예측을 이용한 효율적인 K-Means 알고리즘

지태창*, 이현진**, 이일병*
*연세대학교 컴퓨터과학과
**한국싸이버대학교 컴퓨터정보통신학부
e-mail : garura@csai.yonsei.ac.kr

An Efficient K-means Clustering Algorithm using Prediction

Tae-Chang Jee*, Hyunjin Lee**, Yillbyung Lee*
*Dept. of Computer Science, Yonsei University
** Dept. of Computer, Information & Communication, Korea Cyber University

요 약

본 논문에서 k-means 군집화 알고리즘을 효율적으로 적용하는 방법을 제안했다. 제안하는 알고리즘의 특징은 속도 향상을 위해 예측 데이터를 이용한 것이다. 군집화 알고리즘의 각 단계에서 군집을 변경할 데이터만 최인접 군집을 계산함으로써 계산 시간을 줄일 수 있었다. 제안하는 알고리즘의 성능 비교를 위해서 KMHybrid 와 비교했다. 제안하는 알고리즘은 데이터의 차원이 큰 경우에 KMHybrid 보다 높은 속도 향상을 보였다.

1. 서론

군집화는 주어진 입력 데이터를 유사한 특징을 가지는 부분집합으로 구분하는 계산 작업이다. 일반적으로 이 부분집합들을 군집이라고 부른다. 군집들은 군집을 이루는 데이터들은 서로 유사한 성질을 가지고, 다른 군집들과는 다른 성질을 가지게 된다. 군집화를 수행함으로써 원 데이터 집합의 정확도는 잃어버리지만, 단일성을 얻게 된다[1].

군집화는 데이터의 구조 등과 같은 사전 정보 없이 분석을 시작한다는 점에서 '자료의 탐험' 또는 '자료의 발굴' 과정으로 볼 수 있다. 그리고 양적으로 많은 데이터를 훨씬 작은 개수의 동질적 집단으로 분류함으로써 최소의 정보손실을 통한 '자료의 집약' 내지 '자료의 단순화'의 과정으로 볼 수 있다. 군집화의 결과는 모집단의 구조적 특성에 관한 정보를 도출함으로써 '가설의 형성' 단계와도 연계된다[2].

대상들을 군집화하는 방법은 매우 다양하지만 모든 방법이 공통적으로 가지고 있는 기본전제는 군집 내의 객체들 간의 유사성을 극대화 하고, 군집간의 유사성은 극소화하는 것이다. 군집화 방법으로는 계층적 기법 (hierarchical algorithms), 분할 기법 (partitioning algorithms) 등이 있다.

가장 많이 사용되는 분할 기법은 K-Means 라고 많이 불리는 Lloyd's algorithm 이다. K-Means 의 주요 장점은 단순하고, 데이터 분석의 기초가 되며 일반적으로 효율적이라는데 있다. 단점은 사용자가 미리 군집의 개수를 설정해야 하고, 예외 값을 다루기 어렵다.

본 논문에서는 예측을 이용한 효율적인 K-Menas 알고리즘을 제안하다. 예측을 사용해서 군집화 계산 시간을 단축시키는 것이다. 제안하는 방법은 원래의 K-Means 와 KMHybrid[3][4]와 비교하여 좋은 성능을 보였다.

본 논문의 구성은 다음과 같다. 먼저, 2 장에서는 기존 방법에 대해서 기술하고 3 장에서는 제안하는 방법을 설명한다. 그리고 4 장에서는 실험결과를 분석하고, 마지막 5 장에서 결론을 보인다.

2. Related Work

K-Means 에서 가장 일반적인 알고리즘은 Lloyd's 알고리즘이다[5]. 최근에 이 알고리즘을 효율적으로 적용하는 방법들이 개발되고 있다[3]. 이러 방법들은 알고리즘에서 가장 시간이 오래 걸리는 단계인 가장 가까운 이웃을 계산하는 시간을 줄이는 것을 목적으로 한다.

KMHybrid 는 K-Means 의 중심을 swap 하는 방법과 다양한 simulated annealing 기법을 결합한 방법이다[3][4]. 이 알고리즘은 각 단계에서 군집의 중심을 swap 하고 용인 실험을 한다. 변경된 솔루션이 실험을 통과하면, 변경된 솔루션으로 계속 진행하고, 그렇지 않으면, 이전 솔루션을 복원한다. 용인 실험은 simulated annealing 을 이용하여 수행한다. 추가로 kD-tree 를 사용하여 최인접 이웃을 찾는 시간을 단축했다. 자세한 알고리즘은 [3][4]에서 찾을 수 있다.

3. 제안하는 방법

K-Means 를 수행할 때 최인접 이웃을 찾는 단계를 보면, 이전 군집 소속이 변경되는 경우는 초기 단계에서는 많을 지 몰라도 단계가 진행 될수록 점점 적어지게 된다. 따라서, 소속이 변경될 데이터 만 인접 군집을 다시 계산하면 전체 계산 시간을 단축 시킬 수 있다.

제안하는 방법을 수행하는 과정은 일반적인 K-Means 의 단계와 동일하다. 단, 최인접 군집을 계산하는 과정에서 전체 입력 데이터에 대해서 계산을 수행하는 것이 아니라, 변경될 가능성이 있는 데이터에 대해서만 계산을 수행함으로써 계산 수행 시간을 단축하는 것이다.

입력 데이터를 $\mathbf{X}$ 라고 하면, 각 단계에서 소속이 변하는 데이터는 $\mathbf{X}'_l$, $l \in \{1...L\}$ 이 된다. 여기서, L 은 알고리즘의 반복 횟수이다. $\mathbf{X}'_l$ 을 정확하게 찾기는 어렵기 때문에 이의 예측치인 $\hat{\mathbf{X}}_l$ 을 찾아서, 이에 대해서만, 최인접 군집을 계산하게 된다.

