Proceedings of Joint Conference of Korean Data And Information Science Society and The Korean Data Analysis Society, April 30 - May 1, 2004, pp. 51-60

K-means Clustering using a Center Of Gravity for grid-based sample

Hee Chang Park[1], *Sun Myung Lee*[2]

Abstract

K-means clustering is an iterative algorithm in which items are moved among sets of clusters until the desired set is reached. K-means clustering has been widely used in many applications, such as market research, pattern analysis or recognition, image processing, etc. It can identify dense and sparse regions among data attributes or object attributes. But k-means algorithm requires many hours to get k clusters that we want, because it is more primitive, explorative. In this paper we propose a new method of k-means clustering using a center of gravity for grid-based sample. It is more fast than any traditional clustering method and maintains its accuracy.

Keywords : data mining, k-means clustering, grid-based sampling

1. 서론

데이터 마이닝(data mining)은 대량의 데이터로부터 알려지지 않은 유용한 정보를 추출하는 과정이다. 정보를 추출하기 위한 데이터 마이닝의 기법에는 여러 가지 기법이 활용되고 있는데, 이들 중에서 클러스터링은 다양한 특성을 지닌 관찰대상을 유사성을 바탕으로 동질적인 집단으로 분류하는데 쓰이는 기법이다. 이 기법은 단지 개체간의 유사성 또는 거리에 의하여 군집을 형성하고, 형성된 군집의 특성을 파악하여 군집들 사이의 관계를 분석하는 기법이다. 클러스터링의 기법에는 분할 군집법, 계층적 군집법, 밀도에 의한 군집법, 그리드에 의한 군집법, 모형에 근거한 군집법 등이 있다. 그 중에서 분할 군집법은 데이터들을 임의의 부분집합으로 분할을 한 후 데이터들을 유사한 그룹으로 재배치하는 군집방법이다.

분할 군집법의 종류에는 본 연구에서 고려한 k-means 알고리즘과 k-medoids 알고리즘, k-prototypes 알고리즘, k-modes 알고리즘 등이 있다. k-means 알고리즘은 MacQueen(1967)에 의해 처음 소개되었으며, 데이터들을 k개의 군집으로 임의로 분할을 하여 군집의 평균을 대표값으로 분할

[1]Professor, Department of Statistics, Changwon National University, Changwon, Kyungnam, 641-773, Korea. E-mail : hcpark@sarim.changwon.ac.kr

[2]Graduate Student Department of Statistics, Changwon National University, Changwon, Kyungnam, 641-773, Korea

해 나가는 방법으로 데이터들을 유사성을 바탕으로 재배치를 하는 방법이다. Kaufman과 Rousseeuw(1990)는 k-means 알고리즘이 이상값에 민감한 것을 보완하여 군집의 대표값을 중위수로 하는 k-medoids 방법인 PAM(partitioning around medoids)과 CLARA(clustering large applications) 알고리즘을 제안하였다. Ng 등(1995)은 CLARA를 더욱 향상시킨 CLARANS(clustering large applications based on randomized search)를 제안하였다. CLARA 알고리즘이 조사의 각 단계에서 고정된 표본을 가지는 반면에 CLARANS는 조사의 각 단계에서 임의의 표본을 가지며, 이상점을 발견할 수도 있다. Huang(1997, 1998)은 k-means가 연속형 데이터에 대해 한정된 단점을 보완하여 연속형과 범주형의 혼합된 데이터에 대한 k-prototypes 알고리즘과 범주형 데이터에 대한 k-modes 알고리즘을 제안하였다. Chu 등(2002a, b)은 k-medoids가 이상점에 강한 반면 수행속도가 느리다는 약점을 보완하기 위해 효과적인 샘플링을 기법을 추가한 MCMRS(Multi-Centroid, Multi-Run Sampling Scheme) 및 IMCMRS(Incremental Multi-Centroid, Multi-Run Sampling Scheme) 알고리즘을 제안하였다.

