ISSN 1598-0170 (Print)
ISSN 2287-1136 (Online)
http://www.jksii.or.kr

K-Means 클러스터링에서 초기 중심 선정 방법 비교

Comparison of Initial Seeds Methods for K-Means Clustering

이 신 원1*
Shinwon Lee

요 약

클러스터링 기법은 데이터에 대한 특성에 따라 몇 개의 클러스터로 군집화 하는 계층적 클러스터링이나 분할 클러스터링 등 다양한 기법이 있는데 그 중에서 K-Means 알고리즘은 구현이 쉬우나 할당-재계산에 소요되는 시간이 증가하게 된다. 또한 초기 클러스터 중심이 임의로 설정되기 때문에 클러스터링 결과가 편차가 심하다. 본 논문에서는 클러스터링에 소요되는 시간을 줄이고 안정적인 클러스터링을 하기 위해 초기 클러스터 중심 선정 방법을 삼각형 높이를 이용하는 방법을 제안하고 비교 실험해 봄으로서 할당-재계산 횟수를 줄이고 전체 클러스터링 시간을 감소시키고자 한다. 실험결과로 평균 총소요시간을 보면 최대평균거리를 이용하는 방법은 기존 방법에 비해서 17.9% 감소하였고, 제안한 방법은 38.4% 감소하였다.

☞ 주제어 : 클러스터링, 초기 중심

ABSTRACT

Clustering method is divided into hierarchical clustering, partitioning clustering, and more. K-Means algorithm is one of partitioning clustering and is adequate to cluster so many documents rapidly and easily. It has disadvantage that the random initial centers cause different result. So, the better choice is to place them as far away as possible from each other. We propose a new method of selecting initial centers in K-Means clustering. This method uses triangle height for initial centers of clusters. After that, the centers are distributed evenly and that result is more accurate than initial cluster centers selected random. It is time-consuming, but can reduce total clustering time by minimizing the number of allocation and recalculation. We can reduce the time spent on total clustering. Compared with the standard algorithm, average consuming time is reduced 38.4%

☞ keyword : K-Means, Clustering, Initial Seeds

1. 서 론

클러스터링 방법은 사용자가 요구하는 문서를 빠르고 정확하게 분석해내기 위한 방법이다. 클러스터링은 서로 관련이 있는 문서들을 클러스터로 형성하는 방법으로 사용자 질의에 대해 관련이 높은 클러스터에 있는 모든 문서를 검색 결과로 제시하여 문서에 포함되어 있는 단어들의 부합 정도가 높으면 두 문서의 유사도는 높아지고 유사도가 높은 문서들을 우선적으로 클러스터로 형성해 나가는 것이다.

데이터에 대한 특성 값에 따라 몇 개의 클러스터로 군집화 하는 클러스터링 기법은 계층적 클러스터링[1,9]이나 분할 클러스터링[6,10] 등 다양한 기법으로 나누어 설명할 수 있는데 현대 사회의 정보 대량화는 계층적 클러스터링이나 그래프 이론 클러스터링으로는 처리할 수 있는 데이터에 한계가 있고 시간 복잡도 측면에서 비효율적이다[3].

본 논문에서는 대량 데이터에 대한 클러스터링 기법으로 용이한 분할 클러스터링 중 K-Means 알고리즘을 다루고자 한다. K-Means 알고리즘은 구현이 쉽고, 패턴 수가 n일 때 시간 복잡도가 $O(n)$인 장점을 가지고 있다. 그러나 K-Means 알고리즘은 초기 클러스터 중심에 상당히 종속적이다. 즉, 초기 클러스터 중심을 어떻게 선정하는가에 따라 클러스터링 결과가 달라진다.

일반적으로 K-Means 알고리즘의 할당-재계산 과정에서 중심이 이동하면서 적절한 위치로 이동하게 된다. 그러나 초기 클러스터 중심이 어느 한쪽에 편중되어 선정되면 클러스터링 결과가 적절하지 못하게 산출되거나 할당-재계산에 소요되는 시간이 증가하게 된다. 이에 본 논

1 Department of Computer System Engineering, Jungwon University, 367-805, Korea
* Corresponding author (swlee@jwu.ac.kr)
[Received 19 September 2012, Reviewed 26 September 2012, Accepted 15 October 2012]

문에서는 초기 클러스터 중심 선정 방법으로 삼각형 높이를 이용하는 방법을 제안하고 비교 실험하여 K-Means 알고리즘의 성능을 개선하고자 한다.

