K-Means 클러스터링 성능 향상을 위한 최대평균거리 기반 초기값 설정

Refining Initial Seeds using Max Average Distance for K-Means Clustering

이 신 원* 이 원 휘**
Shin-Won Lee Won-Hee Lee

요 약

대규모 데이터에 대한 특성에 따라 몇 개의 클러스터로 군집화하는 클러스터링 기법은 계층적 클러스터링이나 분할 클러스터링 등 다양한 기법이 있는데 그 중에서 K-Means 알고리즘은 구현이 쉬우나 할당-재계산에 소요되는 시간이 증가하게 된다. 본 논문에서는 초기 클러스터 중심들 간의 거리가 최대가 되도록 하여 초기 클러스터 중심들이 고르게 분포되도록 함으로써 할당-재계산 횟수를 줄이고 전체 클러스터링 시간을 감소시키고자 한다.

ABSTRACT

Clustering methods is divided into hierarchical clustering, partitioning clustering, and more. If the amount of documents is huge, it takes too much time to cluster them in hierarchical clustering. In this paper we deal with K-Means algorithm that is one of partitioning clustering and is adequate to cluster so many documents rapidly and easily . We propose the new method of selecting initial seeds in K-Means algorithm. In this method, the initial seeds have been selected that are positioned as far away from each other as possible.

☞ keyword : 클러스터링(clustering), K-Means, 초기값(initial seed)

1. 서 론

대규모 데이터에 대한 특성 값에 따라 몇 개의 클러스터로 군집화하는 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 논문에서는 기존 방법인 무작위 추출을 통한 초기 클러스터 중심 선정 방법에서 벗어나 계산을 통한 초기 클러스터 중심을 선정함으로써 K-Means 알고리즘 성능을 개선하고자 한다.

본 논문에서는 초기 클러스터 중심들 간의 거리가 최대가 되도록 한다. 이를 통해 초기 클러스

* 정 회 원 : 중원대학교 IT공학부 교수
swlee@jwu.ac.kr
** 정 회 원 : 전북대학교 대학원 컴퓨터공학과(공학박사)
wony@jj.ac.kr(교신저자)
[2010/11/30 투고 - 2010/12/15 심사(2011/02/23 2차) - 2011/03/09 심사완료]

터 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 최대 평균 거리를 이용한 기법을 제안한다. 4장에서는 제안한 클러스터링 기법에 대한 실험을 하고 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)과 같다.

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

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

(그림 1)은 K-Means 알고리즘이다.

K-Means$\left(\{\vec{x_1}, \ldots \vec{x_N}\}, K\right)$
1. $\left(\vec{s_1}, \vec{s_2}, \ldots \vec{s_K}\right)$ <- Select Random Seeds $\left(\{\vec{x_1}, \ldots \vec{x_N}\}, K\right)$
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

종료 조건은 다음과 같은 경우를 사용할 수 있다.

(1) 미리 정의된 반복 횟수만큼 반복한다. 이 조건은 클러스터링 알고리즘의 실행 시간을 제한한다. 그러나 반복 횟수가 모자라서 클러스터링의 질이 떨어질 수 있다.

(2) 벡터가 속한 클러스터가 더 이상 변하지 않을 때까지 반복한다. 이 조건은 지역 최소(local minimum)에 들어가는 경우를 제외하고는 군집화의 질이 매우 좋다. 그러나 실행 시간이 길어진다는 단점이 있다.

(3) 중심이 더 이상 변하지 않을 때까지 반복한다.

(4) RSS가 임계치 이하로 떨어질 때까지 반복한다. 이 기준에 따라 종료되면 군집화의 질이 매우 좋다. 실질적으로 반복 횟수를 제한하는 방법과 이 조건을 결합해서 종료 조건으로 사용한다.

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.

[11]에서는 초기 클러스터 중심의 특성이 특정 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.