$\hat{\mathbf{X}}_l$ 은 예측 거리 $\hat{\mathbf{D}}$ 를 계산해서 구한다. $\hat{\mathbf{D}}$ 은 다음과 같이 계산하다.

$$\hat{D}_{ji} = \begin{cases} D_{ji} - \hat{C}_j & if \;\; X_i \notin C_j \\ D_{ji} + \hat{C}_j & if \;\; X_i \in C_j \end{cases}$$

$\mathbf{D}$ 는 현재의 데이터와 군집간의 거리 행렬이고, $\hat{\mathbf{C}}$ 은 현재 군집의 중심과 변화된 군집의 중심과의 거리 차이다. $\hat{\mathbf{D}}$ 은 현재 군집에 속한 데이터의 거리는 멀게 하고, 군집에 속하지 않은 데이터의 거리는 더 가깝게 함으로써 항상 $\mathbf{X}'_l \subset \hat{\mathbf{X}}_l$ 이 되게 한다. $\mathbf{X}'_l \not\subset \hat{\mathbf{X}}_l$ 이 되면, 각 단계의 결과가 원 군집 결과와 달라지기 때문에 최종 군집 결과는 달라지게 된다.

$\hat{\mathbf{D}}$ 을 스캔해서 이전 단계에서 소속했던 군집이 변경된 데이터를 $\hat{\mathbf{X}}_l$ 에 할당한다. $\hat{\mathbf{D}}$ 을 계산할 때는 거리만을 이용하기 때문에 데이터의 차원은 고려할 필요가 없다. 따라서, $\hat{\mathbf{D}}$ 을 계산하는 시간은 고차원 데이터의 경우에 상대적으로 작게 된다.

$\hat{\mathbf{X}}_l = \mathbf{X}_l$ 이면, $\hat{\mathbf{D}}$ 을 계산하는 시간이 추가되기 때문에 계산 시간은 늦어지게 된다. 일반적으로는 $\hat{\mathbf{X}}_l < \mathbf{X}_l$ 이기 때문에 원 알고리즘 보다는 계산 시간은 빨라지게 된다.

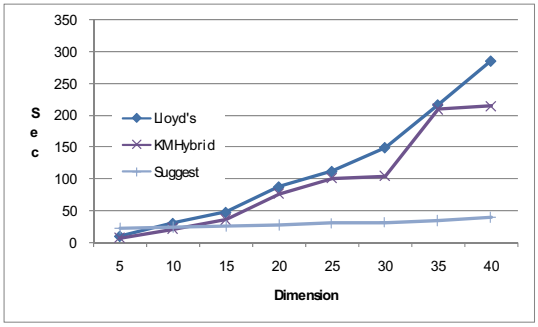
4. 실험 결과

<표 1> 데이터 집합

Instance	Size	Clusters	Dimension
Art05D	300,000	20	5
Art10D	300,000	20	10
Art15D	300,000	20	15
Art20D	300,000	20	20
Art25D	300,000	20	25
Art30D	300,000	20	30
Art35D	300,000	20	35
Art40D	300,000	20	40

실험은 Intel Core2Duo E6550, 2GB RAM 시스템 상에서 .Net Framework 2.0 을 이용해서 수행했다. 실험 데이터는 임의로 (표 1)과 같이 데이터 크기와 군집의 개수는 고정하고, 데이터 차원을 5 부터 40 까지 순차적으로 증가시켜서 데이터 집합을 구성했다.

KMHybrid[4]와 Original Lloyd's[4] 와 비교한 수행 결과는 <그림 1>과 같다.



<그림 1> 수행 시간 비교

<그림 1>을 보면, KMHybrid 는 Lloyd's 에 비해서 전체적으로 수행 시간이 좋은 것을 알 수 있다. 반면에 제안하는 방법은 Dimension 이 5 일 때는 Lloyd's 보다도 나쁘고, 10 일 때는 KMHybrid 보다 나쁘지만, 15 이상부터는 Lloyd's 와 KMHybrid 의 성능을 월등히 능가하는 것을 알 수 있다. 따라서, 제안하는 방법은 데이터의 차원(Dimension)이 큰 경우에 효율적이라고 할 수 있다.

5. 결론

본 논문에서는 예측을 이용해서 K-Means 군집화 알고리즘을 효율적으로 개선한 알고리즘을 제안했다. 제안하는 알고리즘은 KMHybrid[4]와 비교하여 데이터의 차원이 큰 경우에 좋은 성능을 보였다. 향후, 데이터의 크기, 군집의 개수, 알고리즘의 반복 횟수에 대한 비교 및 저차원 데이터에서의 성능 개선 방안에 대해 연구할 계획이다.

참고문헌

[1] G. Frahling and C. Sohler, "A fast k-means implementation using corsets," Proceedings of the twenty-second annual symposium on Computational Geometry (SoCG'06), pp. 135-143, 2006.

[2] A.Jain and R.Dubes, "Algorithms for Clustering Data," Prentice Hall, New Jersey, 1998.

[3] T. Kanungo, D.Mount, N. Netanyahu, C. Piatko, R. Silverman and A. Wu, "A Local Search Approximation Algorithm for k-Means Clustering," Proceedings of the 18th Annual Symposium on Computational Geometry (SoCG'02), pp. 10-18, 2002.

[4] D.Mount, "KMlocal: A Testbed for k-menas Clustering Algorithms" Availabel at http://www.cs.umd.edu/~mount

[5] S.Lloyd, "Least Squares Quantization in PCM," IEEE Transactions on Information Theory, Vol.(28), pp. 129-137, 1982.