본 논문에서는 초기 클러스터 중심을 임의로 잡는 방법, 초기 클러스터 중심들 간의 거리를 최대로 하는 방법과 삼각형 높이를 이용하는 방법을 비교 실험하여 K-Means 알고리즘의 성능을 개선하고자 한다. 이를 통해 초기 클러스터 중심들이 데이터 집합에 고르게 분포되도록 한다. 고르게 분포된 초기 클러스터 중심은 무작위로 선정된 초기 중심에 비해 좀 더 정확한 클러스터링 결과를 산출하게 된다. 또한 기존 알고리즘에 비해 초기 클러스터 중심 선정에 추가적인 시간이 소요되나 할당-재계산 횟수를 감소시킴으로써 전체 클러스터링 시간을 감소시킬 수 있다.

본 논문은 2장에서 K-Means 알고리즘에 대해 간략히 살펴보고 기존의 중심 선정 방법에 대해 살펴본다. 3장에서 K-Means 알고리즘을 개선하기 위한 초기 중심 선정 방법으로 삼각형 높이를 이용한 기법을 제안한다. 4장에서는 초기 중심선정 방법을 비교 실험하고 분석 및 평가를 한다. 5장에서 결론을 맺는다.

2. 관련 연구

2.1 K-Means 알고리즘

K-Means 알고리즘은 가장 일반적으로 사용되는 분할 클러스터링 알고리즘이다. 이 알고리즘의 개념은 패턴들과 그 패턴이 속하는 클러스터의 중심과의 평균 유클리디안(Euclidean) 거리를 최소화하는 것이다[4,5]. 클러스터의 중심은 그 클러스터에 속한 패턴의 평균 혹은 중심(centroid) $\vec{\mu}$라 하고 다음처럼 정의된다.

$$\vec{\mu}(\omega) = \frac{1}{|\omega|}\sum_{x \in \omega}\vec{x} \tag{1}$$

이 식에서 ω는 클러스터에 속한 패턴집합이며, $\vec{x}$는 클러스터에 속한 특정 패턴이다. 패턴은 실수 값을 가지는 벡터로 표현된다. K-Means에서 클러스터는 중력의 중심과 같이 무게 중심을 가지는 구형(sphere)으로 생각한다. 중심이 클러스터에 속한 패턴들을 얼마나 잘 표현했는가를 나타내는 척도(RSS : Residual Sum of Squares)는 각 클러스터에 속하는 모든 패턴들에 대하여 각 패턴과 중심까지의 제곱거리의 합으로 나타내며 다음 식(2)와 같다.

K-Means$(\{\vec{x_1},\ldots\vec{x_N}\}, K)$
1. $(\vec{s_1},\vec{s_2},\ldots\vec{s_K})$ <- Select Random Seeds$(\{\vec{x_1},\ldots\vec{x_N}\}, K)$
2. for $k \leftarrow 1$ to K
3. $do\ \vec{\mu_k} \leftarrow \vec{s_k}$
4. while stopping criterion has not been met
5. do for $k \leftarrow 1$ to K
6. $do\ \omega_k \leftarrow \{\}$
7. for $n \leftarrow 1$ to N
8. $do\ j \leftarrow \arg\min_j |\vec{\mu_j} - \vec{x_n}|$
9. $\omega_j \leftarrow \omega_j \cup \{\vec{x_n}\}$(벡터 재할당)
10. for $k \leftarrow 1$ to K
11. $do\ \vec{\mu_k} \leftarrow \frac{1}{|\omega_k|}\sum_{x \in \omega_k}\vec{x}$ (중심 재계산)
12. return $\{\vec{\mu_1},\ldots\vec{\mu_K}\}$

(그림 1) K-Means 알고리즘
(Figure 1) K-Means Algorithm

$$RSS_k = \sum_{x \in \omega_k} |\vec{x} - \vec{\mu}(\omega_k)|^2 \tag{2}$$
$$RSS = \sum_{k=1}^{K} RSS_k$$

RSS는 K-Means의 목적 함수이고, 이를 최소화해야 한다[2].