[8]은 클러스터간의 분리 크기에서 거리를 고려한다면, 각 최적 중심은 초기 센터를 가질 수 있을 것이라는 점에서 출발한다. 최적 중심에 매우 가까운 위치에 k개의 초기 중심을 찾기 위한 분리 조건을 개발하는 과정을 거치게 된다. 초기 중심을 얻는 과정은 다음 (그림 2)와 같다.

1. $\|x-y\|^2$의 비례확률에 의한 x, y를 선택하여 이를 c_1, c_2로 한다. 여기서 $x,y \in X$이며, X는 전체 데이터 집합이다.
2. 기존에 선정된 2개 이상($i \geq 2$)의 중심 $c_1,...,c_i$을 이용하여, 임의의 데이터 $x \in X$에 대한 비례확률 $\min_{j\in\{1,2,...i\}}\|x-c_j\|^2$을 구해 이를 c_{i+1}로 한다.
3. i가 k일 때까지 (2)단계를 반복한다.

(그림 2) [8]의 초기 중심 선정 알고리즘

[7]은 통신 보안 시스템에 적용하기 위하여 프로토콜을 대상으로 하는 K-Means 알고리즘을 연구하였는데 이를 Two-Party K-Means 클러스터링 프로토콜이라 한다. Two-Part를 구현하기 전에 우선 단일 데이터 집합을 위한 클러스터링 알고리즘의 초기 중심 선정에 대한 알고리즘이 필요한데 다음 (그림 3)과 같다. 기본 아이디어는 전체 문서의 중앙에서 출발하여 중심을 찾는 방법이다.

1. 전체 문서의 중앙 계산 : $C = \frac{1}{n}\left(\sum_{i=1}^{n} D_i\right)$
2. 모든 데이터와 중앙과의 거리 계산 : $\widetilde{C}_i^0 = Dist^2(C, D_i)$
3. 평균 거리 계산 : $\overline{C} := \frac{1}{n}\left(\sum_{i=1}^{n} \widetilde{C}_i^0\right)$
4. 첫 번째 중심 선택 : $\mu_1 = D_i, \quad \Pr[\mu_1 = D_i] = \frac{\overline{C} + \widetilde{C}_i^0}{2n\overline{C}}$
5. 나머지 중심을 선택하기 위해 반복 : $\mu_j, \quad j = 2,...,k$
   - 5.1 $\widetilde{C}_i^{j-1} = Dist^2(\mu_{j-1}, D_i), \quad 1 \leq i \leq n$
   - 5.2 $\widetilde{C}_i = \min\{\widetilde{C}_i^l\}_{l=0}^{j-1}, \quad 1 \leq i \leq n$
   - 5.3 $\overline{C} = average\,\widetilde{C}_i (over\ all\ 1 \leq i \leq n)$
   - 5.4 $\mu_j = D_i, \quad \Pr[\mu_j = D_i] = \frac{\widetilde{C}_i}{n\overline{C}}$

(그림 3) [7]의 초기 중심 선정 알고리즘

[7]은 위 알고리즘 통신을 주고받는 두 가상 사용자 밥(Bob)과 엘리스(Alice)가 주고받은 데이터

를 각각 D^B, D^A라 하고, 각각의 중심을 μ_j^B, μ_j^A라 하여 위 알고리즘(그림 3)의 모든 수식에 적용하여 초기 중심을 선정한다. 이 알고리즘은 통신을 통해 주고 전송된 메시지의 보안을 위해 제안된 방법으로 동일할 것이라는 두 데이터 집합을 대상으로 한다.