데이터 마이닝은 대용량의 데이터베이스를 분석대상으로 하므로 그 만큼 데이터 처리 시간을 단축하기 위해 많은 방법이 연구되고 있다. 특히 웹이 보편화된 현재 사용자들의 다양한 패턴을 분석하기 위한 데이터 마이닝 방법이 사용되어지고 있는데 처리 속도 문제는 더욱 중요하게 생각하고 있다. 이러한 속도 문제를 해결하기 위해 본 논문에서는 분할 군집법에서 가장 일반적으로 사용되고 있는 k-means 알고리즘에 대해 그리드를 기반으로 한 무게중심 알고리즘을 제안하고자 한다. 2절에서는 그리드 기반 무게중심을 이용한 k-means 알고리즘을 구현하며, 3절에서는 예제와 실험을 통하여 본 연구에서 제시한 기법과 기존의 기법을 비교하여 수행속도와 정확도에서 만족할 만한 수준의 결과가 얻어짐을 확인하고자 한다. 마지막으로 4절에서 본 연구의 결론을 맺고자 한다.

2. 그리드 기반 무게중심의 k-means 군집방법

k-means 군집방법은 데이터들을 k개의 군집으로 임의로 분할하여 군집의 평균을 대표값으로 분할해 나가는 방법으로 데이터들을 유사성을 바탕으로 재배치를 하는 방법이다. 이 방법도 대량의 데이터를 대상으로 계산을 수행하므로 상당한 처리시간이 소요된다. 이러한 속도 문제를 해결하기 위해 박희창 등(2003)은 그리드를 기반으로 한 한점 샘플링 방법을 고려하였다. 하지만 그리드 기반 한 점 샘플링은 샘플링 된 데이터로 k-means를 수행하였으므로 기존의 방법에 비해 군집의 정확성이 떨어지는 문제점을 가지고 있었다. 특히 한 점을 샘플링 했을 때 그리드내의 이상점이 샘플링 되면 전혀 다른 그리드와 군집이 될 수도 있는 문제점을 가지고 있다. 이러한 문제점의 해결방안으로 그리드별 세 점을 샘플링하여 그 세 점의 무게중심을 이용하는 방법을 제안하고자 한다. 그리드에서 한 점을 샘플링 하는 것보다는 다소 시간이 걸리겠지만 정확도는 상당히 향상시킬 수가 있을 것이다.

2.1 유사성 측정도구

군집분석에서 군집간의 유사성 측정은 거리로써 나타낸다. 서로 다른 개체 사이의 거리 $d_{ij}=d(X_i, X_j)$를 구하는 방법에는 유클리디안 거리, 유클리디안 제곱거리, 마할라노비스 거리, 그리고 민코우스키 거리 등이 있으며, 본 논문에서는 식(2.1)의 유클리디안 제곱거리를 이용하고자 한다.

$$d_{ij}=\sum_{k=1}^{p}(t_{ik}-t_{jk})^2 \tag{2.1}$$

여기서 $d_{ij}\geq 0,\ d_{ii}=0,\ d_{ij}=d_{ji}\ d_{ik}+d_{jk}\geq d_{ij}$ 이다.

2.2 그리드의 설정

본 논문에서는 클러스터링의 수행과정을 최소화하기 위해 샘플링 이전에 그리드를 사용하여 데이터 개체들을 적당한 그리드 간격으로 분할한다. 본 논문에서 제시하는 알고리즘의 그리드 간격을 *GI*(Grid Interval)라고 할 때 *GI*는 식 (2.2)와 같이 설정한다.

$$GI_v = \frac{\max_v - \min_v}{n^{\frac{1}{p}}} \tag{2.2}$$

여기서 v는 v번째 변수를 나타내며, $\max_v$와 $\min_v$는 각각 v번째 변수의 최대값과 최소값을 나타낸다. 이러한 그리드 간격은 데이터들이 골고루 분포되어 있다는 가정하에서 한 그리드내에 반드시 하나의 데이터를 배치하기 위한 방법이다.

