데이터 수정 접근에 의한 불완전한 수치형 데이터 분류에 관한 연구

김요승*, 정영철*, 이원돈*
*충남대학교 컴퓨터공학과
e-mail : kingrise@chol.com

A Study on Classifying Numerical Incomplete Data with Data Reparation Approach

Yo-Seung Kim*, Young-Chul Jung*, Won-Don Lee*
*Dept. of Computer Science & Engineering, Chung-Nam National University

요 약

분류는 기계학습에서 매우 중요한 연구주제이다. 그 중에서도 수치형 데이터의 분류를 위한 많은 알고리즘들이 있다. 그러나 불완전한 데이터의 존재는 분류 모델들의 학습(learning) 품질(quality)을 떨어뜨린다. 그 불완전한 데이터는 현실 세계에서 아주 흔하다. 학습 단계와 분류 단계 양쪽에서 불완전한 데이터를 다루는 것이 중요하고 현실세계 문제들을 풀기 위해 적용되는 것이 필요하다. 본 논문에서 Optimal Completion Strategy(OCS)로부터 나온 몇 개의 공식들이 불완전한 데이터를 예측하기 위해 사용되었다. 새로운 방법이 불완전한 데이터를 분류하기 위해서 제시되었고, 그것은 놀라운 성능을 보여준다.

1. 서론

분류는 수년 동안 연구되어 오고 있는 매우 중요한 문제이다. 그러나 불완전한 데이터의 존재는 항상 분류 모델들의 품질을 떨어뜨린다. 분실 데이터의 정의를 더 직관적으로 보여주기 위해서, 예를 들어보면: 만약 X1 이 (1, 2, 3, 4)인데 (?, 2, 3, 4)만 확인되면 25%의 불완전한 데이터이고, (1, ?, ?, 4)이면 50%의 불완전한 데이터이다. 분류 문제는 두 단계로 나누어 질 수 있다: 학습 단계와 분류 단계. 학습 단계는 훈련 데이터 세트로부터 분류 모델을 구축하는 것이고, 분류 단계는 이미 정의된 클래스들로부터 알려지지 않은 경우들을 분류하는 것이다. 분류 문제에서 불완전한 데이터를 다루는 많은 방법들이 제시되었지만 그들 대부분은 단지 학습을 위한 훈련 세트에 있는 불완전한 데이터를 다루는 프로세싱에 초점이 맞춰져 있다. 분류 단계에서 나타나는 불완전한 값에 대해 거의 모든 현재의 접근들이 효과가 있을 수 없다. 의사결정나무 분류자는 그들의 주어진 속성 값들을 근거로 데이터의 클래스를 예측할 수 있다. 이 방법론은 많은 응용 분야에서 성공적으로 사용되고 있다.

학습 단계와 분류 단계에서 자연스럽게 불완전 데이터 분류 문제를 풀기 위해서 새로운 의사결정나무 방법이 제시된다. 그 제시된 방법은 불완전 데이터 분류 문제를 매우 잘 풀 수 있다. 불완전 데이터 분류 문제뿐만 아니라, 그 제시된 방법은 다른 중요한 문제를 풀 수 있다: 규칙 개선 문제(rule refinement problem). 개선 문제는 훨씬 더 개선된 의사결정나무를 생성하기 위해서 새로 들어오는 데이터를 의사결정나무에 더하는 것이다. 3 장에서 이 문제와 해결책을 자세하게 논의할 것이다.

4 장에서 실험 결과들이 설명되어 있고, 그 결과들은 그 제시된 방법이 매우 좋은 성능을 나타내고 있음을 보여준다

2. 불완전 데이터 분류 문제에 대한 관련 연구

J. Ross Quinlan[1]에 의해 개발된 유명한 C4.5 는 불완전 데이터를 무시하는 대표적인 알고리즘이다. C4.5 는 학습 단계와 분류 단계에 있는 불완전 데이터 분류 문제에 효과가 있는 유일한 알고리즘이다. 그러나 C4.5 는 불완전한 데이터를 다룰 때 그렇게 효과적이지는 못하다. C4.5 결과는 실험에서 제시된 방법과 비교해서 보여줄 것이다.

C4.5 는 불완전 데이터에 대해 많은 유용한 정보들을 낭비한다. 왜냐하면 이득을 계산하는 동안 불완전 데이터를 가진 이벤트를 무시하므로, 이 이벤트에 있는 데이터를 가진 속성들의 정보를 낭비한다.

