의 결과는 나빠질 수 있다.

6) 입력 순서에 무반응

객체 순서에 따라 클러스터 결과가 달라질 수 있다. 객체 입력 순서에 영향을 받지 않는 클러스터링 알고리즘

7) 고차원

고차원 데이터에 대해서도 좋은 결과를 얻을 수 있는 클러스터링 알고리즘

8) 해석과 유용성

클러스터 결과를 사용자가 쉽게 이해할 수 있어야 하며 해석 가능하여야 한다.

본 논문에서는 이러한 알고리즘 요구사항들 중에서 대용량 처리를 위한 확장성과 데이터 속성에 따른 클러스터링 알고리즘, 분할적 클러스터링에서 고려해야 할 초기 클러스터 결정 문제를 바탕으로 하는 알고리즘을 제안한다.

III. K-means 기반 클러스터링

본 장에서는 그 동안 연구되어왔던 클러스터링 알고리즘에 대해 알아본다. 먼저 수치 데이터에 대한 K-means 기반 클러스터링 알고리즘을 살펴보고, 다음으로 범주 데이터, 혼합 데이터에 대한 K-means 기반 클러스터링 알고리즘을 알아본다.

1. 수치 데이터에 대한 K-means 기반 알고리즘

분할적 클러스터링은 n개의 객체, p개의 수치 속성으로 구성된 데이터에서 k개의 클러스터로 분할하면 알고리즘은 종료한다.

클러스터 구성은 유사도 함수를 사용하여 같은 클러스터에 속한 객체들은 유사도를 크게 하고, 다른 클러스터에 속한 객체와는 유사도를 작게 한다. 분할적 클러스터링 알고리즘에 대해서는 K-means, K-medoids 알고리즘을 알아본다.

1) K-means 알고리즘

K-means 알고리즘(Macqueen 1967, Hartigan 1974)은 수치 데이터에 대한 분할적 클러스터링 방법으로 가장 많이 사용되는 기법 중 하나이다. 이 알고리즘은 각 객체를 유사한 특성을 갖는 k개의 클러스터로 분할하는 방법으로, 각 클러스터에 속하는 객체들의 평균을 중심으로 근접한 거리에 있는 객체들을 묶어서 분할한다.

K-means 알고리즘은 객체간의 거리를 유클리디안 거리로 정의한 후 클러스터의 평균을 계산하여 비용함수를 최소화하도록 클러스터를 구성해 나가는 알고리즘이다.

K-means 알고리즘은 (1) k개의 클러스터를 구성하기 위하여 k개의 초기 클러스터 지정, (2) 객체와 클러스터 중심과의 유사도 계산, (3) 해당 객체와 가장 가까운 클러스터에 할당, (4) 새로이 할당된 객체들을 사용하여 해당 클러스터의 중심을 갱신하는 과정을 기준 함수(criterion function) 값이 수렴할 때까지 (2)~(4) 과정을 반복한다.

기준 함수는 객체들의 오차 제곱의 합으로 다음과 같다.

$$Cost = \sum_{i=1}^{k}\sum_{j=1}^{C_i}|x_{ij} - c_i|^2 \quad (1)$$

$$c_i = \frac{1}{C_i}\sum_{j=1}^{C_i}x_{ij},\ i = 1, 2, ..., k$$

k: 클러스터 수, C_i: i 번째 클러스터의 수

x_{ij}: i 번째 클러스터의 j번째 객체, c_i: i 번째 클러스터의 중심

(1) k개의 객체를 랜덤하게 선택하여 클러스터의 중심으로 할당
(2) 반복
(3) 각각의 데이터 객체와 클러스터 평균과 비교하여 가장 가까운 클러스터에 해당 객체를 (재)할당
(4) 클러스터의 평균을 새로 갱신
(5) 클러스터에 변화가 없을 때까지 반복

Figure 1. K-means algorithm

K-means 알고리즘은 클러스터내의 중심 계산을 평균으로 하기 때문에 클러스터의 평균이 정의되어 질 수 있는 수치 데이터에만 사용할 수 있다는 제약이 있으며, 이상값에 민감하다. 이를 해결하기 위해 실제 관측치인 메도이드(medoid)를 중심으로 사용하는 K-medoids 알고리즘 등이 연구되어 왔다.

메도이드란 클러스터 내에서 객체들간의 평균 상이도가 가장 작은 객체를 의

미한다.

2) K-medoids 알고리즘

K-means 알고리즘으로 얻어진 클러스터 결과는 이상값에 민감하게 반응한다. 극단적으로 큰 값을 갖고 있는 객체가 클러스터에 포함되는 경우 클러스터내의 중심인 평균에 영향을 주기 때문이다.

이상값에 영향을 적게 받기 위해서는 클러스터의 중심으로 평균을 사용하지 않고, 클러스터의 중심에 위치한 중앙값(medoid)을 클러스터의 중심으로 대표하여 사용한다.

클러스터의 중심을 평균 대신에 중앙값으로 사용하는 것 이외에 동일 클러스터 안에서 중심과 객체 사이의 거리 합을 최소화하여 클러스터를 구성한다는 개념은 K-means 알고리즘과 동일하다. 이 알고리즘을 K-medoids 알고리즘이라 한다.

(1) k개의 객체를 랜덤하게 선택하여 클러스터의 메도이드로 할당
(2) 반복
(3) 각각의 데이터 객체와 k개의 클러스터 메도이드와 비교하여 가장 가까운 클러스터에 해당 객체를 (재)할당
(4) 클러스터에 할당되어 있는 메도이드를 제외한 데이터 객체 O_{random}을 랜덤하게 하나 선택
(5) (4)에서 선택된 O_{random}을 중심으로 놓고 나머지 데이터 객체들과의 총 거리의 합 S를 계산
(6) 만약 총합 $S < 0$ 이면 O_{random}을 중심으로 선택하고, 새로운 k개의 메도이드 집합으로 생성
(7) 클러스터에 변화가 생기지 않을 때까지 반복

Figure 2. K-medoids algorithm

K-medoids 알고리즘은 K-means에 비해 이상값에 덜 민감하다고 알려졌다 (Kaufman과 Rousseeuw 1990).

K-medoids 알고리즘은 n개의 데이터 객체 중에서 k개의 데이터 객체를 랜덤하게 선택하여 중앙값으로 놓고, 남아있는 나머지 데이터 객체들은 k개의 메도이드와 거리 측정을 통해서 가장 가까운 메도이드를 갖고 있는 클러스터에 할당한다.

모든 객체들이 클러스터에 속하게 되면, 클러스터에서는 메도이드가 아닌 다른 임의의 데이터 객체를 메도이드로 놓고 현재의 클러스터에 속한 데이터 객체들과의 거리를 측정한다. 만약 새로 선택된 메도이드가 더 작은 거리 총합을 갖게 된다면 그 값을 메도이드로 놓고 이전의 메도이드가 더 작은 거리 총합을 갖는다면 이전의 메도이드를 선택한다. 이러한 과정은 현재의 클러스터에 할당되어 있는 모든 데이터 객체에 대하여 반복한다.

k개의 클러스터에 대하여 메도이드 선택 과정이 끝나면 다시 k개의 메도이드를 초기 클러스터로 앞의 과정을 반복한다. 이후 클러스터가 수렴할 때까지 이 과정을 반복한다.

(1) PAM(Partition Around Medodis)

PAM 알고리즘(kaufman과 Rousseeuw 1990)은 K-medoids의 한 방법으로 클러스터의 중심을 실제 객체인 메도이드를 사용한다.

이상적인 메도이드를 찾기 위해 반복을 통하여 메도이드들을 변화시킨다. 메도이드들을 변화시킬 때마다 객체들이 가까운 메도이드들을 중심으로 객체를 형성하기 위해 재분류된다. 변화된 메도이드로 인하여 이동하는 객체와 본래의 메도이드, 변화된 메도이드와의 거리 차이를 비용이라 한다. 비용함수는 메도이드와 나머지 객체간의 차이 값으로 계산된다.

PAM 알고리즘은 객체들이 이동하면서 발생한 비용을 모두 더한 총 비용을 이용하여 이상적인 메도이드를 찾아낸다. 다음 식 (2)는 PAM 알고리즘에서 이상적인 메도이드를 찾아내기 위한 비용함수이다.

$$TC_{ik} = \sum_{j}(C_{jih}) \quad (2)$$

K-means 알고리즘에서는 객체를 클러스터에 할당하기 위해 거리의 차이를 사용하지만, PAM 알고리즘은 비용함수로 대신한다.

PAM 알고리즘은 모든 가능한 객체들의 쌍을 분석하면서 반복적으로 최선의 메도이드를 선택하기 때문에 대용량 데이터에 비효율적이다(Han과 Kamber 2001).

(2) CLARA(Clustering LARge Applications)

CLARA 알고리즘(kaufman과 Rousseeuw 1990)은 모든 데이터를 사용하지 않고 샘플링을 사용하여 대표 집합을 추출한 후 대표 집합에 PAM 알고리즘을 적용하여 최선의 메도이드를 찾는다.

클러스터의 정확성을 측정할 때 샘플링에서 얻어진 메도이드들과 샘플링된 객체간의 유사도를 계산하는 것이 아니라, 샘플링에서 얻어진 메도이드들과 전체 데이터의 모든 객체와 유사도를 계산한다.

CLARA 알고리즘은 데이터에서 샘플링하기 때문에 표본 크기에 의존한다. PAM 알고리즘은 전체 데이터에서 메도이드를 선택하지만, CLARA 알고리즘은 표본에서 최적의 메도이드를 찾는다. 무작위로 추출된 샘플이 적절하다면 매우 효과적인 방법이며 PAM에 비해 대용량 데이터를 처리할 수 있는 장점을 가지고 있다.

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

CLARANS 알고리즘(Ng와 Han 1994, Ng와 Han 2002)은 PAM과 CLARA 알고리즘을 결합한 방법으로 효과적인 샘플링 방법이 필요하며 샘플링 결과에 따라 알고리즘의 성능이 결정된다.

CLARANS 알고리즘은 그래프 개념을 이용한다. 그래프 $G_{n,k}$는 n개의 객체, k개의 클러스터를 가지는 데이터에서 각각의 노드(node)들의 집합을 의미한다. 노드란 메도이드들의 집합인 $\{O_{m1}, O_{m2}, ..., O_{mk}\}$이다. 만약 두 개의 노드에서 한 개의 메도이드만 다르고 다른 메도이드들은 동일하다면, 두 노드를 이웃이라고 한다. 즉 노드 $S_1 = \{O_{m1}, O_{m2}, ..., O_{mk}\}$과 $S_2 = \{O_{w1}, O_{w2}, ..., O_{wk}\}$는 $|S_1 \cap S_2| = k-1$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 이동할 경우 발생하는 비용은 PAM 알고리즘에서 정의한 (2)의 TC_{ih}를 이용하여 계산한다.

메도이드를 변형시킬 때, 메도이드 O_i는 S_1에 속하는 객체이고, 새로운 메도이드 O_h는 S_2에 속하는 객체이다. O_i, $O_h \not\subset S_1 \cap S_2$로 메도이드 O_i와 O_h는 노드의 교집합에 속하지는 않지만, $O_i \in S_1$, $O_h \in S_2$로 노드에 속하는 유일하게 다른 하나의 메도이드로 새로운 메도이드로 설정한다.

CLARANS 알고리즘은 노드의 모든 이웃들을 평가하지는 않는다(노드 이웃의 표본을 선택하여 평가). CLARA 알고리즘은 반복할 때마다 정해진 표본의 개수만큼 표본을 선택하고, CLARANS 알고리즘은 단계를 거칠 때마다 노드 이웃의 표본을 선택한다. 따라서 CLARANS도 표본 추출에 의존한다.

