데이터 마이닝의 분류화와 연관 규칙을 이용한 네트워크 트래픽 분석*

이창언**, 김응모**

Analysis of Network Traffic using Classification and Association Rule

Chang Un Lee, Ung Mo Kim

Abstract

As recently the network environment and application services have been more complex and diverse, there has. In this paper we introduce a scheme the extract useful information for network management by analyzing traffic data in user login file. For this purpose we use classification and association rule based on episode concept in data mining. Since login data has inherently time series characterization, convertible data mining algorithms cannot directly applied. We generate virtual transaction, classify transactions above threshold value in time window, and simulate the classification algorithm.

Key Words: 연관 규칙, 분류화, 가상 트랜잭션, 에피소드, 시뮬레이션, 네트워크 트래픽

* 본 논문은 한국과학재단이 지정한 지역협력연구센터(RRC)인 충남대학교 소프트웨어연구센터의 지원으로 수행된 과제의 결과입니다.

* This Research was supported by SOREC(Software Research Center) of KOSEF in Chungnam National University. SOREC is a Regional Research Center Designated by Korea Science and Engineering Foundation(KOSEF).

** 성균관대학교 정보통신공학부

1. 서론

데이터 마이닝은 다양한 유형을 가진 대량의 데이터로부터 데이터 상호간의 관련성, 데이터에 함축적으로 들어 있는 지식이나 패턴 및 각 도메인에서 관심을 가지는 정보를 추출하는 일련의 과정이며 현재 많은 분야에서 실용화되고 있다. 데이터 마이닝 기법은 연관 규칙(association rule), 순차 패턴(sequential pattern), 분류화(classification), 클러스터링(clustering), 기계 학습 등으로 분류된다.

본 논문에서는 데이터 마이닝 기법을 네트워크 관리에 적용 할 수 있는 방안을 제안한다. 사용되는 데이터는 네트워크에 접속한 사용자 로그인 파일이며, 이를 이용하여, 데이터 마이닝의 연관 규칙과 분류화에 적용하여 네트워크 관리에 유용한 정보를 추출한다. 이러한 정보를 이용하면 사용자의 로그인 패턴을 분석하여, 발견된 로그인 패턴과 상이한 접근 시도가 있을 시에 이를 감시하여 해킹 시도 여부에 대하여 조사 가능하며, 유사한 로그인 패턴 주기를 가지는 사용자들을 그룹으로 관리하여, 각 그룹별 사용자 관리를 가능하게 할 수 있다. 또한 어떤 시간대에, 어떤 네트워크에서 많은 접근이 발생하였는지를 분류해 낼 수 있다면 네트워크 관리에서 많은 효율성을 증가할 수 있다고 본다.

본 논문에서는 데이터 마이닝의 연관 규칙과 분류화 기법을 적용하여 시스템 사용자 로그인 데이터 분석을 한다. 여기서 시스템 사용자 로그인 데이터의 특성은 트랜잭션의 형태가 아닌 시계열 이벤트 데이터의 특징을 가지므로 기존의 데이터 마이닝 알고리즘에 바로 적용하기 어려운 한계점이 있다. 위와 같은 시계열 이벤트 특성을 분석하기 위한 에피소드 기법이 존재 하지만 이는 기존의 알고리즘을 적용할 수 없는 문제점이 있다. 따라서 본 논문에서는 시스템 사용자 로그인 데이터에 대하여 일차적으로 가상 트랜잭션을 생성한 이후 데이터 마이닝의 연관 규칙과 분류화 기법을 적용한다.

본 논문은 모두 6장으로 구성되어 있다. 1장 서론에서 데이터 마이닝에 대한 간략한 소개와 기본 접근 방법을 소개한다. 2장 관련 연구에서는 시계열 데이터 분석에 관한 연구[3][4][5][12]와 트랜잭션 데이터를 사용하는 알고리즘에 대한 연구들을 살펴본다. 이어서 3장 가상 트랜잭션에서는 본 논문에서 사용된 가상 트랜잭션의 개념에 대하여 설명한다. 4장에서는 본 논문의 전개 방향 및 각 데이터 마이닝 기법들의 적용 방법을 설명한다. 5장에서는 실험환경, 실험 데이터 및 실험 결과 분석을 한다. 마지막으로 6장에서는 결론을 설명한다.

