2000년 한국정보처리학회 추계 학술발표논문집 제7권 제2호

데이터형태에 적응하는 클러스터링 알고리즘

이기호*, 이기철
홍익대학교 전자계산학과
{kholee, lee}@cs.hongik.ac.kr

Data Clustering Algorithm Adaptive to Data Forms

K. H. Lee*, K. C. Lee
Dept. of Computer Science,
Hong-Ik University

요약

클러스터링에 있어서 k-means[7], DBSCAN[2], CURE[4], ROCK[5], PAM[8], 같은 기존의 알고리즘은 원형이나 타원형 등의 어느 고정된 모양에 의해 클러스터를 결정한다. 만약 클러스터 하려는 데이터의 분포가 우연히 알고리즘의 결정된 모양과 일치하면 정확한 해를 얻을 수 있다. 하지만 자연적인 데이터의 분포에서는 발생하기 어렵다. 데이터의 형태를 추적하여 이러한 문제점을 해결한 CHAMELEON[1] 알고리즘이 최근에 발표되었다. 하지만 모양에는 독립적이나 데이터의 양이 증가함에 따라 소요되는 시간이 폭발적으로 증가한다 이것은 기존의 마이닝 데이터들이 대용량이라는 것을 고려하면 현실에 적용하기 힘든 문제점이 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 K-means[7]를 이용한 대표를 선출하는 방법으로 CHAMELEON[1]의 문제점 개선(EF-CHAMELEON)을 시도하였으며 여러 자연적인 형태의 도형들은 아주 작은 원형들의 집합으로 구성 될 수 있다는 생각을 기본으로 잡음에 영향을 받지 않을 정도로 아주 작은 초기 다수의 소형 클러스터를 K-mean을 이용하여 구성하고 이를 다시 크러스터간의 상대적인 거리를 이용하여 다시 머지 하는 방법으로 모양에 의존적인 문제를 해결하며 비교사 학습(unsupervised learning)에 충실하기 위해 임계값을 적용 적정 단계에서 알고리즘을 멈추게 한 ADF 알고리즘을 소개한다. 실험 데이터는 기존의 여러 클러스터링 알고리즘이 판별 할 수 없었던 다양한 모양을 가지고있는 2차원 배열을 사용하여 ADF, CHAMELEON[1], EF-CHAMELEON,의 성능을 비교하였다.

1. 서론

데이터마이닝에서의 클러스터링이란 대용량의 데이터에서 서로 유사성(similarity)이 많은 것들이 군집화하는[1] 특성을 이용 각각의 그룹의 관계를 정립하는 것으로 다른 마이닝 알고리즘이 특별한 정보나 배경지식을 사용하는 반면 클러스터링은 그러한 정보나 배경지식을 사용하지 않고 데이터로부터 직접 결과를 이끌어 내는 비교사 학습 (unsupervised learning)의 특징을 가지고 있으며 이것은 다른 알고리즘에 비해 클러스터링이 가지고있는 매우 큰장점이며 이러한 장점 때문에 크러스터링의 활용과 중요도는 더욱 증가해 왔으며 수년동안 많은 연구가 이루어져 여러 가지 알고리즘이 발표되었다. 하지만 기존의 클러스터링 알고리즘인(PAM[8], CLARA[8], ROCK[5], DBSCAN[2], K-means[7], CURE[4])들은 어느 고정된 모양에 의해 클러스터를 결정한다. 이것은 데이터의 분포가 고정되어 있지 않다는 것을 고려한다면 매우 큰 문제점이 아닐 수 없다. 이러한 문제점을 개선한 CHAMELEON[1]알고리즘이 발표되었지만 데이터의 양이 증가함에 따라서 소요되는 시간이 폭발적으로 증가한다. 이것은 마이닝 데이터가 대용량이라는 점을 고려하면 매우 큰 문제점이 아닐 수 없다. 이러한 문제점을 해결하기 위해 본 논문에서는 K-means[7]를 이용 대표를 선출하는 방