(그림 1)은 K-Means 알고리즘이다.

2.2 K-Means 초기값 설정

K-Means의 성능은 초기 중심을 어떻게 선정하는가에 따라 크게 달라진다. 기본적인 초기 중심은 무작위로 선정된 k개의 패턴 또는 패턴 집합 범위 내의 임의의 K개의 좌표들로 구성된다. 이렇게 설정된 초기 중심에서 출발한 클러스터링은 결과 클러스터 또한 편차가 클 수밖에 없다. 이러한 문제점을 해결하기 위해서 초기 중심 설정에 관한 많은 연구가 진행되어져 왔다.

기존의 K-Means 알고리즘은 (그림 1)에서 보는 봐와 같이 초기 중심을 선정할 때 임의로 설정하였다. [11]은 초기 클러스터 중심의 특성이 패턴 집합에 속하면서 가능한 한 공통의 속성을 갖는 패턴이라는 점에 착안하여 임의의 한 패턴을 선택하는 대신 선택된 초기 클러스터에서 색인어와 가중치로 표현되는 세 개의 문서를 선택하여 초기 클러스터 중심 벡터로 설정한다. 3배수 중심 설정의 알고리즘은 다음 식과 같다.

$$c_i^{initial} = avgbig\left(\sum_{j=1}^{3} d_j\right) \quad (3)$$

여기서 $c_i^{initial}$는 i번째 클러스터 벡터이며, d_j는 j번째 문서 벡터를 나타낸다.

[12]는 2차 평면의 4분위에 데이터를 놓고 음수 값을 가지는 속성을 양수 값으로 바꾸기 위해서 연산을 한다. 초기 값처럼 중간 값을 주고 데이터 값과 초기 값 사이의 거리를 계산하여 사용한다.

1. 데이터 값이 양수와 음수 값을 포함하고 있으면 2단계로 간다.
   그렇지 않으면 4단계로 한다.
2. 데이터 집합에서 최소 값을 찾는다.
3. 각 데이터에 대해서 최소 값을 뺀다.
4. 각 데이터에 대해서 원래 값과의 거리를 계산한다.
5. 4단계에서 얻어진 거리를 정렬한다.
6. 정렬된 데이터 값을 k개의 집합으로 분리한다.
7. 각 집합에 초기 값처럼 중간 값을 준다.
8. 각 데이터 값과 모든 초기 값과의 거리를 계산한다.

(그림 2) [12]의 초기 클러스터 중심 선정 알고리즘
(Figure 2) Initial Cluster Seeds Setting Algorithm of [12]

[13]은 초기 클러스터의 중심들이 고르게 분포되도록 하기 위해서 초기 클러스터의 중심들 간의 거리를 최대가 되도록 설정한다.

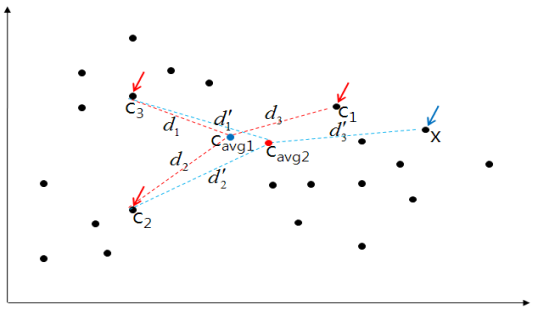
$$C = \max \sum_{i=1}^{K} \|c_{avg} - c_i\|^2 \quad (4)$$

여기서 c_i는 i번째 클러스터의 중심이며, c_{avg}는 c_1부터 c_k까지의 평균이다. 즉, c_1부터 c_k까지의 중심들이 이들의 평균으로부터 최대의 거리를 갖도록 하는 것이다.

초기 중심을 얻는 과정은 다음 (그림 3)과 같다.