2. 관련 연구

시간 기반의 데이터에 대한 많은 연구가 진행되었고 현재도 계속되고 있다. 이런 연구들은 시계열 데이터에 대한 연구와 시간 기반의 트랜잭션 데이터에 대한 연구로 진행되고 있다. 시계열 데이터는 발생시간과 발생이벤트 및 기타 정보를 한 개의 레코드로 구성되며 그중 시간과 이벤트가 관심 필드인 반면에, 트랜잭션 데이터는 트랜잭션 발생시간과 항목들의 집합으로 구성되며 시간과 항목 집합이 관심 필드이다. 시계열 데이터에 대한 대표적인 연구로서는 전화통신 회사에서 발생하는 시간적인 순서를 가지는 네트워크 이벤트 로그로부터 각 이벤트들 사이의 연관성을 찾는 빈발 에피소드 기법이 있고[3] 트랜잭션 데이터의 관련 연구에는 항복 집합으로 구성된 데이터 트랜잭션들로부터 각 항목간의 연관성을 반영하는 규칙을 찾는 연관 규칙을 찾는 기법이 있다[1][2][8][12].

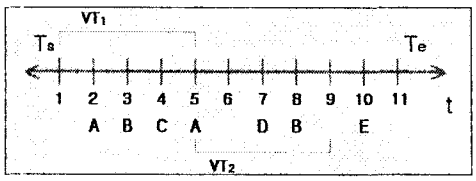
3. 기본 개념

시계열 데이터를 트랜잭션화하여 트랜잭션을 데이터 소스로 사용하는 많은 알고리즘에 적용시킬 수 있다. 여기서는 가상 트랜잭션 개념을 소

개하고 가상 트랜잭션을 생성하는 타임 윈도우 기법 알고리즘을 설명한다.

3.1 가상 트랜잭션

가상 트랜잭션(virtual transaction)은 시계열 데이터를 타임 윈도우 기법과 이벤트 윈도우 기법을 사용하여 만든 이벤트의 집합이다. 시계열 데이터에는 시간 칼럼과 이벤트 칼럼이 존재한다. 이러한 시계열 데이터를 시간 축으로 나열한 후에 정해진 윈도우의 길이만큼 이벤트들을 집합으로 묶는다. 시계열 데이터로부터 가상 트랜잭션 집합을 생성해 내는데 있어서 두 가지의 접근 방법을 시도한다. 한 가지는 고정타임 윈도우를 이용하는 방법이고 다른 한 가지는 고정이벤트 윈도우를 이용하는 방법이다. 위의 두 가지 기법에 '겹침 깊이(overlapping depth)' 를 사용하여 가상 트랜잭션을 생성한다. 그리고 가상 트랜잭션 데이터의 각각의 트랜잭션을 새로운 이벤트로 설정하고 가상 트랜잭션들의 평균 시간을 계산하여 가상 트랜잭션 데이터 집합을 생성한 후 주기 알고리즘과 순차 패턴 알고리즘의 입력 데이터로 사용한다면 주기와 순차 패턴을 분석할 수 있다. 아래 <그림 1>은 시간 축에서의 가상 트랜잭션을 설명한 그림이다.

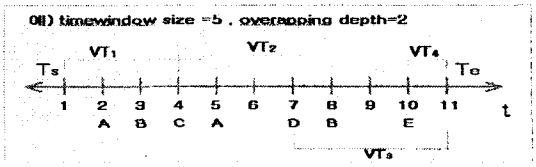


<그림 1> 시간 축에서의 가상 트랜잭션

이벤트의 Ts(start time of event sets)부터 Te(end time of event sets)사이에 발생한 이벤트 유형의 집합을 E라고 표현하고 시간 t에 발생한 이벤트를 (E,t)라고 표현하며 단 시간 t는 Ts보다 크거나 같고 Te보다는 작다. 그리고 Ts부터 Te 사이에 발생한 이벤트들 집합을 VT(Ei, Ts,Te)={(E1,t1), (E2,t2)...(En,tn)}로 표현한다. 위의 그림[3-1]에서 가상 트랜잭션VT1은 Ts가 1이고 Te가 5인 사이에서 발생한 이벤트들의 집합 말하며 {(A,2), (B,3), (C,4), (A,5)}로 나타나고 VT1의 타임 길이는 Te에서 Ts를 뺀 시간이고 VT1의 이벤트 개수는 Te와 Ts사이에 발생한 이벤트의 개수이다.