본 연구는 과학재단 목적기초 97-01-02-0401-1(제 1 세부과제)로 수행되었음

법으로 CHAMELEON[1]의 문제점의 개선을 시도한 EF-CHAMELEON을 소개하며 여러 자연적인 형태의 도형들은 아주 작은 원형들의 집합으로 구성 될 수 있다는 생각을 기본으로 잡음에 영향을 받지 않을 정도로 아주 작은 초기의 다수의 소형 클러스터를 K-mean[7]을 이용하여 구성하고 이를 다시 클러스터 간의 상대적인 거리를 이용하여 다시 머지 하는 방법으로 모양에 의존적인 문제를 해결하며 비교사 학습에 충실하기 위해 임계값을 적용 적정 단계에서 알고리즘을 멈추게 한 ADF알고리즘을 소개한다. 실험 데이터는 기존의 여러 클러스터링 알고리즘이 판별 할 수 없었던 다양한 형태를 가지고있는 2차원 배열을 사용하여 CHAMELEON[1], EF-CHAMELEON ADF의 성능을 비교하였다. 본 논문 구성은 2장에서 기존의 알고리즘의 소개와 한계점을 알아보고. 3장에서는 기존CHAMELEON[1]알고리즘의 개선 알고리즘(EF-CHAMELEON)을 소개하며. 4장에서는 본 논문에서 제안한 ADF알고리즘을 소개한다. 5장에서는 ADF, CHAMELEON[1], EF-CHAMELEON,의 성능을 비교하였고 6장에서 결론을 내린다.

2. 관련연구

기존의 중앙값 기반(Centroid-based)의 클러스터링(K-means[7], ISODATA[9])은 중앙값과의 평균거리의 제곱이 최소가 되도록 클러스터를 형성한다. 또한 Medoid(Medoid-based)값에 의한 클러스터링(PAM[8], CLARANS[10])은 각 점에서 가장 가까운 Medoid로부터의 거리의 합이 최소가 되는 Medoid를 선택하는 방식이다. 하지만 위의 두 알고리즘은 공통적인 문제점을 가지고 있다 그것은 자연적인 데이터에서는 (그림1)과 같이 자기가 속해야 하는 클러스터의 중앙보다 다른 클러스터의 중앙에 보다 가까운 경우가 더 많기 때문이다.

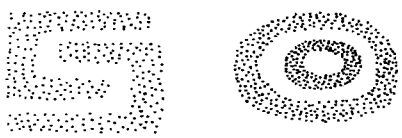


그림 1

이러한 문제점을 해결한 알고리즘인 CHAMELEON[1]에서는(그림2) 데이터들에서 먼저 K-nearest Neighbor Graph를 구하고 그것을 다시 그래프 분할(hMeTis[7])알고리즘을 이용하여 충분히 작은 크기의 초기 클러스터로 분할하고 이들의 상대적 거리(Relative Closeness)와 상대적 연결정도(Relative Inter-Connectivity)를 고려하여 합병하는 방법을 사용하였다. 이것은 위에서 언급한 문제점은 해결하였지만 데이터양이 증가함에 따라서 K-nearest Neighbor Graph 그래프를 구하고 그래프 분할(hMetis[6])알고리즘의 반복적 사용으로 소요되는 시간이 매우 급격히 증가하는 문제점이 있다.