K-means 알고리즘은 각각의 클러스터는 클러스터의 중심(평균)으로 재 표현되어지고, K-medoids 알고리즘은 각각의 클러스터는 클러스터 안의 객체 중 하나로 재 표현 되어진다.

K-medoids 알고리즘의 하나인 PAM 알고리즘은 클러스터의 중심을 실제 객체인 메도이드를 이용하기 때문에 대용량 데이터에는 적합하지 않으며, CLARA, CLARANS 알고리즘은 PAM 알고리즘에 비해 대용량 데이터를 처리할 수 있다는 장점이 있으나, 표본 추출에 의존한다. 따라서 본 논문에서는 대용량 처리에 적합한 K-means 알고리즘을 수정하여 수치 데이터에 적용하고자 한다.

K-means 알고리즘은 거의 모든 형태의 데이터에 적용이 가능하고 특별한 변환 없이 적용이 쉽고 대용량 데이터 처리에 유용하다는 장점이 있으나 클러스터 수 k에 대한 정보가 필요하고, 부적절한 초기 클러스터의 결정은 잘못된 클러스터의 구성과 클러스터 구성 과정에서 많은 반복을 발생하여 클러스터 구성에 많은 시간이 소요되고, 또한 클러스터 분석의 성능에 상당한 영향을 미치게 되므로 이에 대한 연구가 진행되어 왔다(Peña 등 1999, Khan과 Ahmad 2004, Bae와 Roh 2005).

3) K-means 알고리즘에 대한 초기 클러스터 결정 방법

이번 절에서는 K-means 알고리즘의 초기 클러스터 결정 방법으로 MA 방법, KA 방법 및 Max-Min 방법에 대해서 알아보고자 한다. K-means 알고리즘에서 초기 클러스터 결정은 클러스터의 형성 및 클러스터 형성 시간에 중요한 요인이 되므로 이에 대한 많은 연구가 진행되어 왔다.

초기 클러스터 결정에 대한 연구는 FA(Forgy Approach)방법 (Forgy 1965), MA(Macqueen Approach)방법 (Macqueen 1967), KA(Kaufman Approach)방법 (Kaufman과 Rousseeuw 1990), Max-Min 방법(Bae와 Roh 2005) 등이 있다.

Peña 등(1999)은 FA 방법, MA 방법, KA 방법 그리고 random 방법에 대한 초기 클러스터 결정 방법에 대해 비교한 결과 KA 방법이 유용하다고 결론을 내렸다.

Bae와 Roh(2005)는 대용량의 경우 계산양의 증가를 고려하여 비교한 결과 Max-Min 방법은 초기 클러스터를 선택할 때 기존에 선택된 초기 클러스터와의 거리만을 고려하기 때문에 MA 방법에 비해서는 많은 시간이 소요되지만 데이터가 증가하더라도 KA 방법에서 소요되는 시간만큼 증가하지 않으며, 형성될 클러스터의 중심으로 초기 클러스터가 선택되기 때문에 수렴할 때까지의 반복수가 적고, 형성된 클러스터의 오차 제곱 합도 작게 나타난다고 하였다.

(1) MA 방법

MA 방법은 데이터에서 랜덤하게 k개의 초기 클러스터를 선택하고 나머지 객체들은 초기 클러스터에 가장 가까운 클러스터로 포함시킨 후, 클러스터의 중심을 다시 계산하여 클러스터 중심의 변화량이 임계값(threshold) 이하가 될 때까지 반복하여 클러스터를 구성하게 된다.

이 방법은 랜덤하게 초기 클러스터를 선택하기 때문에 쉽고, 편리하게 사용할 수 있으나 부적절한 값이 선택되었을 때는 잘못된 클러스터를 구성할 수 있다.

(1) 랜덤하게 k개의 초기 클러스터 $s_1, s_2, ..., s_k$를 선택

(2) 각 관측값(x_i)에 대해 초기 클러스터 (s_i)까지의 거리를 계산

$$d_j = \|x_i - s_j\|,\ j = 1, 2, ..., k,\ i = 1, 2, ..., n$$

(3) 각 x_i를 단계 2에서 계산된 k개의 d_j중 가장 작은 클러스터로 할당

(4) 클러스터의 중심을 다시 계산

(5) 클러스터 중심의 변화가 없을 때까지 단계 3과 4를 반복

Figure 3. Macqueen approach method

(2) KA 방법

KA 방법은 데이터의 가장 중앙에 위치한 관측치를 첫 번째 초기 클러스터로 설정하고, 나머지 초기 클러스터는 첫 번째 초기 클러스터와 일정한 거리 이상 떨어져 있으면서, 클러스터가 형성되기 쉽도록 초기 클러스터를 선택하게 했다.

이 방법은 MA방법에 의해서 랜덤하게 초기 클러스터를 선택했을 때 발생할 수 있는 문제점을 해결하고, 형성될 클러스터의 내부에서 초기 클러스터의 중심이 선택되도록 초기 클러스터를 단계적으로 설정하지만, 초기 클러스터를 구한 후 다음 단계의 초기 클러스터를 구하는 과정에서 주변의 모든 값들을 고려하기 때문에 데이터의 크기가 커지는 경우 계산양이 많아진다.

(1) 가장 중앙에 위치한 값을 첫 번째 초기 클러스터 s_1으로 선택

(2) 나머지 모든 데이터 $x_i(x_i \neq s_1, i \neq j),\ i = 1,2,...,n$에 대하여

a. 초기값으로 선택되지 않은 관측값 $x_j(x_j \neq s_1,\ i \neq j),\ j = 1,2,...,n$에 대하여 $C_{ji} = max(D_j - d_{ji}, 0)$을 계산

여기서, $d_{ji} = \|x_i - x_j\|,\ D_j = min(d_{sj})$ (x_s는 선택된 초기 클러스터)

b. x_i에 대해서 $\sum_j C_{ji}$를 계산

(3) $\sum_j C_{ji}$를 최대화하는 x_i를 두 번째 초기 클러스터로 선택

(4) k개의 초기값이 모두 선택될 때까지 단계 2와 3을 반복

Figure 4. Kaufman approach Method

(3) Max-Min 방법

Max-Min 방법(Bae와 Roh 2005)은 MA방법에서 초기 클러스터를 랜덤하게 선택하였을 때 생기는 문제점을 해결하기 위하여 제안된 KA방법이 정교하지만 데이터가 많아짐에 따라 초기 클러스터 설정에 따른 시간이 많이 걸리는 단점을 보완하기 위한 방법이다. 이 방법은 단계적으로 초기 클러스터를 선택하되 선택된 초기 클러스터들이 다음 초기 클러스터를 결정하는 데 정보를 줄 수 있도록 하고, KA방법처럼 많은 계산을 하지 않도록 고안되었다.

Max-Min 방법은 랜덤하게 하나의 관측값을 선택하여 첫 번째 초기 클러스터로 선택하고, 첫 번째 초기 클러스터에서 나머지 관측값과의 거리를 구하여 그 거리를 최대로 하는 관측값을 두 번째 초기 클러스터로 선택한다. 즉 초기 클러스터를 선택함에 있어 초기 클러스터들이 한 곳에 모이는 현상을 방지하기 위해 처음 두 초기 클러스터는 멀리 있는 것을 택하게 한다.

(1) 랜덤하게 하나의 관측값을 선택하여 첫 번째 초기 클러스터 s_1을 결정

(2) 나머지 관측값 $x_i (x_i \neq s_1)$, $i = 1, 2, ..., n$에 대하여 첫 번째 초기 클러스터 (s_1)과의 거리를 최대로 하는 관측값을 두 번째 초기 클러스터 s_2로 결정

(3) 선택되지 않은 나머지 관측값 $x_i (x_i \neq s_l,\ l = 1, 2)$, $i = 1, 2, ..., n$에 대하여

a. 초기값 s_1과 s_2와의 거리를 각각 구하여 이들의 최소값 sd_i를 계산하여 대응

$$x_i \leftarrow sd_i = min\{\|x_i - s_1\|, \|x_i - s_2\|\},\ i = 1, 2, ..., n\ (x_i \neq s_l,\ l = 1, 2)$$

b. 각 관측값 x_i에 대응하는 sd_i를 비교하여 이들의 값을 최대로 하는 관측값을 초기 클러스터 s_3로 선택

$$s_3 = x_p \leftarrow max_{1 \le i \le n}(sd_i) = sd_p$$

(4) 다음 단계의 초기 클러스터 s_m, $m = 4, ..., k$을 결정하기 위해 이전 단계에서 결정된 초기 클러스터들을 추가하면서 다음 단계를 반복하여 k개의 초기 클러스터들이 모두 선택되면 정지

나머지 관측값 x_i, $(x_i \neq s_l,\ l = 1, 2, ..., m-1)$, $i = 1, 2, ..., n$에 대하여

a. 이미 구해진 초기값들과 x_j와의 거리를 각각 구하고 이들 거리의 최소값 계산

$$x_i \leftarrow sd_i = min\{\|x_i - s_1\|, \|x_i - s_2\|, ..., \|x_i - s_{m-1}\|\},\ i = 1, 2, ..., n$$
$$(x_i \neq s_l\ \ l = 1, 2, ..., m-1)$$

b. 각 관측값 x_i에 대해서 얻어진 sd_i를 비교하여 이들의 값을 최대로 하는 관측값을 초기 클러스터로 선택

$$s_m = x_q \leftarrow max_{1 \le j \le n}(sd_j) = sd_q$$

(5) 나머지 관측값들은 구해진 초기값들에 가장 가까운 쪽으로 클러스터를 형성한 후, 다시 클러스터의 중심을 구하여 클러스터 중심의 이동이 임계값 이하가 될 때까지 반복

Figure 5. Max-Min method

수치 데이터에 대한 거리 기반 분할적 클러스터링 이외에 STING(STatistical INformation Grid) 알고리즘(Wang 등 1997)은 통계 정보 그리드 방법으로 데이터 공간을 사각형의 셀로 계층 분할하여 클러스터링 하는데 대용량 데이터 클러스터링에 높은 효율성을 갖지만 고차원의 데이터에 대해 비효율적인 단점이 있다.

CLIQUE(Clustering In QUEst) 알고리즘 (Agrawal 등 1998)은 관련된 차원만을 선택하여 선택된 부분 공간에서 클러스터링을 처리하는 방법으로, 고차원 데이터에서 큰 밀도를 가진 영역을 찾는 효과적인 알고리즘이다. CLIQUE는 부분 공간 클러스터링 방법을 이용하여 고차원 데이터 공간에서 차원의 감소를 시도하여 고차원적인 데이터의 클러스터링에 효과적인 방법을 제시하였으나, 각 차원을 일정한 간격으로 분할하기 때문에 클러스터의 정확한 형태를 표현하기 어려운 단점이 있다.

DBSCAN(Density Based Clustering of Application with Noise) 알고리즘(Ester 등 1996, Ester 등 1997)은 저밀도 지역을 이루는 잡음을 클러스터로부터 제거함으로써 잡음이 있는 데이터 집합을 효과적으로 처리할 수 있고 단 하나의 입력 매개변수만을 필요로 하는 장점이 있다.

BIRCH(Balance Iterative Reducing and Clustering using Hierarchies) 알고리즘(Zhang 등 1996, Zhang 등 1997)은 대용량의 데이터 집합을 위해서 CF(Clustering Feature)와 CF-tree를 이용하여, 클러스터링 하는데 CF-tree는 데이터의 삽입 과정에서 동적으로 구축되며 데이터 점들을 한번만 읽어서 트리를 구성할 수 있다는 장점이 있다. 이 알고리즘은 계층적 알고리즘이기 때문에 어떤 객체가 하나의 클러스터에 포함되면 다른 클러스터로는 이동하지 못한다.