3.2 타임 윈도우

타임 윈도우 기법은 빈발 에피소드[3]을 생성하는 알고리즘에서 아이디어를 가져왔다. 시계열 데이터에서 관련 있는 이벤트는 비슷한 시간에 많이 발생한다. 이러한 속성을 이용하여 이벤트 시간 축에서 가상 트랜잭션을 만드는데 타임 윈도우 길이와 겹침 깊이를 사용한다. 얼마만큼의 시간 간격으로 가상 트랜잭션을 유지할 것인가를 결정하는 요인이 타임 윈도우 길이이다. 그리고 겹침 깊이는 인접한 가상 트랜잭션간 얼마만큼의 시간이 겹치게 가상 트랜잭션을 만들 것인가에 대한 요인이다.



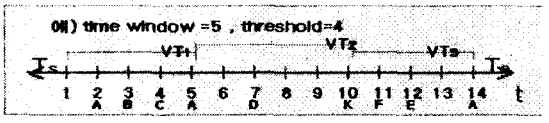
<그림 2> 타임 윈도우

<그림 2>는 타임 윈도우 길이가 5이고 겹침 깊이가 2인 경우의 생성되는 가상 트랜잭션을 표현한 것이다. 가상 트랜잭션 VT1은 VT(Ei,1,5) = { (A,2), (B,3), (C,4), (A,5) }이며 가상 트랜잭션 VT2는 VT(Ei,4,8) = { (C,4), (A,5), (D,7), (B,8) }로 가상 트랜잭션으로 생성되지만 VT4는 { (E,10) }한 개로만 구성되므로 가상 트랜잭션으로 생성하지 않는다.

4. 본론

4.1. 변경된 타임 윈도우 기법과 제한 값을 이용한 가상 트랜잭션 생성

일반적으로 타임 윈도우 기법은 시계열 데이터 전체를 가상 트랜잭션으로 생성하기 때문에 빈발하게 발생하지 않은 이벤트들을 포함하는 가상 트랜잭션을 생성한다. 이러한 가상 트랜잭션을 제거하기 위하여 기존의 타임 윈도우 기법에서 겹침 깊이를 생략하고 생성된 트랜잭션에 포함된 이벤트의 수에 제한을 두어 가상 트랜잭션을 생성한다. 여기서 사용된 제한 값은 타임 윈도우 길이에서 얼마나 빈발하게 이벤트가 발생하였는지를 알 수 있으며 결국은 각각의 사용자들의 빈발하게 접속한 시간대와 주기를 알 수 있다.



<그림 3> 제한 값과 타임 윈도우

<그림 3>은 타임 윈도우 길이가 5이고 제한 값이 4인 경우의 생성되는 트랜잭션을 표현한 것이다. 가상 트랜잭션 VT1은 VT(Ei,1,5)={ (A,2), (B,3), (C,4), (A,5) }이며 발생한 이벤트 개수가 제한 값을 만족하므로 가상 트랜잭션을 생성한다. 그러나 가상 트랜잭션 VT2는 VT(Ei,5,10)={ (D,7), (K,10) }이며 발생한 이벤트 개수가 제한 값을 만족하지 않아서 VT2는 생성되지 않으며 가상 트랜잭션 VT3 역시 생성되지 않는다. 아래 <그림 4>는 가상 트랜잭션을 생성하는 알고리즘이다.