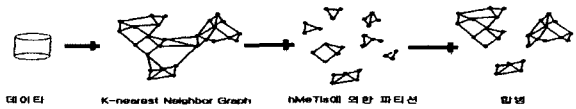


그림2

3. CHAMELEON의 개선(EF-CHAMELEON)

CHAMELEON[1]은 모든 데이터에 대한 k-인접 그래프를 형성 그것을 분할하고 다시 합병한다. 하지만 모든 데이터의 사용은 매우 비효율적이라 하지 않을 수 없다. 여기서는 데이터를 다수의 아주 작은 초기 클러스터로 나누고 클러스터를 대표하는 값을 이용하여 계산의 양을 줄이는 알고리즘을 소개한다. 이 알고리즘에서 초기 클러스터를 구하기 위해 큰 값의 K를 이용한 K-means[7] 알고리즘을 사용하였다. K-means[7] 알고리즘은 다음과 같이 처음에 먼저 고정된 클러스터들의 개수 K가 주어지고 각 클러스터의 중심으로 추정된 중심으로 K개의 임의의 포로토타입(W1,W2...,Wk)이 생성되며 각 입력 패턴(i1,i2,.....,in)은 이중 거리가 가장 가까운 포로토타입에 할당된다. 그 다음 각 클러스터에서 속해 있는 입력 패턴들의 평균을 구하여 그 다음 각 클러스터의 새로운 프로토타입으로 설정하고 위 단계를 반복하게 된다. 보통 클러스터링의 수준은 오류함수(error function)(식1)로 측정된다.

$$E = \sum_{j=1}^{k} \sum_{i \in c_j} | i_i - W_j |^2 \qquad (식1)$$

여기서 실험결과 K값을 충분히 큰 값으로 주면 프로토타입들은 데이터에 고르게 분포함을 알 수 있었

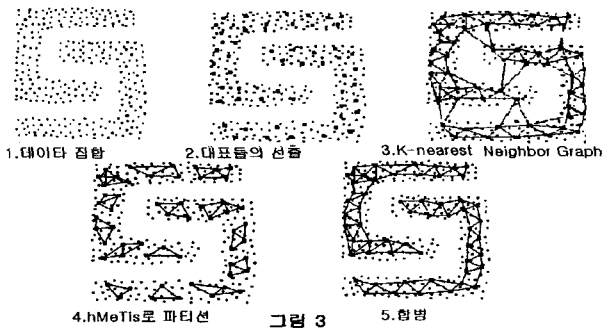


그림 3

다. 이러한 성질을 이용하여 프로토타입들을 초기 클러스터로 간주 하나의 대표들만을 선정하여 CHAMELEON알고리즘에 적용 후 대표들이 속한 클러스터에 각 데이터들을 포함시킨다(그림3).

4. ADF 알고리즘

ADF 알고리즘은 입력 패턴(i1,i2,....in)을 충분히 작은 초기 클러스터(S1,S2,.....Sk)로 분할하고 각각의 Sj(3)에서 대표(Pj)는 Sj가 포함하고 있는 입력 패턴의 평균으로 선출한다. 클러서터 Ci은 Sj의 집합(4)이고 크러스터(Ci,Cj)의 거리dist(Ci,Cj)={dist(Mi,Mj)|Mi: min{dist(Pi,Pj)}의 Pi를 가지는 Si에서 Pj에 가장 가까운 n개의 평균, Mj : min{dist(Pi,Pj)}의 Pj를 가지는 Sj에서 Pi에 가장 가까운 n개의 평균}/((RC(Ci)+RC(Cj))/2)로 정의. RC(Ci)는 (Ci에서 랜덤한 n개의 평균거리-Ci중앙에서 각 점의 평균거리) 최소를 합병한다(그림3). 실험에서 n은 (초기 클러스터의수 /2)를 사용