1 임의의 K개의 중심을 선정한다.
2 $x \in X$에 대해
2.1 x와 가장 가까운 중심 선택
$candidate\,Cluster \leftarrow \min dist_{i=0,\ldots,k}(x, c_i)$
2.2 선택된 $candidate\,Cluster$를 기존 중심에 대체한 후 새로운 평균 거리 계산
$newDistAvg \leftarrow avg \sum_{i=1}^{k} |c_{avg} - c_i|^2$
if $c_i = candidate\,Cluster$ then
$|c_{avg} - x|$
2.3 $newDistAvg$가 기존 중심 간의 거리보다 크다면
if $newDistAvg > oldDistAvg$ then $c_i \leftarrow x$
3. $return\{c_1, \ldots c_k\}$

(그림 3) [13]의 초기 클러스터 중심 선정 알고리즘
(Figure 3) Initial Cluster Seeds Setting Algorithm of [13]



(그림 4) 최대 평균 거리를 이용한 초기 중심 이동
(Figure 4) Initial Seeds Movement using Max Average Distance

(그림 4)는 K가 3일 때, 초기 클러스터 중심 선정을 2차원 데이터를 이용해 묘사한 것이다.

기존 c_1, c_2, c_3의 3개의 중심이 있고, 새로운 데이터 x에 대해 가장 가까운 중심을 찾게 된다. c_1, c_2, c_3 각 중심과 x와의 거리를 비교한 결과 c_1임을 확인할 수 있다. 이제, c_1 대신에 x를 넣고 다음과 같이 각 중심과 평균 간의 거리($\{d_1', d_2', d_3'\}$)를 계산한다.

$$newDistAvg = \frac{1}{K} \sum_{k=1}^{K} d_j' \quad (5)$$

이 거리는 기존 중심들 간의 거리($\{d_1, d_2, d_3\}$)

$$oldDistAvg = \frac{1}{K} \sum_{k=1}^{K} d_j \quad (6)$$

3. 초기 클러스터 중심 선정 비교

3.1 최대평균거리를 이용한 알고리즘

최대평균거리를 이용한 초기 클러스터 중심 선정 방법은 초기 클러스터 중심들을 가능한 한 멀리 선정하도록 하는 방법이다.

와 비교하게 된다. 두 평균 거리를 비교한 결과 c_1 대신에 x를 대입해서 계산한 $\neq wDistAvg$값이 더 크기 때문에 x가 새로운 c_1으로 대체된다. 이제 x, c_2, c_3가 새로운 $oldDistAvg$가 되어 다음 x의 비교 대상이 된다. 이 과정을 데이터 집합X에 속한 모든 x에 대해 반복한다.

3.2 삼각형 높이를 이용한 알고리즘

삼각형 높이를 이용한 초기 클러스터 중심 선정 방법은 중심 간의 거리 대신 높이를 계산하여 높으면 중심을 대체하는 방법이다. 헤론의 공식을 이용하여 삼각형의 세 길이를 알면 높이를 구할 수 있다. 초기 클러스터 중심 집합C는 다음 식(7)과 같다.

$$C = \max \sum_{i=1}^{K} \| c_{height} - c_i \| \tag{7}$$

여기서 c_i는 i번째 클러스터의 중심이며, c_{height}는 c_1부터 c_k까지의 높이이다. 즉, c_1부터 c_k까지의 중심들이 최대 높이를 갖도록 하는 것이다.

1 임의의 K개의 중심을 선정한다.
2 $x \in X$에 대해
2.1 x와 가장 가까운 중심 선택
$candidateCluster \leftarrow \min dist_{i=0,\ldots,k}(x, c_i)$
2.2 선택된 $candidateCluster$를 기존 중심에 대체한 후 새로운 높이 계산
$newHeight \leftarrow \sqrt{c^2 - ((a^2 + c^2 - b^2)/2a)^2}$
$a = \overline{c_2, c_3}, \quad b = \overline{x, c_3} \quad c = \overline{x, c_2}$
2.3 $newHeight$가 기존 중심 간의 높이보다 크다면
if $newHeight > oldHeight$ then $c_i \leftarrow x$
3. $return\{c_1, \ldots c_k\}$

(그림 5) 삼각형의 높이를 이용한 초기 중심 선정 알고리즘
(Figure 5) Initial Seeds Setting Algorithm using Triangle Height