.<표 1> 훈련 데이터 세트 예제

이벤트#	속성 1	속성 2	속성 3	클래스
1	5.1	?	1.3	클래스 2
2	?	3.5	1.4	클래스 2
3	4.7	3.2	?	클래스 1
4	4.7	3.0	?	클래스 1
5	5.1	?	1.3	클래스 1
6	?	3.5	1.4	클래스 1
7	4.9	?	1.3	클래스 2

3. 수치형 불완전데이터를 위한 새로운 의사결정나무

3.1 불완전한 수치형 데이터에 대한 새로운 분류자

표기법은 다음과 같다:

{

$X_i = (X_{i1}, \cdots X_{ik}, \cdots X_{ip})$ $(i = 1, \cdots, n)$, X_i 는 i 번째 p-차원 데이터 벡터이고, 범위는 0<i<n 이다.

$X = (X_1, \cdots X_i, \cdots X_n)$는 완전한 데이터 세트이다.

$X_L = (X_i \in X)$, X_i 는 불완전한 데이터이다.

$Y_i = (Y_{i1}, \cdots, Y_{im}, \cdots Y_{ig})$, Y_{im} 는 m 번째 클러스트에 속하는 i 번째 확률이고, 이 확률은 클러스터 총수로 정의된다. 클러스터 수는 g 이다.

$C_m = (C_{m1}, \cdots, C_{mp})$ $(m = 1, \cdots, g)$, m 번째 클러스터의 중심

}

Optimal Completion Strategy (OCS) 알고리즘은 다음과 같이 요약된다.

Initialize matrix.

Step 1) calculate

$$E(g, r, Y)^{t+1} = \sum_{i=1}^{n} \sum_{m=1}^{g} (Y_{im}{}^{t+1})^r \sum_{k=1}^{p} (X_{ik} - (C_{mk}{}^{t+1}))^2 \quad (1)$$

$$C_{mk}{}^{t+1} = \sum_{i=1}^{n} ((Y_{im}{}^{t+1})^r * X_{ik}) \Big/ \sum_{i=1}^{n} (Y_{im}{}^{t+1})^r$$

$$(m = 1, \cdots, g; k = 1, \cdots, p) \quad (2)$$

$$Y_{im} = \frac{D_{im}{}^{1/(1-r)}}{\sum_{j=1}^{g} D_{im}{}^{1/(1-r)}} \quad (3)$$

Step 2) Compare if

$$(\| \boldsymbol{Y}^{t+1} - \boldsymbol{Y}^{t} \| < \varepsilon) \quad (4)$$

then stop.

Else go to Step 3.

Step 3) for all $X_{mk} \in X_L$, calculate

$$X_{mk}{}^{t+1} = \sum_{i=1}^{g} ((Y_{im}{}^{t+1})^r * C_{ik}) \Big/ \sum_{i=1}^{g} (Y_{im}{}^{t+1})^r \quad (5)$$

Go to Step 1.

예를 들어, 표 1 을 수정하면 표 2 가 된다.

인지하듯이 테스트 데이터 세트 안에 불완전한 데이터는 위의 보상 방법으로 수정될 수 있다. 그래서 불완전 데이터를 가진 테스트 데이터 세트를 분류하는 것은 문제가 되지 않는다.

<표 2> 수정된 훈련 데이터 세트 예제

이벤트#	속성 1	속성 2	속성 3	클래스
1	5.1	3.5	1.3	클래스 2
2	5.1	3.5	1.4	클래스 2
3	4.7	3.2	1.3	클래스 1
4	4.7	3.0	1.4	클래스 1
5	5.1	3.0	1.3	클래스 1
6	4.7	3.5	1.4	클래스 1
7	4.9	3.5	1.3	클래스 2