$$Si = \sum_{j=1}^{m} il \qquad (3)$$

m은 Subset이 포함한 각 입력 패턴 의 수

$$Ci = \sum_{i=1}^{c} Si \qquad (4)$$

c는 클러스터가 포함한 Subset의 수

---------- 알고리즘 ----------

```
Procedure cluster()
Begin
1   clusters  =  Direct k-means(충분히 작은k);
2   While(  minDist < threshold ){
3       for(i=0; i<k; i++){
4           for(j=i+1; j<k ;j++){
5                   MCdis = Closest_cluster(Ci, Cj);
6                   if( minDist >= MCdis){
7                       BestLC=Ci;
8                      BestRC=Cj;
9                       minDist=MCdis;
10                  }
11          }
12      }
13      If (minDist >threshold)
14          break;
15      Merge(BestLC,BestRC,); /* 클러스터들을 합병*/
    }
End
1.Subset(S1,....,Sk)와 대표(P1,....,Pk)를 생성.
5.Closest_Cluster(Ci,Cj){
      /*각각의 클러스터에서 가장 가까운 Pi와 Pj를 찾는다   */
      BestMean(Ci,Cj);
```

$$Meandis = \frac{Dis(Mi, Mj)}{\frac{Rc(Ci) + Rc(Cj)}{2}}$$

```
/**********************************************************
   Rc(ci)      : ci에서 랜덤한 n개의 평균거리 - ci중앙에서 각
                 점의 평균 거리
   Dis (Mi,Mj) : Mi,Mj의 거리
                 Mi: Ci에서 가장가까운 Pj에 속한 Sj에서 Pi에
                 가장 가까운 n 개의 평균
**********************************************************/
   return Meandis;
}
```

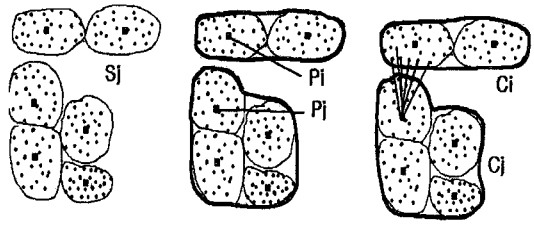


그림 4

5. 실험 및 결과

실험 데이터는 기존의 여러 클러스터링 알고리즘이 판별 할 수 없었던 자연적인 모양(그림5)을 가지고 있는 2차원 배열을 사용하여 ADF ,CHAMELEON, EF-CHAMELEON,의 성능을 비교하였다.

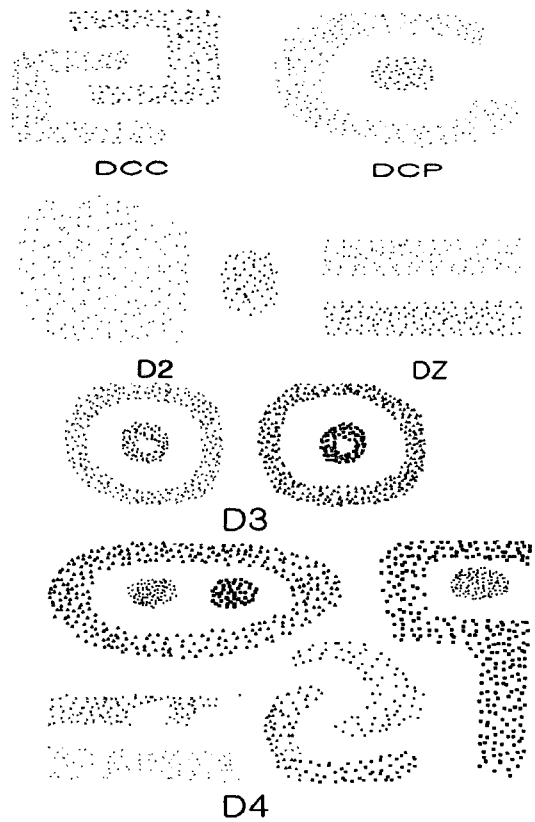
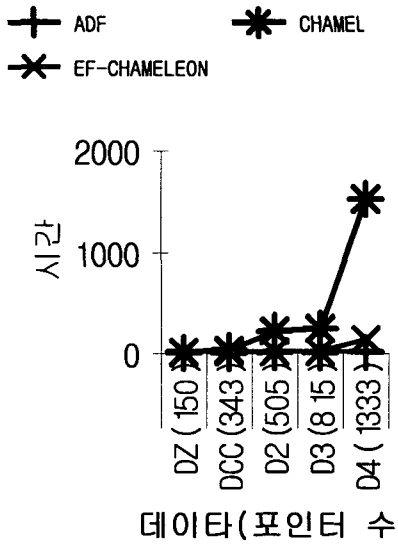