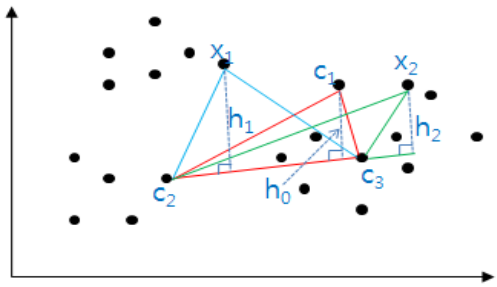
(그림 6)은 초기 클러스터 중심 선정을 K=3일 때 2차원 데이터를 이용하여 표현한 것이다.

기존 선택된 초기 중심이 c_1, c_2, c_3라고 가정할 때 새로운 데이터 x_1에 대하여 가장 가까운 중심을 찾게 된다. 위의 각 중심과 x_1와의 거리를 비교하여 가장 가까운 중심이 c_1임을 알 수 있다. c_1을 대체할 수 있는 x_1을 넣고 x_1, c_2, c_3의 높이 h_1을 계산하고 기존 중심 c_1, c_2, c_3의 높이 h_0를 계산한다.

$$newHeight \leftarrow \sqrt{C^2 - \left((a^2 + c^2 - b^2)/2a\right)^2}$$
$$a = \overline{c_2, c_3}, b = \overline{x_1, c_3}, c = \overline{x_1, c_2} \tag{8}$$

$$oldHeight \leftarrow \sqrt{C^2 - \left((a^2 + c^2 - b^2)/2a\right)^2}$$
$$a = \overline{c_2, c_3}, b = \overline{c_1, c_3}, c = \overline{c_1, c_2} \tag{9}$$

두 높이를 비교하면 h_1의 값이 크기 때문에 x_1은 c_1으로 대체될 수 있다. 그러나 x_2의 경우, x_2, c_2, c_3의 높이 h_2가 기존 초기 중심 c_1, c_2, c_3의 높이 h_0보다 작기 때문에 c_1으로 대체될 수 없다. 이 과정을 데이터 집합 X, $x_i \in$X에 대해서 반복한다.



(그림 6) 삼각형 높이를 이용한 초기 중심 이동
(Figure 6) Initial Seeds Movement using Triangle Height

3.3 시간 복잡도 평가

기존연구의 중심 선정 방법인 최대평균거리를 이용하는 방법과 본 논문에서 제안한 방법은 초기 중심을 이동시키기 위해서 계산하는 과정이 필요하다. 즉, 클러스터링에 소요되는 시간은

$$T(\text{초기중심설정}) + T(\text{할당} - \text{재계산}) \tag{10}$$

으로, K-Means의 할당-재계산 과정에 소요되는 시간 이외에 추가로 시간이 소요된다.

K는 전체 클러스터 수이고 N이 데이터 집합이고 $x_i \in$ N라고 할 때, 1회 반복 시간은 K*N이다. (그림 5)의 알고리즘을 보면 각 단계별로 필요한 소요시간을 알 수 있다.

(1) x와 가장 가까운 중심 선택에 소요된 시간 $1K$,

(2) 선택된 x를 기존 중심에 대체하고 중심들의 높이를 구하기 위한 시간 $2K$,
(3) 새로운 높이와 기존 중심 간의 비교 계산을 위한 시간 $1K$

가 소요되어 총 $4K$ 만큼의 시간이 소요된다. K는 클러스터 수이다. 기존 K-Means 알고리즘의 할당-재계산 과정의 시간 복잡도가 $O(KN)$이라고 할 때, 삼각형 높이를 계산을 위한 시간 복잡도는

$$\fallingdotseq O(4KN) \tag{11}$$

이다.

다음으로 할당-재계산 과정은 각 문서를 클러스터에 할당하기 위한 시간 1단위와 각 클러스터에 속한 문서들을 대상으로 중심을 재계산 하는데 소요되는 시간 1단위가 필요하다. 할당-재계산에 대한 수식은

$$O(2iKN) \tag{12}$$

와 같이 표현할 수 있다. 여기에서 i는 할당-재계산이 완료될 때 까지 반복되는 클러스터링 횟수이다.