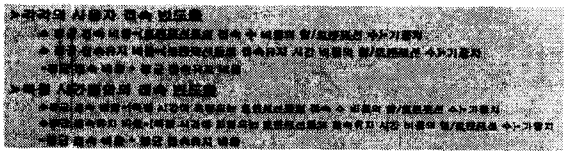
```
FUNCTION : TranGenerator
INPUT : time series source file, windowsize, threshold
OUTPUT : VTS(virtual transaction sets)

//타임 윈도우 안에 포함되는 이벤트들을 찾는다.
while(start.compareTo(preprocess.endDate)<=0)
{  end=getDate(start,timewindowsize);
   chk++;
for(j=m;j<tmpuser.getlogTimeList().getListCount();j++){

if(tmpuser.getlogTimeList().getlogtimeAt(j).getLogTime().compareTo(end
)<=0){
    tmplist.add(tmpuser.getlogTimeList().getlogtimeAt(j).getLogTime());
  waittime+=tmpuser.getlogTimeList().getlogtimeAt(j).getLogwaitTime();
 }
 else
    break;
    m++;
 }
//제한 값 적용
if(tmplist.size()>=threshold){
    TranGubun++;
    logcount=tmplist.size();
    Date tmpdate;
    totalTrnasacitonCount++;
    for(k=0;k<tmplist.size()-1;k++){
        tmpdate=(Date)tmplist.get(k);

// 가상 트랜잭션 리스트에 이벤트들을 추가한다.
timeList.getTranDataSet(tmpuser.getUser()).setAddTimeList(tmpdate,Tra
nGubun,logcount,waittime);
 }
 tmpdate=(Date)tmplist.get(k);
timeList.getTranDataSet(tmpuser.getUser()).setAddTimeList(tmpdate,true
,TranGubun,logcount,waittime);
 k=0;
 start=end;
 tmplist.removeAllElements();
 if(chk==1){
timeList.getTranDataSet(tmpuser.getUser()).setCheck(true);
 }
 if(continueCount){
     count++;
 }
 continueCount=true;
 }
 else{
        TranGubun++;
        start=end;
        tmplist.removeAllElements();
        continueCount=false;
    }
          :
```

<그림 4> 가상 트랜잭션 생성 알고리즘

4.2 가중치 계산 알고리즘

시계열 데이터인 로그인 데이터를 각각의 사용자들에 대해 가상 트랜잭션 생성 알고리즘을 이

용하여 트랜잭션들을 생성하였다. 생성된 트랜잭션에 대하여 사용 시간, 접속 수에 따라 가중치를 주어 각 사용자마다 일정한 시스템 접속 빈도율과 특정 시간의 접속 빈도율을 계산한다. <그림 5>는 사용자 접속 빈도율과 특정 시간 동안의 접속 빈도율 계산 방법을 보여준다.



<그림 5> 접속 빈도율 계산

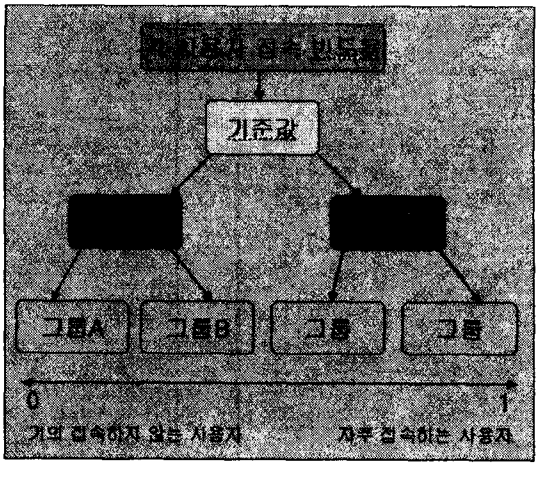
여기서 사용되는 가중치는 어떤 특정한 데이터에 중점을 두기 위해 사용된다. 즉 각 시스템의 특성이 따라 특정 데이터에 가중치의 값을 다르게 함으로서 더욱더 분석에 정확도를 높일 수 있으며, 분석된 결과의 신뢰도는 높아 질 수 있다. 아래 <그림 6>은 가중치를 구하는 알고리즘이다.

```
FUNCTION : setTranAvg
INPUT : 사용자의 가상 트랜잭션 리스트, 가중치
OUTPUT : 사용자의 평균 접속 빈도율

//각각의 사용자의 트랜잭션 리스트를 가져온다.
for(int i=0;i<tranGen.timeList.getListCount();i++){

        tds=tranGen.timeList.geTranDataSet(i);
//평균 접속 비율과 평균 접속유지 비율을 구한다.
if(tds.getTranCount()!=0){
tmplogin=(tds.getloginRate()/100/preprocess.total_login/tds.getTranCount(
)*loginW);
tmpwait=(tds.getloginwaitRate()/100/preprocess.total_wait/tds.getTranCou
nt()*loginwaitW);

  tmpTranAvg=tmplogin+tmpwait;
  } else {
      tmpTranAvg=0;    }
  TotalAVG+=tmpTranAvg;
//각 사용자의 평균 접속 빈도율을 트랜잭션 리스트에 추가한다.
  tds.setTranrate(tmpTranAvg);
```