3. 최대평균거리를 이용한 클러스터 중심 선정

3.1 알고리즘

본 논문에서는 클러스터링 초기 중심에 새로운 방법을 이용하여 K-Means 알고리즘을 개선하고자 한다. 제안하고자 하는 초기 클러스터 중심 선정 방법은 초기 클러스터 중심들을 가능한 한 멀리 선정하도록 한다. 이렇게 함으로써 무작위로 선정된 초기 클러스터 중심이 일부 영역으로 편향되는 현상을 막을 수 있고, 이에 따라 클러스터링 속도 향상과 클러스터링의 정확도를 높이고자 하였다. 제안한 K-Means 알고리즘의 초기 클러스터 중심 집합C는 다음 식(4)와 같다.

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

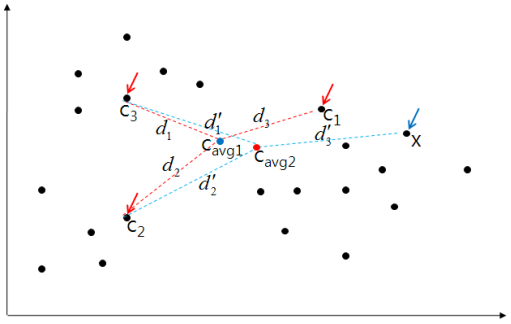
여기서 c_i는 i번째 클러스터의 중심이며, c_{avg}는 c_1부터 c_k까지의 평균이다. 즉, c_1부터 c_k까지의 중심들이 이들의 평균으로부터 최대의 거리를 갖도록 하는 것이다. (그림 4)는 초기 클러스터 중심을 구하는 알고리즘이다.

다음 (그림 5)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($\{d_1', d_2', d_3'\}$)

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 평균 거리 계산
$$newDistAvg \leftarrow avg \sum_{i=1}^{k} |c_{avg} - c_i|^2$$
$$\text{if } c_i = candidateCluster \text{ then } |c_{avg} - x|$$
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 4) 초기 클러스터 중심 선정 알고리즘



(그림 5) 최대 평균 거리를 이용한 초기 중심 이동

를 계산한다.

$$newDistAvg = \frac{1}{K} \sum_{k=1}^{K} d_j' \tag{5}$$

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

$$oldDistAvg = \frac{1}{K} \sum_{k=1}^{K} d_j \tag{6}$$

와 비교하게 된다. 두 평균 거리를 비교한 결과 c_1 대신에 x를 대입해서 계산한 $newDistAvg$값이 더 크기 때문에 x가 새로운 c_1으로 대체된다. 이제 x, c_2, c_3가 새로운 $oldDistAvg$가 되어 다음 x의 비

교 대상이 된다. 이 과정을 데이터 집합X에 속한 모든 x에 대해 반복한다.

3.2 시간 복잡도 평가

기존연구의 중심 선정 방법에 비해 본 논문에서 제안한 방법은 최대 평균 거리를 계산하는 과정이 필요하다. 즉, 클러스터링에 소요되는 시간은

$$T(\text{초기중심설정}) + T(\text{할당} - \text{재계산}) \quad (7)$$

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

(그림 4)의 알고리즘에서 보는 바와 같이 2.1단계의 x와 가장 가까운 중심 선택에 소요된 시간 $1K$, 2.2단계에서 기존 중심들과 x로 대체했을 때의 중심들에 대해, 중심들의 평균값 구하기 위한 시간 $1K$, 평균값과 각 중심 간의 거리 계산을 위한 시간 $1K$가 소요되어 총 $5K$ 만큼의 시간이 소요된다. K는 클러스터 수이다. 기존 K-Means 알고리즘의 할당-재계산 과정의 시간 복잡도가 $O(KN)$이라고 할 때, 최대 평균 거리 계산을 위한 시간 복잡도는

$$\fallingdotseq O(5KN) \quad (8)$$

이다.

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

$$O(2iKN) \quad (9)$$

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

따라서, 전체 클러스터링에 소요되는 시간은

$$O(5KN) + O(2iKN) \approx O(N) \quad (10)$$

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

$$O(5KN) \ll O(2iKN) \quad (11)$$

이므로 초기 중심 선정에 소요되는 시간은 전체 클러스터링 소요 시간에 큰 영향을 미치지 않는다. 이는 실험을 통해 확인하도록 한다.