불완전 데이터 분류 문제에 대한 답을 얻기 위하여 모든 유용한 정보들을 남겨두려면 새로운 데이터 표현이 표 3 처럼 제시된다. 이 데이터 표현에서 각 행은 원래 이벤트로부터 변환된다. 그리고 그 이벤트가 얼마나 중요한지를 보여주는 각 이벤트에 대한 가중치가 있다. 예를 들어, 가중치가 20 인 이벤트는 1 의 가중치를 가진 20 개의 이벤트들과 동일한 중요도를 가진다. 이것은 하나의 값을 한 이벤트의 특정한 속성에 줄 필요가 없지만 한 속성에 가능성으로 여러 의견을 주는 것을 가능하게 한다는 것을 의미한다. 이런 경우에, 모든 훈련 데이터로부터 우리는 주어진 하나의 속성에서 불완전한 데이터를 위한 상응하는 가능성을 알게 된다. 주어진 특정한 속성과 클래스 값으로부터 이 속성의 각각의 가능한 값의 수치가 계산되고, 그리고 이 클래스 값에 대한 이벤트들의 총수로 나누어진다. 그리고 나서 불완전 데이터에 대한 상응하는 수치를 할당한다.

<표 3> 표 2 으로 부터 새로운 데이터 표현

E#	W	속성 1			속성 2			속성 3		클래스	
		5.1	4.9	4.7	3.5	3.0	3.2	1.4	1.3	1	2
1	1	1	0	0	1	0	0	0	1	0	1
2	1	1	0	0	1	0	0	1	0	0	1
3	1	0	0	1	0	0	1	0	1	1	0
4	1	0	0	1	0	1	0	1	0	1	0
5	1	1	0	0	0	1	0	0	1	1	0
6	1	0	0	1	1	0	0	1	0	1	0
7	1	0	1	0	1	0	0	0	1	0	1

* E: Event(이벤트), W: Weight(가중치)

이 데이터 표현을 사용하면 새로운 식들은 하나의 새로운 의사결정나무를 세우는데 사용된다.

그 정의들과 식들은 다음과 같다:

클래스 총 수의 가중치

: $C_1(m), C_2(m), \ldots\ldots C_{k-1}(m), C_k(m)$.

$C_i(m)$ 는 얼마나 많은 m 번째 이번트가 클래스 C_i 에 속하는지를 나타낸다.

여기서, $\sum_{i=1}^{k} \boldsymbol{C_i(m)} = 1$

결과 총 수의 가중치

: $O_{A1}(m), O_{A2}(m), \ldots\ldots O_{A(n-1)}(m), O_{An}(m)$

$O_{Aj}(m)$: 속성 A 에서 얼마나 많은 결과 값 j 가 m 번째 이벤트에서 일어날 수 있는 지를 보여주는 값

여기서, $\sum_{j=1}^{n} \boldsymbol{O_{Aj}(m)} = 1$

U_{Aj} : 속성 A 에 대한 결과 값 j 를 갖는 U 의 서브세트. 예를 들어, 만약 속성 1 이 5.1, 4.9, 4.7 의 세 개 값을 갖는다면, 세트 U 는 3 개의 서브세트들로 나누어진다.

n(u): 세트 U 에 있는 이벤트들의 수

Weight(m,U) : 세트 U 에서 m 번째 이벤트의 가중치

freq(C_i, U) : C_i 의 클래스 값을 갖는 세트 U 에서 이벤트들의 수

freq(C_i, U_{Aj}) : C_i 의 클래스 값을 갖는 세트 U_{Aj} 에서 이벤트들의 수

이들 새로운 정의를 사용하면 엔트로피 식들이 다음과 같이 유도될 수 있다:

$$\text{info(U)} = -\sum_{i=1}^{k}\Big(\sum_{m=1}^{n(u)} Weight(m,U)\cdot C_i(m)/|U|\Big) \cdot \log_2(\text{freq}(C_i,U)/|U|) \quad (6)$$

$$\text{info}(U_{Aj}) = -\sum_{i=1}^{k}\Big(\sum_{m=1}^{n(u)} Weight(m,U)\cdot C_i(m)\cdot O_{Aj}(m)/|U_{Aj}|\Big) \cdot \log_2(\text{freq}(C_i,U_{Aj})/|U_{Aj}|) \quad (7)$$

|U|: 세트 U 에서 이벤트들의 수. 하나의 이벤트는 가중치 1 을 가진 이벤트를 의미한다. 그래서 하나의 이벤트가 가중치 X 를 갖는다는 것은 속성들과 클래스의 동일한 분포 값들을 갖는 이벤트들의 수가 X 개 있다는 것을 의미한다.