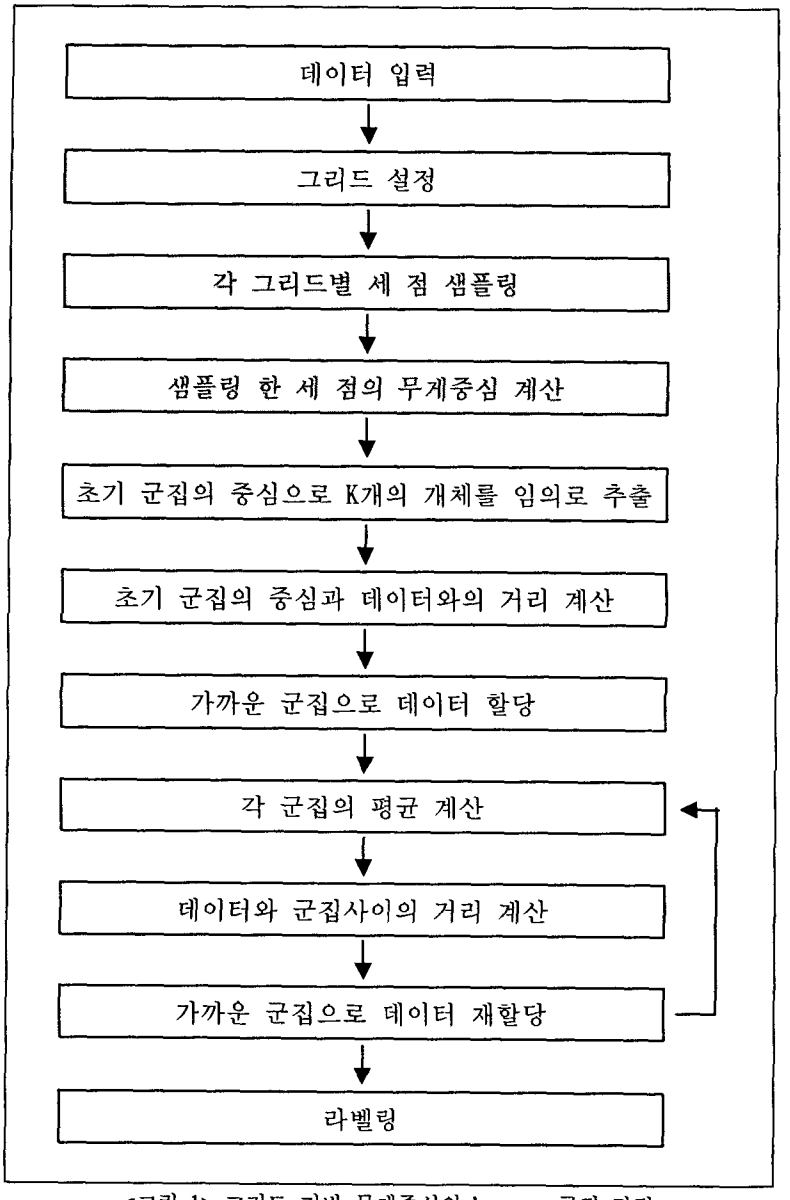
2.3 그리드 무게중심

각 그리드별로 세 점을 단순임의추출 한다. 추출한 세 점의 무게중심을 구하여 클러스터링을 수행한다. 본 논문에서는 무게중심을 *COG*(Center Of Gravity)으로 표기한다. 무게중심을 구하는 식은 식 (2.3)과 같다.

$$COG=\left(\frac{x_{11}+x_{12}+x_{13}}{3},\ \frac{x_{21}+x_{22}+x_{23}}{3},\ \cdots\right) \tag{2.3}$$

2.4 알고리즘 구현

그리드 기반 무게중심의 k-means 군집분석을 위한 수행 과정은 다음과 같다.



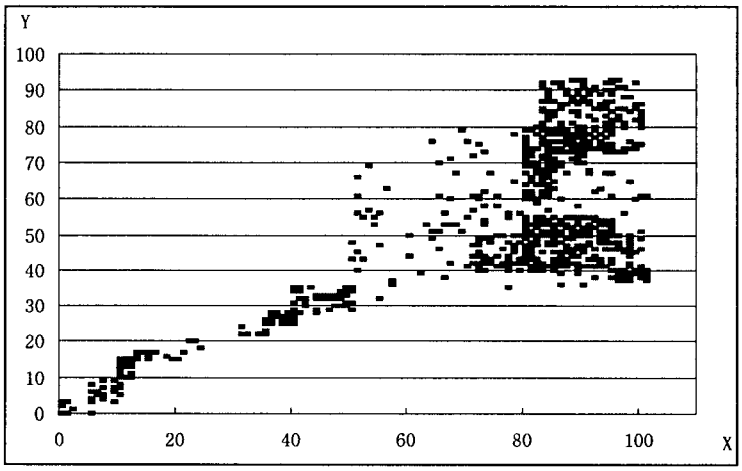
<그림 1> 그리드 기반 무게중심의 k-means 군집 과정

3. 예제 및 모의실험

앞 절에서 구현한 그리드 기반 무게중심에 의한 k-means 알고리즘을 바탕으로 수행 시간 및 정확도를 비교하기 위하여 실험을 실시하였다. 본 실험의 구현환경은 다음과 같다.