따라서, 전체 클러스터링에 소요되는 시간은

$$O(4KN) + O(2iKN) \approx O(N) \tag{13}$$

이다. i와 k가 상수이기 때문에 결국 전체 클러스터링 소요 시간은 N에 선형이다. 또한

$$O(4KN) \ll O(2iKN) \tag{14}$$

이다. 두 방법 모두 초기 중심 선정에 소요되는 시간은 전체 클러스터링 소요 시간에 큰 영향을 미치지 않고 N에 비해 아주 작은 상수이기 때문에 결국 클러스터링 소요시간은 N의 크기에 비례하여 증가한다. 이는 실험을 통해 확인하도록 한다.

4. 실험 및 성능 평가

4.1 데이터집합

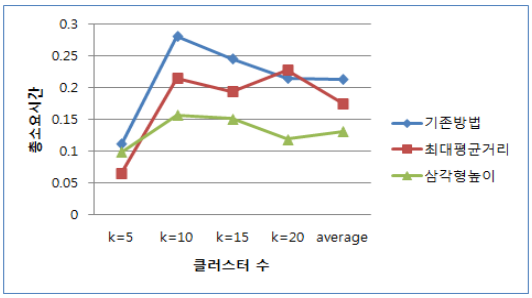
클러스터링 결과에 대한 수치적 평가를 위해 데이터를 300개, 200개로 나누어 실험을 하였고 초기 클러스터의 개수는 5, 10, 15, 20개로 하였으며, 기존 방법과 최대평균거리, 삼각형높이를 이용한 방법으로 클러스터링 횟수와 소요되는 시간을 실험하였다. 실험은 각 초기 클러스터 중심 선정 방법에 대하여 10회씩 실시하여 결과를 확인하였다.

4.2 클러스터링 횟수와 소요 시간

실험 데이터를 이용하여 클러스터링에 소요되는 시간을 측정하였다. 실험은 300개 데이터와 200개 데이터에 대해서 K값을 5~20으로 조정하며 각각을 10회씩 실행하여 평균을 구하였다. (표 1)은 200개 데이터에 대한 실험 결과를 표현한 것이며, 이 중 총소요시간에 대한 그래프가 (그림 5)이다.

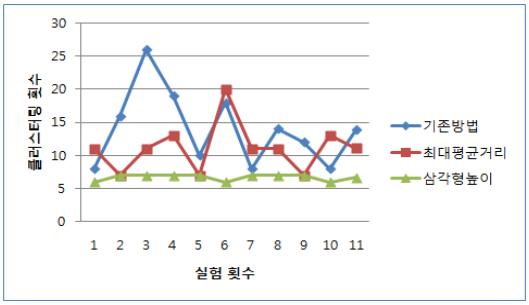
(표 1) 200개 데이터 비교 실험 결과
(Table 1) Comparison Experiment Result of 200 data

	소요시간		K=5	K=10	K=15	K=20	평균
기존방법	할당-재계산	평균	0.011828	0.01996	0.019145	0.01956	0.017623
		횟수	9.4	14.1	12.8	11	11.825
		합계	0.11118	0.28143	0.245059	0.215163	0.213208
	총소요시간		0.11118	0.28143	0.245059	0.215163	0.213208
최대평균거리	할당-재계산	평균	0.008444	0.015947	0.016116	0.018508	0.014754
		횟수	6	11.9	10.5	10.6	9.75
		합계	0.050662	0.189767	0.169217	0.196182	0.151457
	중심조정		0.014802	0.025083	0.024027	0.030818	0.023683
	총소요시간		0.065464	0.21485	0.193244	0.227	0.175139
삼각형높이	할당-재계산	평균	0.010589	0.012416	0.013154	0.012054	0.012053
		횟수	6.9	9.4	8.4	6.1	7.7
		합계	0.073061	0.116708	0.110494	0.073532	0.093449
	중심조정		0.025718	0.040122	0.040447	0.04518	0.037867
	총소요시간		0.098779	0.15683	0.150942	0.118712	0.131316