Park과 Ryu(2005)는 그리드 기반으로 표본을 추출한 후, 그리드 별 각 셀 단위로 무게중심을 이용하여 클러스터링을 수행한 결과, 전형적인 기법에 비해 수행시간을 단축시키는 동시에 클러스터링 결과 정확도가 다소 높아졌음을 보였고, 그리드 기반 표본 클러스터링 기법과 비교하여 수행속도 면에서는 차이가 없었으나 더 정확한 클러스터링 결과를 보였다.

2. 범주 데이터에 대한 K-means 기반 알고리즘

이번 절에서는 범주 데이터를 클러스터링 하기 위한 K-means 기반 클러스터링 알고리즘을 알아본다. 대용량 데이터를 다루는 데이터 마이닝은 범주 데이터를 포함하는 경우가 있다.

Ralambondrainy(1995)는 범주 데이터를 포함하는 대용량 데이터를 클러스터링 하기위해 K-means 알고리즘을 이용하였다. 이 알고리즘은 데이터 집합이 많은 범주와 속성을 포함할 때 이진(binary)값으로 바꾸어야 하는 계산 비용 및 저장 공간을 증가시킨다.

김보화와 김규성(2002)은 초기 모드를 랜덤하게 선택하는 경우 범주(category)의 수가 적은 변수에서는 각 수준이 골고루 뽑히지 않을 가능성이 있어, 범주의 수가 적은 변수의 수준이 골고루 선택되도록 하기 위해 객체들을 정렬한 후 각 범주에서 랜덤하게 객체를 선택하는 방법을 제안하였다.

K-modes 알고리즘(Huang 1997a)은 범주 데이터를 대상으로 K-means 알고리즘의 형식을 유지하면서 유사도를 이용하여 범주 데이터에 적합하도록 제안한 방법으로 K-means 알고리즘과 마찬가지로 절차가 간단하고 수렴속도가 빠른 반면, 초기 클러스터 모드(mode)의 결정에 따라 클러스터 결과가 달라질 수 있다.

ROCK 알고리즘(Guha 등 1999)은 범주 데이터를 부울(boolean) 속성 데이터로 변환하여 객체간의 유사도를 정의한 후 데이터의 모든 객체를 동시에 비교하여 유사도가 가장 큰 객체들을 순차적으로 병합해 가는 계층적 클러스터링 방법이다. 이 방법은 유사도가 가장 큰 객체들을 우선적으로 병합하기 때문에 거대 클러스터를 형성한 후에도 클러스터를 형성하지 못하고 남는 객체들이 있을 수 있다.

데이터 집합 $\mathbb{X} = \{X_1, X_2, ..., X_n\}$는 n개의 객체를 포함하고 있고, 각 객체는 m개의 범주 속성을 갖는다고 하자.

$$X_i = (X_{i1}, X_{i2}, ..., X_{im})', \ i = 1, 2, ..., n$$

j번째 범주 속성은 A_j는 l_j개의 수준을 가지며 각 수준을 $c_{j1}, c_{j2}, ..., c_{jk}$라고 하자.

1) ROCK(RObust Clustering using linKs) 알고리즘

ROCK 알고리즘은 클러스터간의 링크(link) 값을 이용하여 클러스터를 순차적으로 병합하는 계층적 클러스터링 방법이다.

(1) 유사도 정의

$$sim(x_i, x_j) = \frac{m - \sum_{k=1}^{m} \delta(x_{ik}, x_{jk})}{m + \sum_{k=1}^{m} \delta(x_{ik}, x_{jk})} \tag{3}$$

$$\delta(a, b) = \begin{cases} 0 & , a = b \\ 1 & , a \neq b \end{cases}$$

유사도 $sim(x_i, x_j)$는 0과 1사이의 값을 가지며, 두 객체가 유사할수록 큰 값을 갖는다. 두 객체간의 유사성이 주어진 기준값(threshold) θ보다 크면 두 객체를 이웃(neighbor)라 하고, 객체 x_i의 이웃군 $G(x_i)$은 x_i와 이웃인 객체들의 집합으로 정의한다.

$$G(x_i) = \{x_i | sim(x_i, x_j) \geq 0, i \neq j\} \tag{4}$$

두 객체의 링크 $link(x_i, x_j)$는 두 객체의 이웃군에 속하는 공통 이웃의 개수로 정의하고, 두 클러스터 C_i와 C_j의 링크는 클러스터에 속하는 객체들의 링크의 합으로 정의한다.

$$link(C_i, C_j) = \sum_{x_p \in C_i, x_q \in C_j} link(x_p, x_q) \tag{5}$$

클러스터의 병합을 위해 클러스터 간의 링크 값, 클러스터에 속하는 객체 수 및 유사성의 기준값 θ를 고려한 $g(C_i, C_j)$을 계산한 후, $g(C_i, C_j)$값이 가장 큰 두 클러스터를 병합한다.

$$g(C_i, C_j) = \frac{link(C_i, C_j)}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}} \tag{6}$$

n_i: 클러스터 C_i에 속하는 객체의 수, $f(\theta) = (1-\theta)/(1+\theta)$

클러스터의 병합 기준 g는 θ의 함수이고, θ값에 따라 병합되는 클러스터의 수가 달라지기 때문에, 실제 데이터에서는 θ의 값을 데이터 특성에 따라 정해주어야 한다.

(1) 주어진 θ에 대하여 객체간의 유사성을 계산하여 객체별로 이웃과 이웃군을 계산
(2) 클러스터간의 링크 값 $link(C_i, C_j)$을 계산한 후, $g(C_i, C_j)$을 계산
(3) $g(C_i, C_j)$값이 가장 큰 두 클러스터를 병합하고 클러스터간의 링크 값을 갱신한다. 이 때, 병합된 클러스터와 다른 클러스터간의 링크 값은 병합되기 전의 링크 값의 합으로 계산
(4) 클러스터의 개수가 일정 수에 이를 때까지 단계 2와 단계 3의 과정을 반복

Figure 6. ROCK algorithm

ROCK 알고리즘은 링크라는 개념을 이용하여 두 객체 뿐 만 아니라 데이터의 모든 객체를 동시에 비교하여 유사한가를 판단한다. 그리고 어떤 클러스터를 먼저 병합할 것인가를 판단할 때 클러스터의 크기까지 고려하기 때문에 상대적으

로 크기가 작은 클러스터도 제대로 유지할 수 있다는 장점이 있다. 그러나 병합하는 과정에서 거대 클러스터와 클러스터를 형성하지 못하고 남아 있는 객체가 존재 할 수 있으며, 병합을 계속 할 경우 남은 객체가 거대 클러스터에 포함되는 것이 아니라 거대 클러스터들이 서로 병합되는 일이 생길 수 있고 기준값(θ)에 따라 클러스터의 수와 크기가 달라지기 때문에 최적의 기준값을 정해야 하나 현재로서는 경험적으로 정하는 방법이외에는 존재하지 않는다.

2) K-modes 알고리즘

K-modes 알고리즘은 K-means 알고리즘을 범주 속성의 도메인으로 확장한 알고리즘이다. 이 알고리즘은 K-means 알고리즘에 대해 다음 3가지를 수정한 알고리즘이다.

- 유사도(상이도) 측정
- 클러스터의 중심은 모드 사용
- 모드의 갱신은 빈도 기반

(1) 유사도(상이도) 측정

m개의 범주 속성을 갖는 두 객체 X_{i1},X_{i2}의 유사도 측정 방법은 객체 사이에 대응하는 속성 범주의 일치하지 않는 횟수의 총합을 고려한다. 일치하지 않는 횟수가 적을수록 두 객체 사이의 유사도는 작게 된다.

$$d(X_{i1}, X_{i2}) = \sum_{j=1}^{m} \delta(X_{i1j}, Y_{i2j}) \tag{7}$$

$$\delta(a,b) = \begin{cases} 0 & , a = b \\ 1 & , a \neq b \end{cases}, \; j = 1, 2, \ldots, m$$

(2) 클러스터의 모드(mode of cluster)

$\mathbb{X}$를 범주 속성 $A_1, A_2, ..., A_m$을 갖는 범주 객체 집합 이라 하자. $\mathbb{X}$의 모드는 다음을 최소로 하는 벡터 $Q=(q_1, q_2, ..., q_m)'$로 정의된다.

$$D(Q, \mathbb{X}) = \sum_{i=1}^{n} d(X_i, Q), \text{ where } \mathbb{X} = \{X_1, X_2, ..., X_n\} \tag{8}$$

집합 $\mathbb{X}$에 대응되는 모드인 $Q=(q_1, q_2, ..., q_m)'$는 다음과 같다.

집합 $\mathbb{X}$에서 속성 A_j가 수준 c_{jk}를 갖는 빈도 수를 n_{jk}라 하면, A_j가 c_{jk}를 취할 상대빈도는 다음과 같다.

$$fre(A_j = c_{jk}|\mathbb{X}) = \frac{n_{jk}}{n},\ k = 1, 2, ..., l_j \tag{9}$$

모든 $j = 1, 2, ...m$에 대하여 다음 식 (10)

$$fre(A_j = q_j|\mathbb{X}) \geq fre(A_j = c_{jk}|\mathbb{X}) \tag{10}$$

을 만족하는 $q=(q_1, q_2, ..., q_m)'$는 상이도의 합 $D(Q, \mathbb{X})$을 최소로 하므로, 결과적으로 집합 $\mathbb{X}$의 모드가 된다. 즉, 속성별로 가장 큰 속성 범주 값들의 조합이 그 집합의 모드가 된다.

$$\sum_{i=1}^{n} d(X_i, Q) = \sum_{i=1}^{n}\sum_{j=1}^{m} \delta(x_{i,j}, q_j) = \sum_{j=1}^{m}(\sum_{i=1}^{n} \delta(x_{i,j}, q_j))$$

$$= \sum_{j=1}^{n} n(1 - \frac{n_{q_j}}{n}) = \sum_{j=1}^{m} n(1 - f(A_j = q_j|\mathbb{X})) \tag{11}$$

(1) k개의 객체를 랜덤하게 선택하여 초기 클러스터 모드 $\{q_1^0, q_2^0, ..., q_k^0\}$를 선택

(2) 반복

(3) 각각의 데이터 객체와 k개의 클러스터 모드와 거리 측정을 통하여 가장 유사한 클러스터에 해당 객체를 (재)할당

(4) 클러스터 모드를 새로 갱신

(5) 클러스터에 변화가 없을 때까지 반복

Figure 7. K-modes algorithm

K-modes 알고리즘은 K-means 알고리즘 형식을 유지하면서 몇 가지 요소들만을 범주 데이터에 적합하도록 수정한 것이라 할 수 있다. 따라서 대용량 범주 데이터에 대해 효율적인 클러스터링 알고리즘이라 할 수 있고, 데이터를 변환하지 않아도 되기 때문에 결과에 대한 해석이 바로 이루어질 수 있다는 장점이 있다.

K-modes 알고리즘은 각 클러스터를 대표하는 모드를 결정할 때 수치 데이터의 평균처럼 모든 데이터의 값이 골고루 반영된 것이 아니라, 비용 함수를 최소로 하는 모드를 결정한다. 모드는 클러스터 내에 포함된 모든 데이터 객체의 평균과 같이 균일하게 표현한 것이 아니라 해당 클러스터에서 제일 많은 비중을 차지하는 데이터의 선택 값을 채택한다. 이 결과는 클러스터의 중심인 모드는 해당 클러스터를 제대로 대표하지 못하고 있음을 보여준다. 따라서 K-modes 알고리즘은 속도 면에서는 효과적일 수 있으나 클러스터의 중심을 제대로 찾지 못하기 때문에 정확도 측면에서 좋지 않은 결과를 보일 수 있다.