4. 실험 및 성능 평가

4.1 데이터집합

클러스터링 결과에 대한 수치적 평가를 위해 148개의 데이터를 6개의 군집으로 생성하고 이를 대상으로 클러스터링 성능을 실험하였다. 데이터의 개수는 툴 출력되는 데이터 포인트들이 육안으로 식별하기 쉽도록 적은 수로 제한하였다. 생성된 데이터 집합은 다음 (표 1)과 같다.

(표 1) 데이터 집합

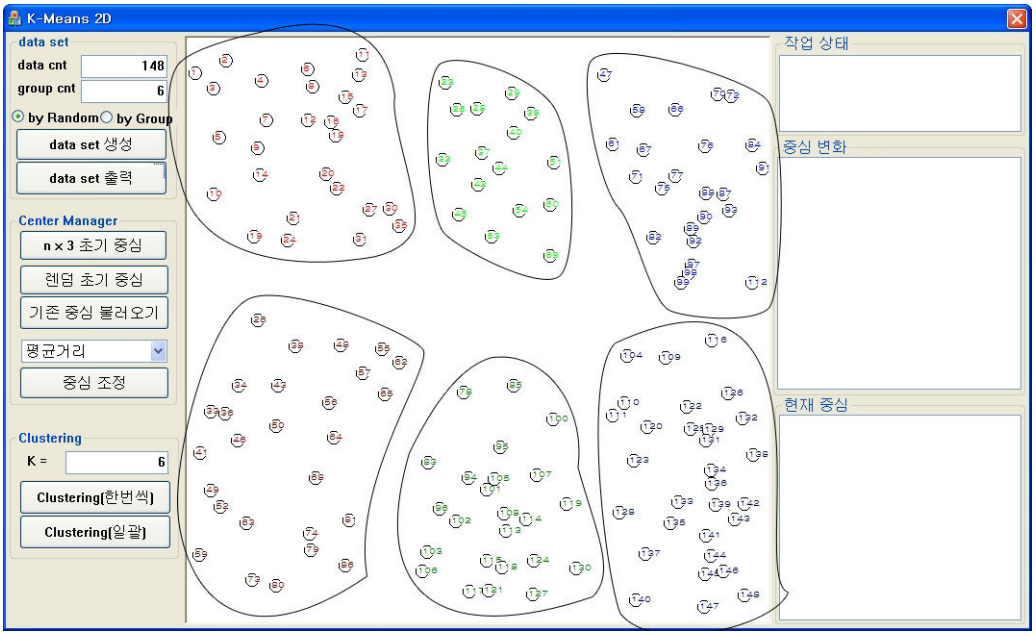
c_i	0	1	2	3	4	5	계
데이터 수	27	16	24	27	24	30	148

(그림 6)은 (표 1) 데이터 집합을 표시한 그림이다.

클러스터링 실험은 각 초기 클러스터 중심 선정 방법에 대하여 10회씩 실시하여 결과를 확인하였다.

4.2 정확도 측정

클러스터링 결과에 대한 평가는 F-Measure로 평가한다. F-Measure 수식은 우연성 행렬(contingency matrix)이 다음의 (표 2)와 같을 때



(그림 6) 클러스터 데이터 분포

(표 2) 우연성 행렬

Partition C

Partition P	C_1	C_2	$\cdots$	$C_{K'}$	$\sum$
P_1	n_{11}	n_{12}	$\cdots$	$n_{1k'}$	$n_{1.}$
P_2	n_{21}	n_{22}	$\cdots$	$n_{2k'}$	$n_{2.}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
p_K	n_{K1}	n_{K2}	$\cdots$	$n_{KK'}$	$n_{K.}$
$\sum$	$n_{.1}$	$n_{.2}$	$\cdots$	$n_{.K'}$	n

j번째 클러스터의 F-Measure :

$$F-measure(C_j) = \max_i \left[2 \cdot \left(\frac{p_{ij}}{p_i} \cdot \frac{p_{ij}}{p_j} \right) / \left(\frac{p_{ij}}{p_i} + \frac{p_{ij}}{p_j} \right) \right] \quad (12)$$