(그림 7) 200개 데이터, k=5, 10, 15, 20인 경우 총소요시간 비교 실험

(Figure 7) Comparison Experiment of total Spending Time in K=5, 10, 15, and 20, using 200 data

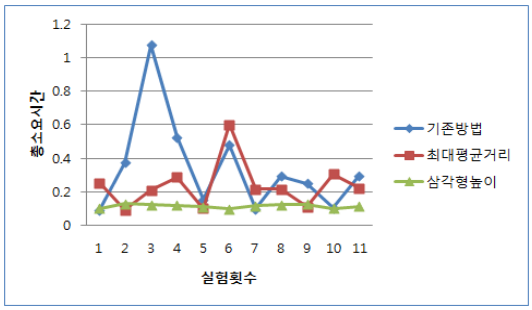


(그림 8) 300개 데이터, k=5인 경우 클러스터링 횟수 비교 실험

(Figure 8) Comparison Experiment of Clustering Count in K=5, using 300 data

기존 방법의 클러스터링 반복횟수가 클러스터 수를 전부 고려하여 평균 11.825이고, 최대평균거리를 이용하는 방법은 평균 9.75이고, 삼각형 높이를 이용하는 방법은 평균 7.7임을 확인할 수 있다. 실험에서 보는 것처럼 초기 중심 값이 바뀜으로서 클러스터링 횟수가 최대평균거리를 이용한 방법은 17.5% 감소하였고, 삼각형높이를 이용한 방법은 34.9% 감소한 것을 알 수 있다. 평균 총소요시간을 보면 최대평균거리를 이용한 방법은 기존 방법에 비해서 17.9% 감소하였고, 삼각형높이를 이용하는 방법은 38.4% 감소하였다.

(그림 8)은 300개 데이터에 대해서 K=5인 경우의 클러스터링 반복 횟수를 실험했고 그림에서 보는 바와 같이 최대평균거리를 이용하는 방법은 20.1% 감소한 것을 알



(그림 9) 300개 데이터, k=5인 경우 총소요시간 비교 실험

(Figure 9) Comparison Experiment of total Spending Time in K=5, using 300 data

수 있고, 삼각형높이를 이용하는 방법은 51.7% 감소한 것을 알 수 있다. (그림 8)은 총소요시간에 대해서 실험했다. 최대평균거리를 이용하는 방법은 기존방법에 비해서 25.3% 감소하였고, 삼각형높이를 이용한 방법은 61% 감소한 것을 알 수 있다.

기존 방법에 의한 초기 중심 선정의 경우 K의 값에 따라 할당-재계산의 횟수가 대체적으로 증가되는 것을 확인할 수 있다. 최대평균거리를 이용하는 방법도 기존방법보다는 총소요시간은 적으나 편차가 조금 있는 것으로 나타났다. 그러나 삼각형 높이를 이용하는 방법은 대체적으로 클러스터링 하는 횟수가 기존 방법보다 현저하게 줄어든 것을 볼 수 있다. 시간복잡도상의 수식으로 보면 최대평균거리나 삼각형높이를 이용하는 방법이 같을 것으로 생각되나 본 논문에서 제안한 방법이 좀 더 안정적으로 클러스터링 되는 것을 알 수 있다.

5. 결론 및 향후 과제

본 논문에서는 K-Means 알고리즘의 성능을 개선하기 위하여 초기 중심 선정 방법을 제안하였다. K-Means는 구현이 쉽고, 패턴 수가 N일 때 시간 복잡도가 선형적이기 때문에 일반적이다. 그러나 초기 클러스터 중심이 어떻게 설정되는가에 따라 클러스터링 결과가 이 초기 클러스터 중심에 종속적이다.

본 연구에서는 초기 중심을 선정하기 위하여 삼각형 높이를 이용하는 방법을 제안하고, 기존 연구인 임의로 초기 중심을 선정하는 방법, 최대평균거리를 이용하는 방법과 비교실험을 하였다. 실험에서 보는 것처럼 200개

데이터로 실험했을 때 초기 중심 값이 바뀜으로서 클러스터링 횟수가 최대평균거리를 이용한 방법은 17.5% 감소하였고, 삼각형 높이를 이용한 방법은 34.9% 감소한 것을 알 수 있다. 평균 총소요시간을 보면 최대평균거리를 이용한 방법은 기존 방법에 비해서 17.9% 감소하였고, 삼각형 높이를 이용하는 방법은 38.4% 감소하였다.