3. 혼합 데이터에 대한 K-means 기반 알고리즘

이번 절에서는 혼합 데이터를 클러스터링 하기 위한 K-means 기반 클러스터링 알고리즘을 알아본다.

Huang(1997b)은 혼합 데이터를 대상으로 K-means paradigm을 이용하여 수치 데이터와 범주 데이터를 나눈 후 비용함수를 계산하는 알고리즘을 제시하였으며, Ahmad와 Dey(2007)는 혼합 데이터를 대상으로 K-means paradigm을 이용하여 수치 데이터와 범주 데이터를 나눈 후 유사도 측도를 객체 간 속성 값의 동시발생(co-occurrences) 값으로 측정하고 새로운 비용함수를 제안하였다.

1) K-prototypes 알고리즘

K-prototypes 알고리즘은 수치 데이터와 범주 데이터에 대한 클러스터링을 K-means 알고리즘을 기반으로 한다.

K-prototypes 알고리즘의 비용 함수는 다음과 같다.

$$\zeta = \sum_{i=1}^{n} \vartheta(d_i, C_j) \tag{12}$$

m_r: 수치 속성, m_c:범주 속성, $m = m_r + m_c$,

$$\vartheta(d_i, C_j) = \sum_{t=1}^{m_r} (d_{it}^r - C_{jt}^r)^2 + \gamma_j \sum_{t=1}^{m_c} \delta(d_{it}^c, C_{jt}^c) \tag{13}$$

d_i: 객체, d_{it}^r: 수치 속성, d_{it}^c: 범주 속성

$C_j = (C_{j1}, C_{j2}, ..., C_{jm})$: 클러스터 j의 중심

C_{jt}^r, C_{jt}^c: j번째 클러스터 수치 속성 t의 평균과 범주 속성 t의 모드

범주 속성은 다음과 같은 지시함수를 갖는다.

$$\delta(x_i, y_i) = \begin{cases} 0 & , x_i = y_i \\ 1 & , x_i \neq y_i \end{cases}$$

γ_j는 클러스터 j와 범주 속성의 클러스터링을 위한 가중값이다.

혼합된 데이터 클러스터링에 대해 비용함수 ζ는 최소화된다.

K-prototypes 알고리즘은 수치 데이터와 범주 데이터를 독립적으로 클러스터링 하며 다음과 같은 단점이 있다.

(a) 범주 속성을 클러스터링에서 각각의 클러스터를 대표하는 모드를 정할 때 수치 데이터의 평균처럼 모든 데이터의 값을 반영하는 하지 않는다. 모드는 클러스터 안에 포함된 모든 데이터들의 값을 평균과 같이 균일하게 표현한 것이 아니라 해당 클러스터에서 제일 많은 비중을 차지하는 값을 선택한 것이다. 따라서 클러스터의 중심인 모드는 해당 클러스터를 제대로 대표하지 못한다.

(b) 두 범주 속성 값 x_i, y_i 대한 거리는 이진 값으로 구성된다.

$$\delta(p, q) = 0, p = q,\ \delta(p, q) = 1, p \neq q.$$

$\delta(p, q)$는 클러스터내 속성 값의 쌍에 대한 상대 빈도에 의존한다.

(c) K-prototype 알고리즘은 모든 속성이 수치인 경우 가중값은 1이다. 범주 속성이 포함되어 있는 경우 가중값 γ_j은 사용자가 결정해야 한다.

(1) k개의 객체를 랜덤하게 선택하여 k개의 초기 프로토타입을 선택
(2) 각각의 데이터 객체를 식 (13)에 의해 가장 가까운 프로토타입에 해당하는 클러스터로 할당한 후, 각 클러스터의 프로토타입을 갱신
(3) 클러스터에 할당한 후, 현재 프로토타입과 객체간의 유사도를 계산한다. 만일, 다른 클러스터의 프로토타입과의 유사도가 더 크면 해당 객체를 해당 클러스터로 할당하고, 그렇지 않으면 클러스터내의 프로토타입를 갱신
(4) 단계 (3)을 변화가 없을 때까지 반복

Figure 8. K-prototypes algorithm

2) Ahmad와 Dey의 제안 알고리즘

Ahmad와 Dey가 제안한 알고리즘(2007)은 수치 데이터와 범주 데이터에 대해 분할적 클러스터링을 기반으로 한다. Ahmad와 Dey가 제안한 알고리즘은 Huang의 비용함수가 갖는 단점을 보안한 알고리즘이다.

첫째, 범주 속성에 대해 이진 값을 사용하지 않는다. $\delta(x_i, y_i)$는 x_i와 y_i가 다른 속성 값과 동시에 발생하는 값으로 정의한다.
둘째, $\delta(x_i, y_i)$의 계산을 통한 significance는 속성의 클러스터링에 대한 기여도를 나타낸다.

가중값은 속성 값의 분포로부터 계산되므로 사용자 정의가 필요하지 않는다.
수치 속성의 significance는 범주 속성과 동일하게 계산되고, 모든 수치 속성은 클러스터링에 똑같은 기여를 하지 않는다(모든 수치 속성의 significance는 같지 않다).
Ahmad와 Dey가 제안한 비용함수는 다음과 같다.

$$\zeta = \sum_{i=1}^{n} \vartheta(d_i, C_j) \tag{14}$$

m_r: 수치 속성, m_c:범주 속성, $m = m_r + m_c$,

$$\vartheta(d_i, C_j) = \sum_{i=1}^{m_r} (w_t (d_{it}^r - C_{jt}^r))^2 + \sum_{i=1}^{m_c} \Omega(d_{it}^c, C_{jt}^c)^2 \tag{15}$$

$\sum_{t=1}^{m_r} (w_t (d_{it}^r - C_{jt}^r))^2$: 수치 속성에 대한 객체 d_i와 클러스터 중심 C_j의 유사도

$\sum_{t=1}^{m_c} \Omega(d_{it}^c, C_{jt}^c)^2$: 범주 속성에 대한 객체 d_i와 클러스터 중심 C_j의 유사도,

w_t: t번째 수치 속성의 significance(데이터로부터 계산)

K-prototypes 알고리즘을 보완한 Ahmad와 Dey가 제안한 알고리즘의 특징은 다음과 같다.

첫째, 사용자 정의 가중값을 사용하지 않으며 모든 수치 속성(attribute)은 표준화된 거리를 사용한다.

둘째, 범주 속성(attribute)에 대해 이진 값을 사용하지 않는다. 두 속성 사이의 거리는 동시발생 값으로 계산한다.

Ahmad와 Dey가 제안한 알고리즘에서 클러스터 중심 C_j는 이중의 중심으로 표현된다.

수치 속성에서 클러스터의 중심 C_j는 클러스터내 모든 속성 값에 대한 평균으로 표현되고, 범주 속성에서 클러스터의 중심 C_j는 클러스터내 모든 속성 값에 대한 비례 분포로 표현된다.

$\Omega(d_{it}^c, C_{jt}^c)$는 클러스터내 범주 속성 값의 비례 분포로 계산된다.

Huang의 K-prototypes 알고리즘과 Ahmad와 Dey가 제안한 알고리즘은 분할적 클러스터링의 K-means paradigm을 이용하였으며, 초기 클러스터를 랜덤하게 선택한다.

4. 클러스터 ensemble

Zengyou 등(2005)은 범주 데이터 클러스터링 문제를 클러스터 ensemble 관점에서 최적화 문제로 정의하여, 범주 데이터를 클러스터링 하기 위해 클러스터 ensemble 접근을 적용하였으며, Strehl 등(2002a)은 클러스터 ensemble 문제를 최적화 문제로 정의하여, hyper-graph 모델에 기반하여 문제를 해결하는 방법에 대해 combiner를 제안하였고, Strehl 등(2002b)은 멀티 클러스터링 방법을 다음과 같이 제안하였다. 클러스터링 알고리즘의 여러 독립적인 결과는 초기 클러스터에 영향을 받지 않으며, 클러스터링 결과를 이루는 불안정성을 극복하는 클러스터링 결과를 얻기 위해 적합한 결과로 결합된다고 하였다.

다음 장에서는 혼합 데이터를 클러스터링 하기 위한 초기 클러스터 결정 문제와 클러스터 ensemble을 이용한 클러스터링 방법을 제안한다.

IV. 데이터 속성에 따른 클러스터링

본 장에서는 혼합 데이터 클러스터링을 위한 제안 알고리즘을 설명한다. 먼저, 데이터 속성에 따른 K-means 기반 클러스터링 문제점을 살펴보고, 문제점을 해결하기 위한 제안 알고리즘을 제시한다. 제안된 알고리즘에 대해 (1) K-means 알고리즘에 기반한 수치 데이터에 대한 초기 클러스터 결정 및 클러스터링 방법 (2) K-means 알고리즘에 기반한 범주 데이터에 대한 초기 클러스터 결정 및 클러스터링 방법 (3) K-means 알고리즘에 기반한 혼합 데이터에 대한 초기 클러스터 결정 및 클러스터링 방법을 설명한다.

1. 수치 데이터 클러스터링 문제

1) 계층적 클러스터링 문제

계층적 클러스터링은 매 단계 유사한 클러스터만을 고려하므로 사전 클러스터의 개수를 결정할 필요는 없으나, 병합적인 방법을 이용한 계층적 클러스터링은 유사도에 따라 병합되어 하나의 클러스터로 묶여지게 되는데, 새로운 클러스터가 형성됨에 따라 클러스터간의 유사도는 점차적으로 줄어들게 되며, 분리적인 방법을 이용한 계층적 클러스터링은 어느 단계에서 알고리즘을 멈추게 할 것인가?를 결정해야 한다. 또한 거대 클러스터를 형성한 후에도 클러스터를 형성하지 못하고 남는 객체들이 있을 수 있다. 이러한 문제점은 범주 데이터에도 발생한다.

2) K-means 알고리즘 문제

분할적 클러스터링은 임의의 객체가 단지 하나의 클러스터에 포함되는 단순

클러스터링 방법이다. 분할적 클러스터링은 클러스터 구성 과정에서 클러스터에 포함되는 객체들에 대해 (재)할당이 반복적으로 발생하기 때문에 초기에 어떤 객체가 부적절하게 클러스터에 할당된다 하더라도 다른 클러스터에 포함될 수 있고, 대용량 데이터에 적합하다. 특히 K-means 알고리즘은 클러스터내의 객체들에 대한 정보를 이용한 평균을 이용하기 때문에 좋은 분할적 클러스터링 알고리즘이지만, 초기 클러스터 결정이 최종 클러스터 결과에 영향을 크게 미친다.