$|U_{Aj}|$: 세트 U_{Aj} 에서 이벤트들의 수

$$|U_{Aj}| = \sum_{m=1}^{n(u_{Aj})} Weight(m,U_{Aj})\cdot O_{Aj}(m) \quad (8)$$

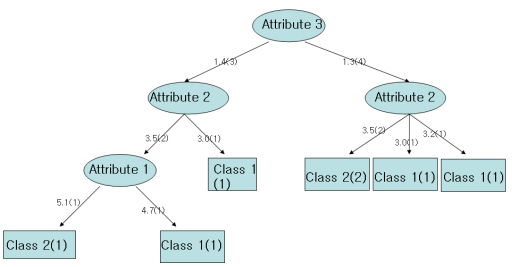
식(8)에 보여진 것처럼, $|U_{Aj}|$는 $O_{Aj}(m)$와 $weight(m,U_{Aj})$ 을 곱하고, 모든 곱들의 결과를 더함으로 계산될 수 있다. $O_{Aj}(m)$는 U_{Aj} 에서 각 이벤트 결과의 가능성 값을 위한 것이고 $weight(m,U_{Aj})$는 각 m 을 위한 것이다. 그러면 이득 비율을 계산하기 위한 식은 다음 식들과 같다:

$$\text{Split\_info(A)} = -\sum_{j=1}^{n(u)}(|U_{Aj}|/|U|)\cdot\log_2(|U_{Aj}|/|U|) \quad (9)$$

$$\text{Gain(A)} = \text{info(U)} - \text{info}_A(U_{Aj}) \quad (10)$$

$$\text{Gain\_ratio(A)} = \text{Gain(A)} / \text{Split\_info(A)} \quad (11)$$

따라서, 그 노드에서 가장 큰 이득 비율을 갖는 속성이 위 식들을 사용해서 결정될 수 있다. 의사결정나무는 그림 1 처럼 생성된다.



(그림 1) 표 1 로부터 구성된 새로운 의사결정나무

3.2 규칙개선문제를 풀기 위한 새로운 분류자

의사결정나무로부터 생성된 규칙은 원래 데이터 보다 더 간단하고 의사결정나무 만큼이나 정확한 예측을 한다는 것은 잘 알려져 있다. 몇 개의 새로운 데이터로 구성된 결정나무에 추가함으로써 새로운 의사결정나무를 어떻게 세우느냐 하는 것이 규칙 개선 문제이다. 새로운 데이터를 원래 데이터에 추가하고, 그리고 이 모든 데이터로부터 새로운 나무를 세우는 것이 필요한가 ? 결국, 훈련은 시간을 소비하는 과정이다. 그리고 그것은 모든 원래 데이터를 저장하기 위한 많은 저장장치에 돈이 많이 들어간다. 제시된 방법에서는 규칙 개선 문제가 매우 잘 해결되었다.

그림 1 에 있는 나무가 남아있는 유일한 정보이고 그 원래 데이터가 저장되어 있지 않다고 가정하자. 그림 1 로부터 규칙을 생성하는 과정은 이와 같다: 만약 가장 왼쪽 종단 노드가 선택되면, 종단 노드에서 시험 노드를 따라 올라가라. 속성 3, 속성 2, 그리고 속성 1 이 선택된 것을 발견할 수 있다. 속성 1 에 대해서 5.1 은 1 이 되도록 할당되고, 4.9 와 4.7 은 0 이 되도록 할당됐다. 속성 2 에 대해서는 3.5 가 1 이 되도록 할당되고, 3.0 와 3.2 는 0 이 되도록 할당되었다. 속성 3 에 대해서는 1.4 가 1 이 되도록 할당되고, 1.3 은 0 이 되도록 할당되었다. 그리고 클래스에 대해서는 클래스 1 은 0 이 되도록 할당되었고, 클래스 2 는 1 에 할당되었다. 그래서 종단 노드에 대해서는, 생성된 규칙이 테이블 4 에 있는 이벤트 #1 과 같다. 특별한 속성이 없는 종단 노드에 대해서는, 의사결정나무를 세워갈 때 시험 노드로서 선택되지 않기 때문에 " don't care" 로 설정된다. 따라서 그것은 동등한 확률로 된다. 예를 들어, 그림 1 에서 가장 오른쪽 종단 노드를 보면 속성 1 에 대한 속성은 없다. 그래서, 5.1, 4.9, 그리고 4.7 모두 테이블 4 에 있는 이 이벤트에 대해서 동등한 확률 1/3 로 할당된다.