여기서,

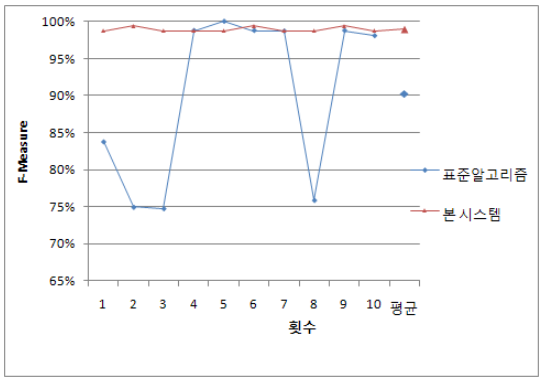
$p_{ij} = n_{ij}/n, \quad p_i = n_{i.}/n, \quad p_{j=n_{.j}}/n$이다.

(표 3)은 실험을 통해 얻어진 F-Measure이다.

(표 3)과 (그림 6)에서 보는 바와 같이 표준 알고리즘의 경우 정확하게 클러스터링 되는 경우도 있으나 상대적으로 그렇지 않은 경우도 발생하여 클러스터링 결과가 초기 클러스터 중심에 종속적임을 확인할 수 있다. 그러나 본 논문에서 제안한 초기 클러스터 중심을 이용하는 경우 대체적으로 일관된 성능을 보이는 것을 확인할 수 있다.

(표 3) 실험 결과(F-Measure)

횟수	표준알고리즘	본 시스템
1	83.80%	98.77%
2	75.00%	99.38%
3	74.81%	98.77%
4	98.77%	98.77%
5	100.00%	98.77%
6	98.77%	99.38%
7	98.69%	98.77%
8	75.93%	98.77%
9	98.77%	99.38%
10	98.07%	98.77%
평균	90.26%	98.95%



(그림 7) 실험 결과(F-Measure) 그래프

(그림 7)은 결과를 그래프로 표현한 것이다.

이상에서 살펴본 바와 같이, K-Means 알고리즘이 초기 클러스터 중심 선정에 따라 클러스터링 성능이 높아지고 낮아지는 현상을 본 논문에서 제안한 방법을 통해 해소할 수 있었으며, 클러스터링 계산 시간을 단축시킴으로써 초기 클러스터 중심을 조정하기 위해 소요되는 시간을 상쇄할 뿐만 아니라 전체적인 실행시간 또한 줄일 수 있었다.

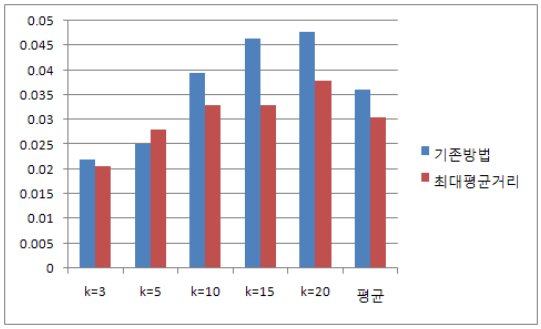
4.3 소요 시간

분산도 측정에 사용한 데이터를 이용하여 클러스터링에 소요되는 시간을 측정하였다. 실험은 분산도 측정과 마찬가지로 K값을 3~20으로 조정하며 각각을 10회씩 실행하여 평균을 구하였다. (표 4)는 실험 결과를 표로 표현한 것이며, 이 중 총소요시간에 대한 그래프가 (그림 8)이다.