Table 1. K-means algorithm for cluster update of the Iris data(all attribute)

초기 클러스터				
클러스터	sepal length	sepal width	petal length	petal with
1	5.1	3.8	1.9	0.4
2	6.3	3.4	5.6	2.4
3	5.0	3.2	1.2	0.2
클러스터 갱신				
1	5.958	2.748	4.321	1.323
2	6.592	3.006	5.539	2.058
3	5.006	3.418	1.464	0.244
실제 중심				
1	5.936	2.770	4.260	1.326
2	6.588	2.974	5.552	2.026
3	5.006	3.418	1.464	0.244

Table 2. K-means algorithm for cluster update of the Iris data (choice attribute)

초기 클러스터			
클러스터	1	2	3
속성 1	6.3	6.0	2.5
속성 3	4.9	3.3	1.0
속성 4	5.8	5.1	1.9
클러스터 갱신			
속성 1	5.952	4.337	1.338
속성 3	6.626	5.574	2.074
속성 4	5.006	1.464	0.244
실제 중심			
속성 1	5.936	4.260	1.326
속성 3	6.588	5.552	2.026
속성 4	5.006	1.464	0.244

Table 3. K-means algorithm for cluster update of the Wine data(all attribute)

초기 클러스터

클러스터	1	2	3
속성 1	12.752	12.414	13.609
속성 2	2.644	2.248	2.058
속성 3	2.343	2.296	2.433
속성 4	20.482	20.815	17.666
속성 5	96.875	91.821	107.433
속성 6	1.922	2.265	2.714
속성 7	1.347	2.089	2.727
속성 8	0.412	0.375	0.300
속성 9	1.372	1.549	1.850
속성 10	5.281	3.733	5.589
속성 11	0.871	0.993	1.029
속성 12	2.198	2.668	3.022
속성 13	613.528	401.615	1091.194

클러스터 갱신

속성 1	12.929	12.517	13.804
속성 2	2.504	2.494	1.883
속성 3	2.408	2.289	2.423
속성 4	19.890	20.823	17.023
속성 5	103.597	92.348	105.511
속성 6	2.111	2.071	2.867
속성 7	1.584	1.758	3.014
속성 8	0.388	0.390	0.285
속성 9	1.503	1.452	1.910
속성 10	5.650	4.087	5.703
속성 11	0.884	0.941	1.078
속성 12	2.365	2.491	3.114
속성 13	728.339	458.232	1195.149

실제 중심

속성 1	13.154	12.279	13.745
속성 2	3.334	1.933	2.011
속성 3	2.437	2.245	2.456
속성 4	21.417	20.238	17.037
속성 5	99.313	94.549	106.339
속성 6	1.679	2.259	2.840
속성 7	0.181	2.081	2.982
속성 8	0.448	0.364	0.290
속성 9	1.154	1.630	1.899
속성 10	7.396	3.087	5.528
속성 11	0.683	1.056	1.062
속성 12	1.684	2.785	3.158
속성 13	629.896	519.507	1115.712

Table 4. K-means algorithm for cluster update of the Wine data (choice attribute)

초기 클러스터			
클러스터	1	2	3
속성 4	20.321	20.648	17.519
속성 5	100.635	91.500	106.610
속성 13	665.381	433.893	1131.017
클러스터 갱신			
속성 4	19.890	20.823	17.023
속성 5	103.597	92.348	105.511
속성 13	728.339	458.232	1195.149
실제 중심			
속성 4	21.417	20.238	17.037
속성 5	99.313	94.549	106.339
속성 13	629.896	519.507	1115.712

Table 1과 Table 2는 Iris 데이터를 K-means 알고리즘에 적용한 결과이다. Iris 데이터는 150개 객체와 4개의 속성으로 이루어져 있으며, 실제 클러스터 결과는 3개로 Setosa: 50, Versicolour: 50, Virginica: 50으로 이루어져 있다. Table 1의 결과는 모든 속성을 이용한 클러스터와 클러스터링에 대한 최종 중심결과이고, Table 2의 결과는 분산이 가장 큰 속성들을 선택한 클러스터와 클러스터링에 대한 최종 중심결과이다.

Table 3과 Table 4는 Wine 데이터를 K-means 알고리즘에 적용한 결과이다. Wine 데이터는 178개 객체와 13개의 속성으로 이루어져 있으며, 실제 클러스터 결과는 3개로 class 1: 59, class 2: 71, class 3: 48이다. Table 3의 결과는 모든 속성을 이용한 클러스터와 클러스터링에 대한 최종 중심결과이고, Table 4의 결과는 분산이 가장 큰 속성들을 선택한 클러스터와 클러스터링에 대한 최종 중심결과이다.

Table 1~Table 4의 결과를 볼 때 K-means 알고리즘은 클러스터 결과에 가장 큰 영향을 미치는 중요한 속성(분산이 가장 큰 속성)을 선택하여 클러스터링 한 결과가 모든 속성을 사용하여 클러스터링 한 결과와 큰 차이를 보이지 않고 있으며, 중요한 속성의 선택으로 초기 클러스터 결정에 영향을 주는 속성을 선택하게 되므로, 클러스터 결과의 정확성이 높아지며, 계산양 및 저장 공간을 줄일 수

있다. 따라서 제안하는 수치 데이터에 대한 클러스터링 알고리즘은 클러스터링에 영향이 큰 속성을 선택하여 K-means 알고리즘에 적용한다.

2. 범주 데이터 클러스터링 문제

1) 도메인과 속성

$A_1, A_2, ..., A_m$을 m개의 범주형 속성이라 하고, DOM(A_1), DOM(A_2),…, DOM(A_m)을 속성의 도메인이라 하자. 도메인 DOM(A_j)는 순서(order)가 없는 유한개의 범주로 정의된다.

Table 5. An instance of categorical example

Attribute / Object	Categorical 1	Categorical 2	Categorical 3	Cluster
Object 1	A	a	X	C1
Object 2	B	a	X	C1
Object 3	C	b	Y	C2
Object 4	C	c	Y	C2
Object 5	D	c	Z	C2
Object 6	D	b	Z	C2

Table 5는 각 데이터 객체가 3차 속성({Categorical 1, Categorical 2, Categorical 3})을 갖고 있다.

Categorical 1 도메인은 {A, B, C} 3개의 범주 값, Categorical 2 도메인은 {a, b, c} 3개의 범주 값, Categorical 3 도메인은 {X, Y, Z} 3개의 범주 값을 갖는다. 이러한 범주 데이터의 유사도는 수치 데이터와는 달리 자연적으로 정의될 수 없다.

2) K-modes 알고리즘 문제

Figure 9는 클러스터의 모드와 객체 사이의 유사도 측정에 대한 예이다. {C, b, Y}와 {C, c, Y}는 모드 {A, a, X}에 대해 유사도가 3으로 나타나고, 모드 {C, c,

Y}에 대해서는 유사도가 1로 계산되어, {C, b, Y}와 {C, c, Y}는 {C, c, Y} 클러스터로 할당된다.

Attribute / Object	Categorical 1	Categorical 2	Categorical 3
Object 1	A	a	X
Object 2	B	a	X
Object 3	C	b	Y
Object 4	C	c	Y
Object 5	D	c	Z
Object 6	D	b	Z

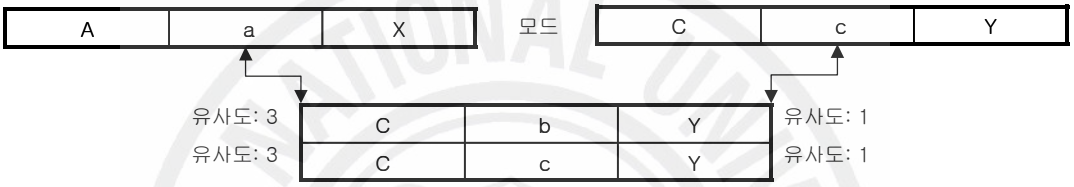

Figure 9. Example of the similarity between object and cluster(true case)

Attribute / Object	Categorical 1	Categorical 2	Categorical 3
Object 1	A	a	X
Object 2	B	a	X
Object 3	C	b	Y
Object 4	C	c	Y
Object 5	D	c	Z
Object 6	D	b	Z

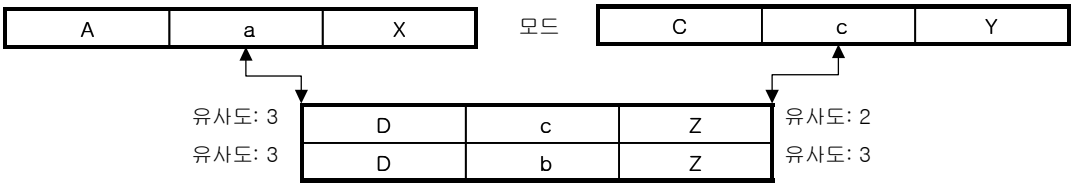

Figure 10. Example of the similarity between object and cluster(false case)

Figure 10은 클러스터의 모드와 객체 사이의 유사도 측정에 대한 예이다. {D, c, Z}와 {D, b, Z}는 모드 {A, a, X}에 대해 유사도가 3으로 나타나고, 모드 {C,

c, Y}에 대해서는 유사도가 2와 3으로 계산되어, {C, c, Y} 클러스터로 할당되지 못한다.

이와 같이 K-means paradigm에 기반한 K-modes 알고리즘은 해당 클러스터를 대표하는 모드를 선택할 때 클러스터 안에 포함되어 있는 데이터들의 값을 골고루 반영하지 못하기 때문에 클러스터링 결과가 나쁠 수밖에 없다.

이러한 문제를 해결하기 위해서는 클러스터에 속한 데이터 객체의 정보를 최대한 사용하여 클러스터를 대표하는 값으로 지정해 주어야 한다.

수치 데이터를 대상으로 하는 K-means 알고리즘에서는 클러스터에 포함되어 있는 모든 객체들에 대한 평균을 클러스터의 중심으로 사용하기 때문에 모든 객체들이 균등하게 반영된다. 반면 K-modes 알고리즘에서는 범주 데이터를 객체로 사용하기 때문에 클러스터의 대표로 평균을 구할 수 없고, 클러스터별로 각각의 차원(속성)에서 가장 많은 수를 차지하고 있는 값인 모드를 클러스터의 중심으로 사용한다.

K-modes 알고리즘은 K-means 알고리즘을 범주 데이터에 맞게 변형한 형태이다. 그러나 K-modes 알고리즘은 현재 클러스터를 구성하고 있는 여러 데이터들의 모든 성질을 균등하게 반영하여 클러스터를 대표하는 모드를 선택하지 않았기 때문에 해당 모드가 클러스터를 대표한다고는 할 수 없다

K-modes 알고리즘은 클러스터의 중심인 모드 값이 클러스터를 대표할 수 없기 때문에 클러스터링 결과에 대한 정확성이 떨어진다. K-priority 알고리즘은 클러스터의 대표 값을 찾을 때 K-modes 알고리즘과 같이 차원별로 하나의 선택 값으로 해당 클러스터를 대표하도록 하지 않고, 차원별로 클러스터의 특성에 영향을 줄만한 모든 선택 값을 반영한다.

클러스터의 대표값을 찾는 방법은 클러스터에 포함되어 있는 모든 객체들의 정보를 반영하여 프로파일이란 요약정보를 생성하고, 프로파일에서 클러스터의 특성에 영향을 주는 선택 값만을 선별하여 priority를 생성하는 것이다.

프로파일 생성은 다음과 같다.

(a) 각 차원(속성)별로 선택 값(범주 값)들을 나열하여 선택 값들이 모두 표현되는 표를 생성한다.

(b) n개의 데이터를 각 차원별로 탐색하여 각 차원이 갖고 있는 값에 해당하는 (a)의 항목을 찾아 값을 1씩 증가시켜 나간다.

프로파일을 사용하여 클러스터를 대표하게 되면 K-modes 알고리즘처럼 차원별로 하나의 선택 값(단일 중심)을 사용할 때보다 클러스터를 표현하는 정도가 향상되기 때문에 보다 나은 클러스터링 결과를 기대할 수 있다. 하지만 클러스터 안의 모든 데이터가 거리 측정에 사용되기 때문에 클러스터의 특성에 영향을 미치는 선택 값까지 모두 포함되어 클러스터를 해석하는데 노이즈가 생길 수 있다. 중요하지 않은 선택 값들의 제거여부에 따라 클러스터와 데이터 객체 사이의 거리 측정값이 변할 수 있게 된다.

priority의 경우 클러스터의 특성에 영향력이 적은 선택 값이 제거되었기 때문에 클러스터로 할당할 수 있는 가능성은 적어지지만, 프로파일을 사용하는 경우는 해당 클러스터로 할당될 가능성이 생기게 된다. 즉 클러스터링에 큰 영향력이 없는데 프로파일에 존재하기 때문에(프로파일에 존재하는 것은 클러스터링에 조금이라도 영향이 있다는 것을 의미) 선택 값이 많아지게 되므로 다음번 프로파일 갱신 시점에서 해당 선택 값은 클러스터의 특성에 영향을 주게 된다. 따라서 이러한 영향을 없애기 위해서는 클러스터의 특성에 영향력이 큰 선택 값들을 우선적으로 채택하여 유사도를 측정해야 한다.