CPU : Intel Pentium4-1.8GHz Northwood
RAM : 512MB
O/S : Microsoft Windows XP Professional
Language : JAVA J2SDK 1.4.0
Database : MySQL 3.23.51 (External Linux Server)

실험을 위해 변수의 범위가 각각 0 ~ 101, 0 ~ 93인 두 개의 변수로 이루어진 1000건의 데이터를 랜덤하게 발생시켜 기본 데이터 셋으로 사용하였다. 이들의 편균 및 표준편차는 각각 $\overline{X}$ = 75.7, S_x = 24.9 $\overline{Y}$ = 53.9, S_y = 22.0이다. 실험은 기본 데이터 셋에서 랜덤 샘플링하여 사용을 하였다. 데이터 셋의 분포는 <그림 2>와 같다.



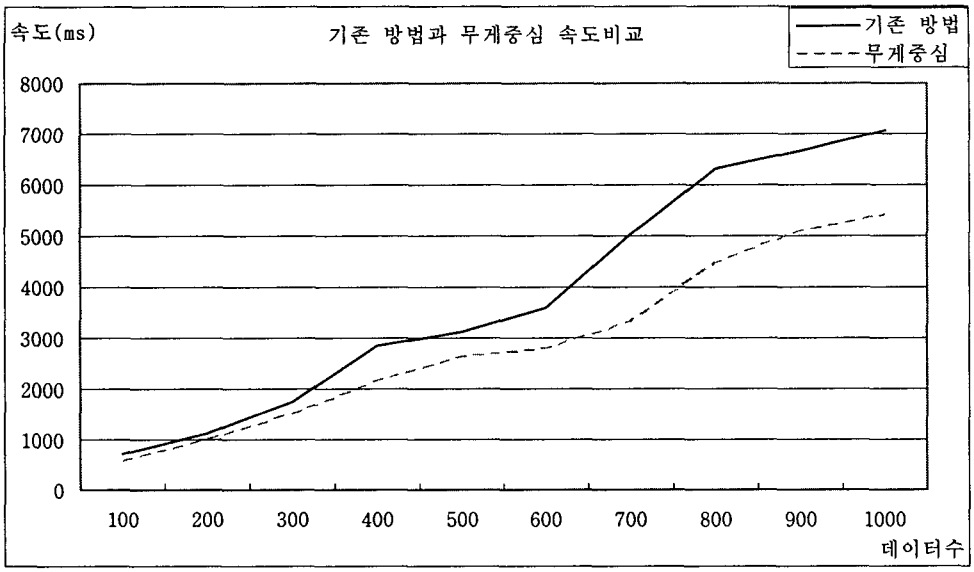
<그림 2> 데이터 셋의 분포도

그리드를 기반으로 세 점을 샘플링하여 그 무게중심으로 k-means를 수행하였을 때 수행 속도와 정확도는 <표 1>과 같다.

<표 1> 그리드 기반 무게중심 수행 속도와 정확도

데이터 수	속도(ms)	정확도
100	681	99.0%
200	1102	97.5%
300	1321	97.7%
400	2003	99.0%
500	2634	98.8%
600	2784	99.0%
700	3305	99.0%
800	4467	98.6%
900	5097	98.2%
1000	5398	98.2%

기존의 방법과 그리드 기반 무게중심에 의한 방법의 수행 속도를 비교한 결과는 <그림 3>과 같다.