새로운 데이터는 테이블 4 있는 규칙에 쉽게 추가되는데, 이는 그것들이 동일한 데이터 표현을 갖기 때문이다. 테이블 5 에서, 이벤트 7 에서 이벤트 9 까지는 새로 들어온 데이터들이고, 그것들은 테이블 4 에 있는 규칙들에 결합된다. 그리고 나서 테이블 5 는 새로운 의사결정나무를 생성하기 위해 사용될 수 있다. 이런 종류의 의사결정나무의 값을 평가하기 위해서 실험들이 또한 4 장에 설명되었다.

<표 4> 그림 1 로 부터 생성된 규칙 표

E#	W	속성 1			속성 2			속성 3		클래스	
		5.1	4.9	4.7	3.5	3.0	3.2	1.4	1.3	1	2
1	1	1	0	0	1	0	0	1	0	0	1
2	1	0	0	1	1	0	0	1	0	1	0
3	1	1/3	1/3	1/3	0	1	0	1	0	1	0
4	2	1/3	1/3	1/3	1	0	0	0	1	0	1
5	1	1/3	1/3	1/3	0	1	0	0	1	1	0
6	1	1/3	1/3	1/3	0	0	1	0	1	1	0

<표 5> 새로 들어온 데이터가 추가된 규칙

E#	W	속성 1			속성 2			속성 3		클래스	
		5.1	4.9	4.7	3.5	3.0	3.2	1.4	1.3	1	2
1	1	1	0	0	1	0	0	1	0	0	1
2	1	0	0	1	1	0	0	1	0	1	0
3	1	1/3	1/3	1/3	0	1	0	1	0	1	0
4	2	1/3	1/3	1/3	1	0	0	0	1	0	1
5	1	1/3	1/3	1/3	0	1	0	0	1	1	0
6	1	1/3	1/3	1/3	0	0	1	0	1	1	0
7	10	0	1	0	1	0	0	1	0	1	0
8	1	1	0	0	0	1	0	1	0	0	1
9	1	0	1	0	0	0	1	0	1	1	0

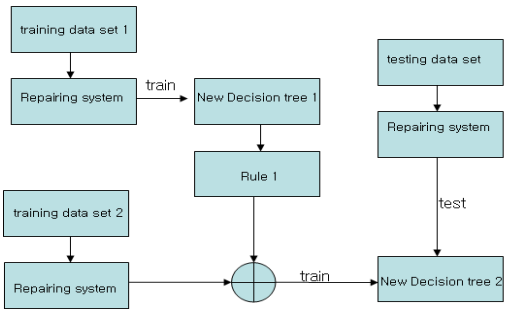
4. 실험결과

제시된 방법의 성능을 평가하고 비교하기 위해서 UCI Machine Repository 의 수집된 데이터 세트가 실험에 사용되었다. 그 데이터 세트들은 임의로 주어진 비율의 데이터를 지움으로써 불완전한 데이터 세트들이 되도록 변환된다. 불완전 데이터의 생성은 두 가지 제약이 있다: 1.각 이벤트는 적어도 하나의 속성은 남겨두어야 한다. 2.각 속성은 적어도 하나의 값은 남겨두어야 한다.

실험 결과는 10 번 돌려서 각 데이터 세트의 10 단계 상호 검증의 평균 에러 비율로 평가된다. 10 단계 상호 검증은 데이터 세트를 10 개 블록으로 나누는 과정이다. 9 개 블록들은 훈련 데이터로 통합되고, 나머지 블록은 시험 데이터를 위한 것이다.

실험은 두 부분으로 나뉜다:

첫번째 부분에서 불완전 데이터는 단지 훈련 데이터 세트에 사용되고, 시험 데이터는 완전하다. 테이블 6 은 이 부분의 결과이다. 이 부분에서 "규칙+데이터"는 이렇게 설계된다: 훈련 데이터 세트를 위해 9 개 블록을 통합한 후, 이 훈련 데이터 세트는 임의로 두 개 블록으로 나뉜다. 블록 1 은 규칙을 얻기 위해서 새로운 나무를 생성하는데 사용되고, 이 규칙은 다른 블록에 추가된다. 그리고 이 "규칙+데이터"는 분류를 위한 새로운 나무를 다시 생성하는데 사용된다. 그 결과로부터 그것은 첫 블록의 데이터가 삭제되더라도 첫 블록의 새로운 나무로부터 그 규칙은 새로운 추가 데이터와 결합되어 좋은 결과를 얻을 수 있음을 보여준다. 이 과정을 그림 2 에서 보여준다.