각각의 클러스터가 해당 클러스터에 속해 있는 모든 데이터들의 차원별 선택 값들을 사용하여 알고리즘을 수행하기 때문에 K-modes 알고리즘처럼 차원별로 하나의 선택 값만을 포함하는 알고리즘보다 공간, 시간 복잡도가 크다.

따라서 클러스터의 특성에 영향력이 적은 선택 값을 제거하고 복잡도를 줄이기 위해서 클러스터의 성질에 영향을 줄만한 차원별 주요 선택 값들만을 선별하고 클러스터의 대표 값으로 지정하여 프로파일의 크기를 축소한다.

priority를 이용한 거리 측정은 다음과 같다.

$$\sum_{i=1}^{d}\Big(\sum_{j=1}^{A_l}\delta(x_i, y_i) \times p_{ij}\Big) \Big/ \sum_{j=1}^{A_l} p_{ij} \tag{16}$$

d: 속성의 수, A_l: i번째 속성 안에서 (n/k)보다 큰 카테고리 수

p_{ij}: 프로파일, $\delta(x_i, y_i)$는 지시함수

3. 혼합 데이터 클러스터링

본 절에서는 혼합 데이터를 클러스터링 하기 위한 divide-and-conquer 방법을 제안한다. divide-and-conquer 방법은 첫 번째, 주어진 데이터에서 범주 속성을 갖는 객체와 수치 속성을 갖는 객체를 분리한다. 다음으로 분리된 데이터를 이용하여 범주 속성에 대해 클러스터링 한다. 범주 속성에 대한 클러스터링은 K-priority(Figure 14) 및 수정된 K-modes(Figure 16) 알고리즘으로 클러스터링 하고, 분리된 수치 데이터를 이용하여 수치 속성에 대해 클러스터링 한다. 수치 속성에 대한 클러스터링은 수정된 K-means(Figure 13) 알고리즘을 이용한다. 마지막으로 수치 데이터 클러스터 결과와 범주 데이터 클러스터 결과를 결합하여 클러스터링 한다.

클러스터 ensemble은 데이터를 분할하여 서로 다른 클러스터링 알고리즘으로부터 얻어지는 결과를 결합하기 위한 방법이다.

혼합 데이터를 클러스터링 하기 위한 제안 방법은 다음과 같다. 1) 주어진 데이터를 범주 속성을 갖는 객체와 수치 속성을 갖는 데이터 객체로 분리한다. 2) 분리된 범주 데이터와 수치 데이터에 대해 클러스터링 한다. 3) 수치 데이터 클러스터 결과와 범주 데이터 클러스터 결과를 결합하여 범주 데이터로 정의한 후 범주 데이터 클러스터링 한다.

Figure 11은 CEBMDC(Cluster Ensemble Based Mixed Data Clustering) 알고리즘을 구조이다. 이 알고리즘 구조는 범주 데이터와 수치 데이터에 대한 클러스터 결과를 분할하여 얻는다.

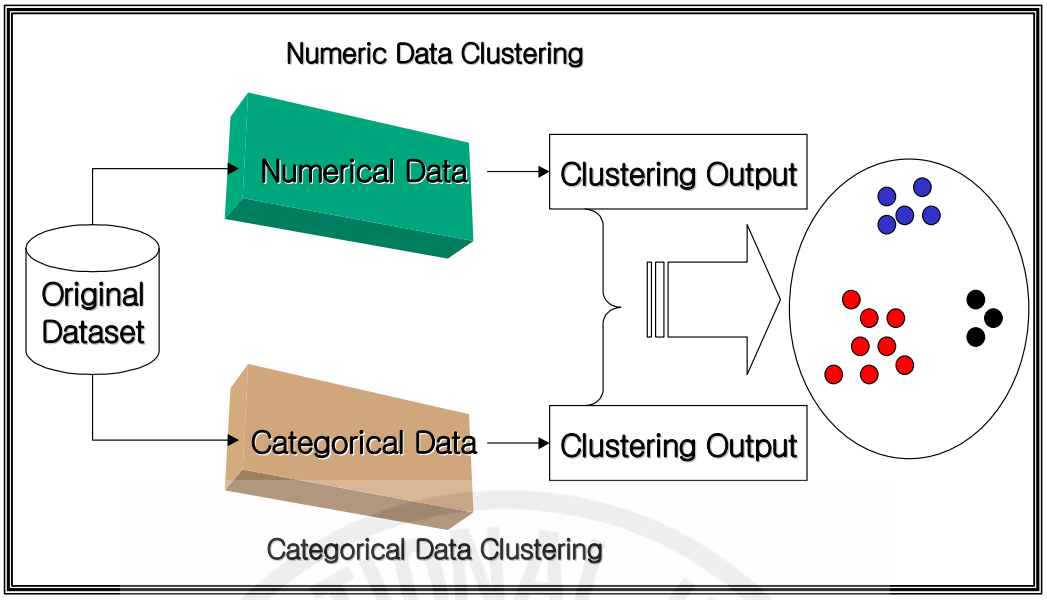

Figure 11. Overview of CEBMDC algorithm framework

1) 혼합 데이터 클러스터링 알고리즘

혼합 데이터를 클러스터링 하기 위한 알고리즘은 다음과 같다.

(1) 데이터 집합에서 범주 속성과 수치 속성으로 분할한다.
(2) 범주 데이터에 대해 클러스터링(K-priority, Modified K-modes)
(3) 수치 데이터에 대해 클러스터링(Modified K-means)
(4) 클러스터링 결과를 결합하여 범주 데이터로 정의
(5) 결합된 범주 데이터에 대해 클러스터링(K-priority, Modified K-modes)

Figure 12. Cluster Ensemble Based Mixed Data Clustering

2) 초기 클러스터 결정을 위한 Modified K-means 알고리즘

K-means 알고리즘은 클러스터 결과에 가장 큰 영향을 미치는 중요한 속성을 선택하여 클러스터링 한 결과가 모든 속성을 사용하여 클러스터링 한 결과와 큰 차이를 보이지 않으며 모든 속성을 사용하지 않으므로 계산양 및 저장 공간을

줄일 수 있다. 따라서 제안하는 수치 데이터에 대한 클러스터링 알고리즘은 클러스터링에 영향이 큰 속성을 선택하여 K-means 알고리즘에 적용한다.

K-means 알고리즘에서 초기 클러스터 결정을 위한 알고리즘은 다음과 같다.

(1) 각각의 속성별로 분산을 계산하고, k개의 속성을 선택한 다음 k개의 객체를 임의로 선택하여 클러스터의 중심으로 할당
(2) 반복
(3) 각각의 데이터 객체와 클러스터 평균과 비교하여 가장 유사한 클러스터에 해당 객체를 (재)할당
(4) 클러스터의 평균을 새로 갱신
(5) 클러스터에 변화가 없을 때까지 반복

Figure 13. Modified K-means algorithm

3) 초기 클러스터 결정을 위한 K-priority 알고리즘

범주 데이터를 클러스터링 하기 위한 초기 클러스털 결정 및 클러스터링 알고리즘은 다음과 같다.

(1) 각 속성(차원)별로 카테고리 값에 대한 프로파일을 생성
(2) k개의 객체를 임의로 선택해서 k개의 클러스터 priority를 작성
(3) 반복
(4) 각각의 객체별로 k개의 클러스터 priority와 거리 측정을 통하여 가장 유사한 클러스터에 해당 객체를 (재)할당
(5) 모든 객체들에 대해서 클러스터의 할당이 끝나면 k개의 클러스터로부터 priority를 새로 갱신
(5) 클러스터에 변화가 생기지 않을 때까지 반복

Figure 14. K-priority algorithm

범주 데이터를 클러스터링 하기 위해 각 속성별로 priority를 계산한다. 어떤 기준으로 임의의 선택 값이 다른 선택 값보다 우선적으로 클러스터의 특성에 영향력을 주는지 파악하기 위해 K-priority 알고리즘에서는 uniform 분포로 각 차원의 선택 값들이 분포하는 가정 하에 uniform 분포의 $p(x)$를 기준으로 각 차원별 임의의 선택 값들이 클러스터의 특성 표현에 우선적으로 참여할 수 있게 하였다.

4) 초기 클러스터 결정을 위한 Modified K-modes algorithm

(1) 유사도 측정

K-modes 알고리즘에서 초기 클러스터 모드를 결정하는 알고리즘을 설명하기 위해 두 객체 X_i, X_j의 유사도를 다음과 같이 정의한다.

$$d(X_i, X_j) = \frac{\sum_{j=1}^{m} \delta(X_{i1j}, X_{i2j})}{\sum_{j=1}^{m} \delta(X_{i1j}, X_{i2j}) + w} \tag{17}$$

$$\delta(a,b) = \begin{cases} 1 & , a = b \\ 0 & , a \neq b \end{cases}, \; j = 1, 2, ..., m$$

가중치 w는 $\delta = 0$일 때, 즉 동일한 속성에 대한 각 객체의 값이 일치하지 않을 때 그 속성의 (1/수준 수)들의 합이다. 즉 2개의 수준을 갖는 변수보다 3개의 수준을 갖는 변수가 서로 다를 가능성이 높기 때문에 이를 유사도에 포함하였다.

$$w = 2 \times \sum_{j=1}^{m} \frac{1}{|l_j|} \tag{18}$$

K-modes 알고리즘에서는 수준 수가 다른 변수를 수준 수가 동일한 변수로 취

급하여 비유사도를 계산하였기 때문에 이 부분을 개선하였다.

K-modes 알고리즘에서 초기 클러스터 결정을 위한 알고리즘은 다음과 같다.

(1) 모든 객체간의 유사도를 계산

(2) 유사도의 분산이 가장 큰 객체를 첫 번째 초기 모드로 결정하고, 이를 q_1^0라 하자. q_1^0와 유사도가 가장 작은 객체를 두 번째 초기 모드 q_2^0로 결정

(3) 나머지 초기 모드 q_m^0, $m = 3, 4, ..., k$를 계산하기 위해 이미 계산되어진 초기 모드를 추가하면서 나머지 객체 X_i에 대하여 다음을 계산

$$X_i \leftarrow D_i = max\{d(X_i, q_1^0), d(X_i, q_2^0), ..., d(X_i, q_{m-1}^0)\}, i = 1, 2, ..., n$$

(4) 각 객체 X_i에 대하여 얻어진 D_i를 비교하여 이들의 값을 최소로 하는 객체를 다음 초기값으로 선택

$$q_m^0 = X_q \leftarrow min_{1 \le j \le} (D_j) = D_q$$

(5) 각 객체별로 모드와 유사도를 계산하여 가장 유사한 클러스터에 객체를 할당

(6) 모든 객체들에 대해서 클러스터로 할당이 끝나면 모드를 갱신

(7) 단계 (5)를 변화가 없을 때까지 반복

Figure 15. Max-Min method for K-modes algorithm

(1) Max-Min 방법을 이용하여 초기 클러스터 모드 $\{q_1^0, q_2^0, ..., q_k^0\}$를 선택

(2) 반복

(3) 각각의 데이터 객체와 k개의 클러스터 모드와 거리 측정을 통하여 가장 유사한 클러스터에 해당 객체를 (재)할당

(4) 클러스터 모드를 새로 갱신

(5) 클러스터에 변화가 없을 때까지 반복

Figure 16. Modified K-modes algorithm

수정된 K-modes 알고리즘은 초기 클러스터 모드 결정을 위해 Max-Min 방법을 이용하여 초기 클러스터를 결정하고, K-means paradigm을 적용한 알고리즘이다.