(표 4)를 보면 기존 방법에 의한 초기 중심 선정의 경우 K의 값이 커질수록 할당-재계산의 횟수가 대체적으로 증가되는 것을 확인할 수 있다. 이는 데이터 집합이 무작위로 생성되어 경계가 모호한 부분이 많기 때문에 발생되는 현상이다. 그러나 최대 평균 거리의 경우 증가되는 편차가 크지 않게 나타나는 것을 확인할 수 있다. 이는 중심들이 데이터 집합의 가장자리에서 안쪽으로 이동하면서 클러스터를 형성하기 때문에 기존 방법보다 데이터 집합의 분포에 영향을 덜 받는 것으로 판단된다. 이로써 초기 중심 선정에 추가적인 소요 시간이 필요하다 하여도 할당-재계산 횟수를 줄임으로써 클러스터링 총 소요 시간을 감소시킬 수 있었다.

(표 4) 소요 시간

	소요시간		k=3	k=10	k=20	평균
기존방법	할당-재계산	평균	0.0017	0.0022	0.0023	0.0021
		횟수	12.8	17.4	20.4	16.94
		합계	0.02176	0.03828	0.04692	0.035574
	총소요시간		0.02176	0.03828	0.04692	0.035574
최대평균거리	할당-재계산	평균	0.0020	0.0021	0.0025	0.0022
		횟수	7.3	9.2	9.3	8.24
		합계	0.0146	0.01932	0.02325	0.018128
	중심조정		0.0057	0.0134	0.0142	0.0118
	총소요시간		0.0203	0.03272	0.03745	0.029928



(그림 8) 총소요시간

5. 결론 및 향후 과제

본 논문에서는 대량의 데이터에 대한 클러스터링에 주로 사용되는 분할 클러스터링 중 K-Means 알고리즘의 성능을 개선하기 위하여 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.

초기 클러스터 중심을 무작위로 선정하는 방식에서 벗어나 초기 중심들을 최대한 멀리 배치함으로써 클러스터링의 성능을 향상시키고자 하였다. 정확도 측면에서 보면, 표준 알고리즘의 경우 정확하게 클러스터링 되는 경우도 있으나 상대적으로 그렇지 않은 경우도 발생하여 클러스터링 결과가 초기 클러스터 중심에 종속적임을 확인할 수 있다. 그러나 본 논문에서 제안한 초기 클러스터 중심을 이용하는 경우 대체적으로 일관된 성능을 보이는 것을 확인할 수 있었으며, 표준 알고리즘에 비해 약 8.69%의 높은 정확도를 나타내었다. 또한, 본 논문에서 제안하는 초기 중심 선정 방법을 이용하면, 기존 방법과 달리 초기 클러스터 중심 선정을 위한 추가적인 시간이 소요된다. 그러나 문서를 각 클러스터에 할당하고 중심을 다시 계산하는 할당-중심 재계산 과정의 횟수를 감소시킴으로써 시간 복잡도는

$$O(5kN) + O(2ikN) \approx O(N) \quad (13)$$

으로 문서 수에 선형적이며 전체 클러스터링에 소요되는 시간을 감소시킬 수 있었다. 또한, 클러스터링 결과가 초기 클러스터 중심에 종속적이던 현상을 해소하여 일관된 결과를 얻을 수 있었다.

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

[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] 이신원 "정보검색을 위한 개선된 K-Means 알고리즘을 이용한 계층적 클러스터링에 관한 연구", 박사학위 논문, 전북대학교, 2005.

◐ 저 자 소 개 ◑

이 신 원

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

이 원 휘

1997년 전주대학교 경영학과(경영학사)
1999년 전주대학교 대학원 컴퓨터공학과(공학석사)
2010년 전북대학교 대학원 컴퓨터공학과(공학박사)
관심분야 : 정보검색, 자연어처리, 클러스터링
E-mail : wony@jj.ac.kr