<그림 3> 기존 방법과 그리드 기반 무게중심에 의한 방법의 수행 속도

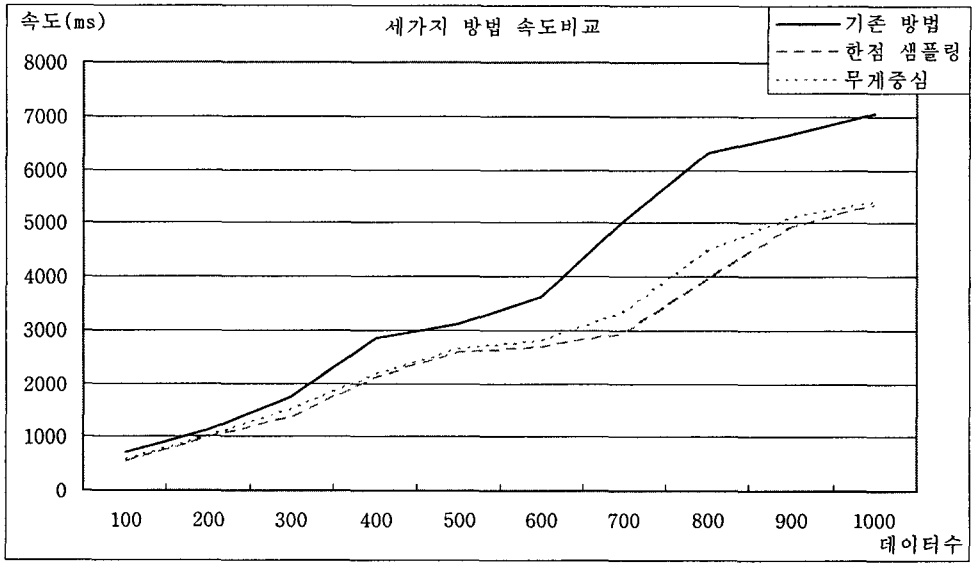
위의 결과에서 보는 바와 같이 기존의 방법에 비해 그리드 기반 무게중심이 속도에 있어서 효과적인 것을 알 수 있다. 사전연구의 그리드 기반 한 점 샘플링 k-means 방법과 무게중심의 속도와 정확도에 대해 비교를 해본 결과 <표 3>과 같다.

<표 3> 기존 방법, 한 점 샘플링과 무게중심의 속도와 정확도

데이터 수	기존 방법	한 점 샘플링		무게중심	
	속도(ms)	속도(ms)	정확도	속도(ms)	정확도
100	721	541	99.0%	551	99.0%
200	1112	971	97.5%	1012	97.5%
300	1742	1372	96.7%	1502	97.7%
400	2854	2113	98.5%	2164	99.0%
500	3115	2584	98.8%	2634	98.8%
600	3606	2663	99.0%	2784	99.0%
700	5038	3465	99.0%	3305	99.0%
800	5658	4336	98.6%	4397	98.6%
900	6659	4908	98.1%	5097	98.2%
1000	7051	5348	98.2%	5398	98.2%

<표 3>을 보면 수행 속도는 한 점 샘플링이 가장 빠른 것을 알 수 있으며, 정확도는 한 점 샘플링에 비해 무게중심이 더 정확한 것을 알 수 있다.

기존 방법과 한 점 샘플링, 그리고 무게중심의 수행 속도를 비교한 결과는 <그림 7>과 같다.



<그림 7> 기존 방법, 한 점 샘플링과 무게중심 수행 속도

위 <그림 7>에서 알 수 있듯이 기존 방법에 비해 한 점 샘플링과 무게중심의 수행 속도가 훨씬 빠르고, 한 점 샘플링과 무게중심의 속도는 크게 차이가 나지 않는 것을 알 수 있다. 이러한 결과로 볼 때 정확도가 더 높은 그리드 기반 무게중심이 효과적인 것을 알 수 있다.

위의 모의 실험에서 검증된 결과를 바탕으로 실제 데이터에 적용을 시킬 수가 있다. 창원 모 중학교 학생들의 신체검사 결과인 실제 데이터를 사용하여 속도를 비교해 보았다. 데이터의 특징은 다음과 같다.

데이터 수 : 1,734 건
사용 변수 :
X = 키
Y = 몸무게
$\overline{X}$ = 159.6, S_x = 8.4, $\overline{Y}$ = 50.9, S_y = 10.8

실험한 결과, 기존의 방식은 27,720ms, 무게중심은 11,016ms로 나타났다. 이 예제에서도 기존의 방법에 비해 그리드 기반 표본에 의한 방법이 속도 면에서 뛰어난 것을 알 수 있다.