다음 장에서는 제안 알고리즘에 대한 성능을 평가한다.

V. 실험결과

1. 실험환경

실험은 Microsoft Windows XP Professional, Pentium(R) 4, 3.00GHz, 512MB RAM 환경에서 공개 소프트웨어 통계 패키지인 R-project를 사용하여 알고리즘을 구현하였다.

2. 실험 데이터

본 절에서는 클러스터링 알고리즘의 실험에 빈번하게 사용되는 UCI Machine Learning Repository(http://mlearn.ics.uci.edu/databases/)의 데이터를 사용하였으며 그 구성은 다음과 같다.

Table 6. Experimental data

데이터 속성	사용 데이터
수치	Iris, Wine
범주	Small Soybean, Mushroom
혼합	Heart disease

실험에 사용된 데이터는 속성의 성분을 수치화하였고, 결측이 포함되는 경우는 결측 처리하였다.

3. 실험결과

1) 평가방법

제안한 알고리즘을 평가하기 위한 방법으로 클러스터 결과와 정밀도 측정을 위한 micro-p를 사용하였다.

$$micro-p = \sum_{l=1}^{k} a_l / n \tag{19}$$

n: 데이터 객체, k: 클러스터 수

2) 혼합 데이터 클러스터링 실험결과

(1) Heart disease data

Heart disease 데이터는 303개의 객체와 14개의 속성을 갖는 혼합 데이터이며 5개의 수치 데이터와 9개의 범주 데이터로 구분된다. 2개의 클러스터를 구성하고 있으며, 첫 번째 클러스터(Normal)은 164개 이고, 두 번째 클러스터(Heart patient)은 139개 이다. 수치 데이터에는 결측이 없고, 범주 데이터 12번째 속성과 13번째 속성에 결측이 존재하여 결측 처리하였다.

Table 7. Cluster recovery result for Heart disease data with proposed algorithm(accuracy)

Algorithm	No. of data objects in desired clusters	micro-p
K-priority	247	0.82
Modified K-modes	211	0.70

Table 7은 5장에서 제안한 수정된 K-modes 알고리즘과 K-priority 알고리즘

에 의한 Heart disease 데이터 클러스터링 결과이다. K-priority에 의한 알고리즘이 수정된 K-modes 알고리즘에 비해 높은 정밀도를 나타내었다.

이 결과는 수정된 K-modes 알고리즘에서 초기 클러스터를 결정함에 있어 한 곳에 모이는 현상을 방지하고, 적절하게 떨어져서 선택되도록 초기 클러스터 모드를 결정하더라도 단일 차원의 중심을 사용하는 문제 때문에 정밀도가 낮게 나타났다.

3) 수치 데이터 클러스터링 실험결과

(1) Iris data

Iris 데이터는 150개의 객체와 4개의 수치 속성을 갖는 데이터이며, 3개의 클러스터를 구성하고 있으며, 첫 번째은 클러스터 50개 이고, 두 번째 클러스터는 50개, 세 번째 클러스터는 50개 이며 결측값은 없다.

Table 8. Cluster recovery result for Iris data set with the proposed algorithm

Cluster No.	Setosa	Versicolour	Virginica
1	50	0	0
2	0	50	2
3	0	0	48

Table 9. Cluster recovery result for Iris data set with the K-means algorithm

Cluster No.	Setosa	Versicolour	Virginica
1	50	0	0
2	0	50	12
3	0	0	38

Table 10. Comparison of different clustering algorithms for Iris data set(best)

Algorithm	No. of data objects in desired clusters	micro-p
Modified K-means	148	0.98
K-means(best)	138	0.92

Table 11. Comparison of different clustering algorithms for Iris data set(average)

Algorithm	No. of data objects in desired clusters	micro-p
Modified K-means	147	0.98
K-means(average)	115	0.77

Table 8은 제안된 알고리즘에 의한 Iris 데이터 클러스터 결과이고, Table 9는 K-means 알고리즘에 의한 Iris 데이터 클러스터 결과이다. 클러스터 결과에 대한 정밀도 측정결과 K-means 알고리즘은 평균 정밀도가 0.77이고, 제안된 방법은 평균 정밀도가 0.98로 나타났다(Table 11).

(2) Wine data

Wine 데이터는 178개의 객체와 13개의 수치 속성을 갖는 데이터이며, 3개의 클러스터를 구성하고 있으며, 첫 번째 클러스터는 59개 이고, 두 번째 클러스터는 71개, 세 번째 클러스터는 48개 이며, 결측값은 없다.

Table 12. Cluster recovery result for Wine data set with the proposed algorithm

Cluster No.	Class1	Class2	Class3
1	58	4	0
2	1	67	1
3	0	0	47

Table 13. Cluster recovery result for Wine data set with the K-means algorithm

Cluster No.	Class1	Class2	Class3
1	58	4	0
2	1	67	1
3	0	0	47

Table 14. Comparison of different clustering algorithms for Wine data set(best)

Algorithm	No. of data objects in desired clusters	micro-p
Modified K-means	172	0.97
K-means (best)	172	0.97

Table 15. Comparison of different clustering algorithms for Wine data set(average)

Algorithm	No. of data objects in desired clusters	micro-p
Modified K-means	154	0.87
K-means (average)	136	0.76

Table 12는 제안된 방법에 의한 Wine 데이터 클러스터 결과이고, Table 13은 K-means 알고리즘에 의한 Wine 데이터 클러스터 결과이다. 클러스터 결과에 대한 정밀도 측정결과 K-means 알고리즘은 평균 정밀도가 0.76이고, 제안된 방법은 평균 정밀도가 0.87로 나타났다(Table 15).

4) 범주 데이터 클러스터링 실험결과

(1) Small Soybean data

Small Soybean 데이터는 47개의 객체와 35개의 범주 속성을 갖는 데이터이며, 4개의 클러스터를 구성하고 있으며, 첫 번째 클러스터(D)은 10개, 두 번째 클러

스터(C)은 10개, 세 번째 클러스터(R)은 10개, 네 번째 클러스터(P)은 17개 이며, 결측값은 없다.

Table 16. Cluster recovery result for Small Soybean data set with the proposed algorithm

Cluster No.	D	C	R	P
1	10	0	0	0
2	0	10	0	0
3	0	0	10	1
4	0	0	0	16

Table 17. Cluster recovery result for Small Soybean data set with the K-modes algorithm

Cluster No.	D	C	R	P
1	10	1	2	1
2	0	9	1	0
3	0	0	7	0
4	0	0	0	16

Table 18. Comparison of different clustering algorithm for Small Soybean data set(best)

Algorithm	No. of data objects in desired clusters	micro-p
K-priority	46	0.98
K-modes (best)	42	0.89

Table 19. Comparison of different clustering algorithms for Small Soybean data set(average)

Algorithm	No. of data objects in desired clusters	micro-p
K-priority	44	0.94
K-modes (average)	41	0.87

Table 16은 제안된 방법에 의한 Small Soybean 데이터 클러스터 결과이고,

Table 17은 K-modes 알고리즘에 의한 Small Soybean 데이터 클러스터 결과이다. 클러스터 결과에 대한 정밀도 측정결과 K-modes 알고리즘은 평균 정밀도가 0.87이고, 제안된 방법은 평균 정밀도가 0.94로 나타났다(Table 19).

(2) Mushroom data

Mushroom 데이터는 8124개의 객체와 22개의 범주 속성을 갖는 데이터이며, 2개의 클러스터를 구성하고 있으며, 첫 번째 클러스터는 4208(edible)개 이고, 두 번째 클러스터는 3916(poisonous)개이다. 결측이 존재하여 결측 처리하였다.

Table 20. Cluster recovery result for Mushroom data set with the proposed algorithm

Cluster No.	edible	poisonous
1	3961	494
2	247	3422

Table 21. Cluster recovery result for Mushroom data set with the K-modes algorithm

Cluster No.	edible	poisonous
1	3352	1284
2	856	2632

Table 22. Comparison of different clustering algorithms for Mushroom data set(best)

Algorithm	No. of data objects in desired clusters	micro-p
K-priority	7383	0.91
K-modes (best)	6029	0.74

Table 23. Comparison of different clustering algorithms for Mushroom data set(average)

Algorithm	No. of data objects in desired clusters	micro-p
K-priority	7072	0.87
K-modes (average)	5984	0.74

Table 24. Comparison of different clustering algorithms for Mushroom data set(execution time)

Algorithm	average time	minimum time	maximum time
K-priority	153.57	61.11	308.16
K-modes	139.93	50.85	218.89

Table 20은 제안된 방법에 의한 Mushroom 데이터 클러스터 결과이고, Table 21은 K-modes 알고리즘에 의한 Mushroom 데이터 클러스터 결과이다. 클러스터 결과에 대한 정밀도 측정결과 K-modes 알고리즘은 평균 정밀도가 0.74이고, 제안된 방법은 평균 정밀도가 0.87로 나타났다(Table 23).

V. 결론

대용량 데이터들은 다양한 속성을 갖는다. 수치 데이터와 범주 데이터가 혼합되어 있는 경우 범주 데이터와 수치 데이터를 동시에 고려해야한다.

범주 데이터 클러스터링 문제는 클러스터 ensemble 관점에서 최적화 문제로 정의되고, 범주 데이터를 클러스터링 하기 위해 클러스터 ensemble 접근을 적용할 수 있으며, 클러스터 ensemble 문제를 범주 데이터와 수치 데이터에 대한 멀티 클러스터링 방법으로 고려하여 클러스터링 결과를 이루는 불안정성을 극복하는 클러스터링 결과는 적합한 결과로 결합된다고 할 수 있다. 이에 본 논문에서는 대용량 데이터에 적합한 클러스터링 방법인 K-means 기반 분할적 클러스터링을 고려하였고, 문제점을 해결하기 위한 알고리즘을 제안하였다. K-modes 알고리즘에서 초기 클러스터 모드를 결정하기 위해 5장에서는 수정된 K-modes 알고리즘과 K-modes 알고리즘이 단일의 중심(모드)을 사용하여 발생하는 문제를 해결하기 위한 5장에서 수정된 K-modes 알고리즘과 K-priority 알고리즘을 제안하였다.

수치 데이터를 분할적 클러스터링 하는 K-means 알고리즘은 대용량 데이터에 효율적인 특징을 갖고 있고, 클러스터에 포함되어 있는 모든 객체들에 대한 정보를 이용한 평균을 클러스터의 중심으로 사용하기 때문에 클러스터 내부의 모든 객체들이 균등하게 반영할 수 있는 장점이 있으나, 클러스터의 중심을 평균으로 사용하기 때문에 잡음 또는 이상값에 민감하고, 초기 클러스터 결정에 따라 클러스터 결과가 달라지게된다.

PAM 알고리즘은 실제 객체인 메도이드를 클러스터의 중심으로 사용하여 이상값 문제를 해결하였으나 대용량 데이터를 처리하는데는 비효율적이며, CLARA 알고리즘 및 CLARANS 알고리즘은 대용량 데이터를 처리할 수 있으나, 부적절한 샘플링 결과는 최적의 메도이드를 찾아낼 수 없다. 이에 분할적 클러스터링 방법 중 하나인 K-means 알고리즘을 이용하여 수치 데이터를 클러스터링 하기 위해 5장에서 초기 클러스터 결정을 위한 수정된 K-means 알고리즘을 제안하였

고, 실험결과 기존 K-means에 비해 정밀도가 높게 나타나고 있음을 확인할 수 있었다.