그림 5

<표1>데이터들을 정확히 구분하기 위해 걸리는 시간

데이터(수) / 방 식	DZ(150)	DCC(343)	D2(505)	D3(816)	D4(1333)
ADF	K=25 시간:1.9	K=25 시간:2.2	K :50 시간 :4.2	K :80 시간 :6.3	K :100 시간:10.6
EF_ CHAMELEON	sample size 30 파티션 :10 합병 :2 시간 :4.5	sample size 50 파티션 :30 합병 :2 시간 :17.5	sample size 100 파티션 :20 합병 :2 시간 :15.9	sample size 80 파티션 :20 합병 :4 시간 :13.7	sample size 300 파티션 :80 합병 :9 시간 :143.9
CHAMELON	파티션 :20 합병 :2 시간 :10.4	파티션 :35 합병 :2 시간 :32.6	파티션 :100 합병 :2 시간 :227.6	파티션 :100 합병 :4 시간 :249.4	파티션 :250 합병 :9 시간 :1528.5



6.결론

실험결과 ADF알고리즘이 기존의 알고리즘이 가지고 있는 고정된 모양의 문제점을 해결하였으며 또한 CHAMELEON[1] 알고리즘보다 우수함을 알 수 있었다. CHAMELEON[1]이 초기 데이터들을 분할하기 위하여 그래프 분할 알고리즘(hMeTis[6])을 사용하고 있다. 하지만 본 논문에서 데이터를 분할하기 위해 사용한(K-Means[7])알고리즘이 더 적은 분할로도 각 고유(노이즈에 영향이 적은)의 특성을 가지는 초기 클러스터로 분할함을 알 수 있었다.

6.참고 문헌

[1] G. Karypis , E. H. Han, V.Kumar"CHAMELEON:A Hierarchical Clustering Algorithm Using Dynamic Modeling", IEEE Computer 1999.

[2] M.Ester, H. Kriegel, J.Sander, and X.Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise",Proc1996.

[3] T.Zhang, R.Ramakrishnam, M.Livny, "BIRCH : An Efficient Data Clustering Method for Very Large Database", In Proc. of ACM SIGMOD International Conferance on Management of Data, 1996.

[4] S.Guha, R.Rastogi, and K. Shim "CURE: An efficient clustering algorithm for large databases. In Proc. of 1998 ACM-SIGMOD Int. Conf. on Management of Data, 1998.

[5] S.Guha, R.Rastogi, and K. Shim "ROCK:a robust clustering algorithm for categorical attributes. In Proc. of the 15th Int'l Conf.on Data Eng.,1999.

[6] G.Karypis, and V.Kumar "hMETiS 4.0: Unstructured graph partitioning and sparse matrix ordering system. Technical report, Department of Computer Science, University of Minnesota,1998. Available on the WWW at URL http://www.cs.umn.edu/~metis.

[7] A.K. Jain and R .C. Dubes. Algorithms for clustering Data. Prentice Hall, 1988.

[8] L. Kaufamn and P.J. Rousseeuw, Finding Groups in Data: An INtroduction to Cluster Analysis, John Wiley & sons,1990

[9] G.H. Ball and D.J Hall. Some fundamental concepts and sysnthesis procedures for patter n recognition preprocessors. In International Conference on Microwaves, Circuit Theory, and Information Theory, 1964

[10] R. Ng and J. Han, " Efficient and Effective Clustering Method for Spatial Data Mining," Proc. of Int. Conf . on VLDB, pp.144-155, 1994