5. 결론 및 향후 과제

아무리 정확하고 유용한 정보라 할지라도 과거의 정보는 흘러간 정보이기 때문에 데이터 마이닝에서 처리 속도 문제는 해결해야 할 과제 중의 하나이다. 이러한 속도 문제를 해결하기 위해 본 논문에서는 분할 군집법에서 가장 일반적으로 사용되고 있는 k-means 알고리즘에 대해 그리드 기반 무게중심을 이용한 샘플링 알고리즘을 제안하였다. 동시에 본 연구에서 제시한 기법과 기존의

기법을 실험 및 예제를 통하여 비교하였으며, 수행속도와 정확도에서 만족할 만한 수준의 결과가 얻어짐을 확인하였다.

<부록> 그리드 기반 무게중심의 k-means 알고리즘

```
/* k = 군집의 수, n = 데이터 수, p = 변수 수
 p = 2 인 경우의 알고리즘
 변수 x, y
*/

clustering()
{
 int k, n, p;
 float x[], y[];

 InsertData();

 tmpMaxx = 0, tmpMaxy = 0;
 while(i <= n)
 {
 if(tmpMaxx <= x[i]) tmpMaxx = x[i];
 if(i = 1) tmpMinx = x[i];
 if(tmpMinx >= x[i]) tmpMinx = x[i];

 if(tmpMaxy <= y[i]) tmpMaxy = y[i];
 if(i = 1) tmpMiny = y[i];
 if(tmpMiny >= y[i]) tmpMiny = y[i];
 }

 Maxx = tmpMaxx , Minx = tmpMinx;
 Maxy = tmpMaxy , Miny = tmpMiny;

 GIx = (Maxx - Minx) / n^(1/p) ;
 GIy = (Maxy - Miny) / n^(1/p) ;
stop = 1, cx = 0, tmpLx = Minx;
while(stop == 1) {
 if(tmpLx <= Maxx)
 {
 cx++;
 GIx_L[cx] = tmpLx + GIx;
 }
 else stop = 0;
}
stop = 1, cy = 0, tmpLy = Miny;
while(stop == 1) {
 if(tmpLy <= Maxy)
 {
 cy++;
 GIy_L[cy] = tmpLy + GIy;
 }
 else stop = 0;
}
```

```
giNo = 0;
while(xi <= cx) {
 while(yi <= cy)
 {
 giNo++;
 while(i <= n)
 {
 if((GIx_L[xi] <= x[i] < GIx_L[xi + 1]) &&
 (GIy_L[yi] <= y[i] < GIy_L[yi + 1]))
 {
 GI_N[i] = giNo;
 }
 }
 }
}
s = 0;
while(s <= giNo)
{
 s++;
 triSample[s] = rand(3);
}

s = 0;
while(s <= giNo)
{
 s++;
 triSum = Sum(triSample[s]);
 gravity[s] = triSum / 3;
}

k_means();

s = 0;
while(s <= giNo)
{
 while(i <= remainData)
 {
 if(s == GI_N[i])
 cluster[i] = cluster[s];
 }
}
```

참고문헌

1. MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations." in *5th Berkeley Symp. Math. statist, Prob.* 1, 281-297.

2. Kaufman, L. and Rousseeuw, P.J. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis.* New York: John Wiley & Sons.

3. Ng, R. and Han, J. (1994). "Efficient and effective clustering method for spatial data mining." in *Very Large Data Bases (VLDB'94).* 144-155.

4. Huang, Z. (1997). "Clustering Large Data Sets with Mixed Numeric and Categorical Values." In *Proceedings of The First Pacific-Asia Conference on Knowledge Discovery and Data Mining, Singapore, World Scientific.*

5. Huang, Z. (1998). "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining." *Data Mining and Knowledge Discovery.*

6. Chu, S.C., Roddick, J.F., and J.S. Pan (2002a). " Efficient k-medoids algorithms using multi-centroids with multi-runs sampling scheme." in *Workshop on Mining Data for CRM,* (Taipei, Taiwan), Springer, 2002.

7. Chu, S.C., Roddick, J.F., and J.S. Pan (2002b). "An Incremental Multi-Centroid, Multi-Run Sampling Scheme for k-medoids-based Algorithms-Extended Report."

8. Han, J. (2001). "Data Mining:Concepts and Techniques".