범주 데이터는 수치 데이터와는 달리 클러스터를 구성할 경우 클러스터 중심(대표값)을 나타내기가 쉽지 않다.

범주 데이터를 분할적으로 클러스터링하는 K-modes 알고리즘은 K-means 알고리즘을 범주 데이터에 적합하도록 확장한 알고리즘으로 대용량 데이터에 효율적이며, 클러스터별로 각각의 속성에서 가장 많은 수를 차지하고 있는 값을 모드라 하여 그 클러스터의 중심으로 사용한다. 그러나 K-modes 알고리즘은 단일의 중심(모드)을 사용하여 현재 클러스터를 구성하고 있는 여러 데이터들의 모든 성질을 균등하게 반영하는 클러스터를 대표하는 모드를 선택하지 않았기 때문에 해당 모드가 클러스터를 대표한다고는 할 수 없다. 또한 K-modes 알고리즘은 K-means paradigm을 범주 데이터로 확장한 알고리즘이므로 K-means 알고리즘에서 발생하는 초기 클러스터 결정 문제가 발생한다. 그리고, 단일의 중심(모드)을 사용하기 때문에 해당 모드가 클러스터를 대표한다고 할 수 없다. 이러한 문제를 해결하기 위해 5장에서 수정된 K-modes 알고리즘과 K-priority 알고리즘을 제안하였다.

K-priority 알고리즘은 각 클러스터별 프로파일을 이용하여 개별 클러스터를 표현한다. 프로파일에서 클러스터의 특성에 영향력을 미치는 주요 차원별 범주값을 선택하여 priority를 생성하고, 선택된 값들을 이용하여 해당 클러스터와 객체 사이의 거리를 측정하여 클러스터링 하게 된다.

본 논문에서는 수치 데이터와 범주 데이터가 혼합되어 있는 클러스터링 방법에 대한 알고리즘을 제안하였다. 수치 데이터와 범주 데이터를 분리하여 수치 데이터에는 수정된 K-means 알고리즘을 적용하였고, 범주 데이터에는 수정된 K-modes 알고리즘과 K-priority 알고리즘을 적용하였으며, 클러스터링 결과 K-priority 알고리즘이 정밀도가 높게 나타났다.

실험결과 초기 클러스터 모드를 결정하기 위한 수정된 K-modes 알고리즘이 K-priority 알고리즘에 비해 정밀도가 떨어졌다. 수정된 K-modes 알고리즘에서 초기 클러스터를 결정함에 있어 한 곳에 모이는 현상을 방지하고, 적절하게 떨어져서 선택되도록 초기 클러스터 모드를 결정하더라도 단일 차원의 중심을 사용

하는 문제 때문에 정밀도가 다소 낮게 나타났으며, 랜덤하게 객체를 선택한 후 priority를 사용한 결과 클러스터에 영향을 적게 주는 속성 값은 클러스터링에서 제외되어 시간 및 공간 복잡도가 절약되며 클러스터 결과의 정밀도가 수정된 K-modes에 비해 높게 나타났다. 따라서 혼합 데이터를 클러스터링 하기 위해 수치 데이터와 범주 데이터로 분류하여 클러스터링된 결과를 결합하여 얻어진 결과를 K-priority 알고리즘을 적용하여 클러스터링 하였다.

실험결과에서 볼 수 있듯이 수치 데이터인 Iris, Wine 데이터에 대한 클러스터링 결과 기존 K-means 방법에 비해 높은 정밀도를 나타내었고, 범주 데이터인 Small Soybean, Mushroom 데이터에 대해서도 높은 정밀도를 나타내었다. 혼합 데이터인 Heart disease 데이터에 대한 클러스터링 결과는 수치 데이터에 대한 클러스터 결과와 범주 데이터에 대한 클러스터 결과를 결합하여 범주 데이터로 정의하고, 범주 데이터에 대한 K-priority 알고리즘으로 클러스터링 결과 82%의 정밀도를 나타내었고, 수정된 K-modes 알고리즘은 70%의 정밀도를 나타내었다.

대용량 데이터는 수치 속성 및 범주 속성이 혼합되어 있으며 이러한 데이터를 클러스터링 하기 위해서는 각 데이터 속성에 맞는 초기 클러스터를 결정 및 클러스터링 방법을 적용하여야 한다. 또한 대용량 처리가 쉽고, 클러스터 결과 해석이 쉬어야 한다.

본 논문에서 제안한 알고리즘은 수치 데이터와 범주 데이터에 대해 다른 변환이 필요가 없고, 직접 알고리즘을 수행할 수 있기 때문에 대용량 데이터에도 적합하며, 클러스터 결과 해석이 용이하다.

참고문헌

A.K. JAIN, M.N. MURTY and P.J. FLYNN, 1999, Data Clustering: A Review, *ACM Computing Surveys*, Vol. 31, No. 3, September.

Amir Ahmad, Lipika Dey, 2007, A K-means clustering algorithm for mixed numeric and categorical data, *Data & Knowledge Engineering*, Vol. 63, No2, pp. 503-527.

A. Strehl ,J. Ghosh, 2002a, Cluster ensembles-a knowledge reuse framework for combining partitions, *in: Proc. of the 8th National Conference on Artificial Intelligence and 4th Conference on Innovative Applications of Artificial Intelligence*, pp. 93-99.

A. Strehl, 2002b, Relationship-based clustering and cluster ensembles for high dimensional data mining, phD thesis, *The University of Texas at Austin*, May

Cen Li, Gautam Biswas, 2002, Unsupervised Learning with Mixed Numeric and Nominal Data, *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, VOL. 14, NO. 4, pp. 673-690.

Forgy, E., 1965. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. Biometrics 21, 768.

H. Ralambondrainy, 1995, A Conceptual Version of the k-Means Algorithm *Pattern Recognition Letters*. 16, pp. 1147-1157.

Hee-Chang Park, Jee-Hyun Ryu, 2005, Clustering Algorithm Using a Center of Gravity for Grid-based Sample, *Journal of Korean Data & Information Science Society*, Vol. 16, No2. pp. 217~226.

J.A. Hartigan, 1974, Clustering Algorithms, *John Wiley & Sons, New York*.

Jiawei Han and Micheline Kamber, 2001, Data Mining Concepts and Techniques. *Morgan kaufmann publishers*. pp. 354-376.

Jiawei Han and Micheline Kamber, 2001, Data Mining Concepts and Techniques. *Morgan kaufmann publishers*. pp. 352-353.

J.M. Pe˜na , J.A. Lozano, P Larra˜naga, 1999, An empirical comparison of four initialization methods for the K-Means algorithm. *Pattern Recognition Letters,* 20, pp. 1027-1040.

L. Kaufman and P.J. Rousseeuw, 1990, Finding Groups in Data: an Introduction to Cluster Analysis. *John Wiley & Sons*.

Macqueen, J.B, 1967, Some methods for classification and analysis of multi variate observations. *Proceedings of the Symposium on Mathematical Statistics and Probability, 5th, Berkely*, Vol 1, 281-297.

Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu, 1996, A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, pp. 226-231.

Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu, 1997, Density Connected Sets and their Application for Trend Detection in Spatial Databases, *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining*, pp. 10-15.

Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, Prabhakar Raghavan, 1998, Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pp. 94-105.

Raymond T. Ng, Jiawei Han, 2002, CLARANS: A Method for Clustering Objects for Spatial Data Mining, *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, VOL, 14, NO. 5, pp.1003-1016.

R.T. Ng and J. Han, 1994, Efficient and effective clustering methods for spatial data mining, *In Proc. Int. Conf. on Very Large Data Bases (VLDB'94)*, pp. 144-155.

Shehroz S. Khan, Amir Ahmad, 2004, Cluster center initialization algorithm for K-means clustering, *Pattern Recognition Letters* 25, pp. 1293-1302.

Sudipto Guha, Rajeev Rastogi, Kyuseok Shim, 1999, ROCK: a robust clustering algorithm for categorical attributes. *Proceedings of the IEEE International Conference on Data Engineering, Sydney*,

Tian Zhang, Raghu Ramakrishnan, Miron Livny, 1996, BIRCH: An Efficient Data Clustering Method for Very Large Databases, *ACM SIGMOD Int. Conf. on Management of Data*, pp. 103-114.

TIAN ZHANG, RAGHU RAMAKRISHNAN, MIRON LIVNY, 1997, BIRCH: A New Data Clustering Algorithm and Its Applications, *Data Mining and Knowledge Discovery*, 1, pp. 141-182.

Wei Wang, Jiong Yang, Richard Muntz, 1997, STING: A Statistical Information Grid Approach to Spatial Data Mining. *Proc. 23rd Int. Conf. on Very Large Data Bases*, pp. 186-195.

Whasoo Bae and Se Won Roh, 2005, A Study on K-Means Clustering, *The Korean Communications in Statistics* Vol. 12, No. 2, pp. 497-508.

Ying Sun, Qiuming Zhu, Zhengxin Chen, 2002, An iterative initial-points refinement algorithm for categorical data clustering, *Pattern Recognition Letters* 23, pp. 875-884.

Zengyou He, Xiaofe I Xu, Shengchun Deng, 2002, Clustering Mixed Numeric and Categorical Data: A Cluster Ensemble Approach.

Zengyou He, Xiaofei Xu, Shengchun Deng, 2005, A cluster ensemble method for clustering categorical data, *Information Fusion* 6, pp. 143-151

Zhexue, Huang, 1997a, A fast clustering algorithm to cluster very large categorical data sets in data mining. *Workshop on research issues on data mining and knowledge discovery*.

Zhexue Huang, 1997b, Clustering large data sets with mixed numeric and categorical values, *Proceedings of the First Pacific Asia Conference*

on Knowledge Discovery and Data Mining, World Scientific, Singapore.

김보화, 김규성, 2002, K-모드 알고리즘과 ROCK 알고리즘의 개선, *응용통계연구 제 15권 2호,* pp.381-393.

양순철, 강형창, 김철수, 2007, Initial Mode Decision Method for Clustering in Categorical Data, *Journal of Korean Data & Information Science Society* Vol. 18, pp. 481-488.

감사의 글

대학에 입학하고, 통계가 좋아 대학원에 들어선 후 오랫동안 저에게 많은 도움을 주신 분들께 지면을 빌어 감사의 말씀을 드립니다.

이 논문이 있기까지 많은 격려를 아끼지 않으시며 지도해 주신 김철수 교수님께 지면을 빌어 깊은 감사를 드립니다. 또한 학부시절부터 관심과 배려로 보살펴 주셨던 김익찬 교수님, 이봉규 교수님, 이정훈 교수님, 박경린 교수님께 감사드리며, 바쁘신 중에도 논문심사를 맡아주신 이동철 교수님, 이윤정 교수님을 비롯한 심사위원 분들께 감사하다는 말씀을 드립니다.

서로 연구한 기간은 짧았지만 저와 함께 많은 얘기를 나누었던 강정석 선생님께도 감사하다는 말씀드립니다. 힘들고 지칠 때 위로해주던 친구들과 옆에서 도움을 줬던 학과조교 들에게도 고맙다는 말을 전합니다. 그리고 논문 막바지 힘든 일이 있었을 때 옆에서 위로해주고 지켜주었던 원효와 성건에게 고맙다는 말을 전합니다.

무엇보다도 저를 위해 걱정하고, 든든한 힘이 되어준 아버지, 어머니, 그리고 동생, 든든한 가족들이 있었기에 오늘의 내가 있음을 생각하고, 감사하게 됩니다.

끝으로 이 논문을 하늘나라로 먼저 간 사랑하는 형관이에게 바칩니다.

2008년 2월

강형창 올림