300개 데이터에 대해서 K=5인 경우의 클러스터링 반복 횟수를 실험했고 최대평균거리를 이용하는 방법은 20.1% 감소한 것을 알 수 있고, 삼각형높이를 이용하는 방법은 51.7% 감소한 것을 알 수 있다. 평균 총소요시간을 보면 최대평균거리를 이용하는 방법은 기존방법에 비해서 25.3% 감소하였고, 삼각형높이를 이용한 방법은 61% 감소한 것을 알 수 있다.

클러스터링은 정보검색이나 이메일 클러스터링, 통신 프로토콜의 클러스터링, 의료 정보에 대한 클러스터링 등 다양한 분야에 활용되고 있다. 본 논문에서 제안한 최대 평균 거리를 이용한 개선된 K-Means 알고리즘 또한 이들 분야에 적용할 수 있을 것이다.

향후, 분할 클러스터링에 국한되지 않고 계층적 클러스터링에 적용하여 정보 검색에 실제 응용할 수 있도록 연구가 지속되어야 한다.

참 고 문 헌(Reference)

[1] Giordano Adami, Paolo Avesani, and Diego Sona, "Clustering documents in a web directory", Proceedings of the 5th ACM international workshop on Web information and data management, pp.66-73, 2003.

[2] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, "Introduction to Information Retrieval", Cambridge University Press, pp.331-338, 2008.

[3] Jain, A. K. and Dubes, R. C., "Algorithms for Clustering Data". Prentice-Hall advanced reference series. Prentice-Hall, Inc., Upper Saddle River, NJ. 1988.

[4] S. P. Lloyd, "Least squares quantization in PCM", Special issue on quantization, IEEE Trans. Inform. Theory, 28, pp.129-137, 1982.

[5] McQueen, J. "Some methods for classification and analysis of multivariate observations", In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp.281-297, 1967.

[6] D.A.Meedeniya, and A.S.Perera, "Evaluation of Partition-Based Text Clustering Techniques to Categorize Indic Language Documents", IEEE International Advance Computing Conference(IACC 2009), pp.1497-1500, 2009.

[7] Paul Bunn, and Rafail Ostrovsky, "Secure Two-Party k-Means Clustering", Proceedings of the 14th ACM conference on Computer and communications security, Alexandria, Virginia, USA, pp.486-497, 2007.

[8] Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman and Chaitanya Swamy, "The Effectiveness of Lloyd-Type Methods for then k-Means Problem", Proceedings of the 47th Annual IEEE Symposium on Foundaions of Computer Science, pp.165-176, 2006.

[9] Nachiketa Sahoo, Jamie Callan, Ramayya Krishnan , George Duncan, and Rema Padman, "Incremental hierarchical clustering of text documents", Proceedings of the 15th ACM international conference on Information and knowledge management, pp.357-366, 2006.

[10] Yu Yonghong, and Bai Wenyang, "Text clustering based on term weights automatic partition", Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference, pp.373-377, 2010.

[11] Shinwon Lee, "A Study on Hierarchical Clustering using Advanced K-Means Algorithm for Information Retrieval", Chonbuk University doctoral thesis, 2005.

[12] Madhu Yedla et al., "Enhancing K-means Clustering Algorithm with Improved Initial Center", International Journal of Computer Science and Information Technologies, Vol. 1(2), pp.121-125, 2010.

[13] Shinwon Lee, Wonhee Lee, "Refining Initial Seeds using Max Average Distance for K-Means Clustering", Korean Society for Internet Information, pp.103-112, 2011.

◐ 저 자 소 개 ◑

이 신 원

1990년 전북대학교 전산통계학과(이학사)
1992년 전북대학교 대학원 전산통계학과(이학석사)
2005년 전북대학교 대학원 전자계산기공학과(공학박사)
2009년~현재 중원대학교 컴퓨터시스템공학과 교수
관심분야 : 정보검색, 한국어정보처리
E-mail : swlee@jwu.ac.kr