(그림 2) 불완전 데이터 분류 과정

두번째 부분에서 불완전 데이터는 훈련 데이터 세트와 시험 데이터 세트 양쪽 모두에 사용된다. 먼저 10% 불완전 데이터를 가진 훈련 데이터는 의사결정나무를 생성하는데 사용되고, 특정한 불완전 비율을 가진 시험 데이터는 에러 비율을 얻는데 사용되고, 테이블 7 이 이 부분의 결과이다.

두 부분에서 새로운 나무와 "규칙+데이터"가 C4.5 보다 낫다. "규칙+데이터" 경우들에서 원래 데이터의 반이 삭제되었고, 다른 원래 데이터의 반에 의해서 추가로 생성된 규칙은 새로운 의사결정나무 생성에 사용된다. 그리고 그 성능은 불완전한 데이터가 있을 때 여전히 C4.5 보다 낫다. 이것은 제안된 방법이 유용한 규칙을 생성하는데 매우 뛰어난 능력이 있음을 의미한다. 그리고 의사결정나무로부터 생성된 규칙이 새로 추가된 데이터와 쉽게 결합될 수 있기 때문에 원래 데이터를 저장할 필요조차 없다.

<표 6> 완전한 시험 데이터로 Iris 데이터 분류 에러 비율

불완전 훈련 비율	C4.5	새로운 분류자	규칙+ 데이터
0	4.7%	4.7%	5.3%
5%	5.69%	4.7%	5.3%
10%	6.16%	5.3%	6.0%
20%	7.7%	5.3%	6.7%
30%	12.09%	7.3%	7.9%

<표 7> 완전한 시험 데이터로 Iris 데이터 분류 에러 비율

불완전 시험 비율	C4.5	새로운 분류자	규칙+ 데이터
0	4.7%	4.7%	5.3%
5%	10.74%	6.0%	9.3%
10%	13.46%	8.6%	12%
20%	19.7%	14.6%	15.9%
30%	29.4%	19.3%	21.6%

5. 결론

이 논문에서 의사결정나무를 생성하는 새로운 방법이 불완전 데이터를 가진 분류 문제를 다루기 위해 제시되었다. 실험은 제시된 방법의 좋은 성능을 보여준다. 이 새로운 분류자는 또한 다른 문제를 다룰 수 있다: 규칙 개선 문제. 규칙 개선 문제를 다룰 때 새로운 의사결정나무로부터 생성된 규칙은 새로 들어오는 데이터에 의해 쉽게 추가될 수 있고, 더 많은 정보를 가진 다른 새로운 결정을 생성하기 위해서도 추가될 수 있다. 규칙과 새로운 데이터에 의해 생성된 새로운 의사결정나무는 불완전한 데이터가 존재할 때 C4.5 보다 더 좋은 결과를 가진다는 것을 보여준다. 이러한 장점들로 인해서 이 새로운 의사결정나무 방법이 강력하게 제안되었다.

참고문헌

[1] J. R. Quinlan, "C4.5:Program for Machine Learning ," San Mateo, Calif, Morgan Kaufmann, 1993

[2] Dong-Hui Kim, Dong-Hyeok Lee and Won Don Lee, "Classifier using Extended Data Expression", IEEE Mountain Workshop on Adaptive and Learning Systems, pp. 154-159, July. 2006

[3] Jun Wu, Chi-Hwa Song, Jung Min Kong, Won Don Lee, "Extended Mean Field Annealing for Clustering Incomplete Data", 2007 International Symposium on, Jeonju, KOREA, 23-24, Nov. pp 8-12.

[4] UCI data summary http://www.ics.uci.edu/~mlearn/MLSummary.html

[5] J.R.Quinlan, "Unknown attribute values in induction", in Proceedings of the Sixth International Workshop on Machine Learning,1989,pp.164-168.

[6] J.Han and M.Kamber, Data mining: Concept and Techniques, Morgan Kaufmann publishers,2001