<그림 6> 가중치 계산 알고리즘

4.3 분류화 알고리즘

계산된 접속 빈도율 대하여 데이터 마이닝의 분류화 기법을 적용하여 각 결과 값이 유사한 사용자들을 하나의 그룹으로 분류한다. <그림 7>에서는 각 분류화의 분류 임계값을 접속 빈도율의 평균값으로 설정하여 분류를 하였다. 그러나 만약 분류된 그룹들의 상이함이 별로 없을 경우에는 위의 임계값을 변경할 수 있다. 간단한 예로 <그림 7>에서 분류된 그룹 A, B의 차이가 미비하다면 최상에 루트에서 분류를 한번 적용한 결과를 최종 분석 값으로 사용할 수도 있다. 또한 그룹 A와 B를 분류함에 있어서 임계값이 분류된 그룹의 평균값으로 제시되어 있기는 하지만, 좀 더 나은 분류의 결과를 얻기 위해서 사용자가 임의로 시스템의 특성에 가장 적합한 임계값을 이용할 수 있다. 앞에서 보는 바와 같이 사용자는 유동적인 임계값을 사용함으로서 각 시스템에 적합한 형태의 분석 결과를 추출 할 수 있다.



<그림 7> 분류화 방법

그리고 아래 <그림 8>은 각각의 사용자들을 그룹화는 알고리즘이다.

```
FUNCTION : setGroup
INPUT : 평균 접속 빈도율, 사용자 평균 접속 빈도율
OUTPUT : 사용자 그룹화

평균 접속 빈도율과 사용자 평균 접속 빈도율을 비교
if(clsvalue <= tmpTranAvg){
      if((clsvalue+clsvalue/2) <= tmpTranAvg){
            tds.setGroup("그룹4");
      }else{
            tds.setGroup("그룹3");    }
}
else{
      if((clsvalue/2) <= tmpTranAvg){
          tds.setGroup("그룹2");
      }
      else{
            tds.setGroup("그룹1");
      }
  }
```

<그림 8> 분류화 알고리즘

5. 실험 및 결과

5.1 실험 환경 및 방법

본 논문에서는 본 학부 서버에서 발생된 사용자 로그인 기록을 샘플데이터로 이용하였다. 시계열 데이터의 특징을 가지는 사용자 로그인 데이터를 이용하여 가상 트랜잭션 생성기에 입력값으로 하여 가상 트랜잭션을 만들었다. 가상 트랜잭션 생성 시 타임 윈도우의 값은 2시간, 3시간, 4시간으로 변경하면서 제한 값은 3으로 고정하여 각각에 대해 가상 트랜잭션을 생성하였으며 또한 타임 윈도우 값을 3시간으로 고정시킨 다음 제한 값을 2와 4를 주었을 때 가상 트랜잭션을 생성하였다. 생성된 트랜잭션에 각각의 데이터 필드에 대하여 가중치를 두어 각 사용자마다 시스템 사용에 대한 일정한 비중을 가지는 빈도율을 계산해 냈다. 마지막으로 각 사용자의 빈도율을 이용하여 분류화 기법을 사용하여 유사한 빈도율을 가지는 사용자끼리 그룹을 생성하였다. 이렇게 분류된 그룹들에 대해 연관 규칙 알고리즘을 적용시켰다. 마지막으로 실험 결과 분석에서는 각 그룹에 속한 사용자들이 어떤 연관 규칙을 가지고 있는지 또한 각 그룹에 속한 사용자들이 어떤 시간대에 가장 많이 접속하는 지를 확인함으로써 본 논문에서 구상한 아이디어가 얼마나 적용 가능한지에 대하여 판단했다. 실험환경은 인텔 펜티엄4 1.7GHz, 512M 메모리, OS 윈도우 2000이며, 사용 프로그램 언어는 Java 이다.

본 논문에서 사용한 데이터는 본 학부 서버에서 생성된 사용자 로그인 데이터를 이용하여 분석하였다. 이벤트의 총 개수는 11843개이며 6월 7일 까지 생성된 시스템에 생성된 사용자의 수는 총 484명이다. <그림 9>는 본 논문에서 사용된 샘플데이터를 보여 준다.

```
사용자ID    접속종류    접속한 ClientIP    접속시간
gonando ftp 10.41.42.58  Fri Jun  7 13:36 - 13:36  (00:00)
Janus     ftp 203.93.174.54  Fri Jun  7 13:29 - 13:29  (00:00)
```

<그림 9> 샘플 데이터

5.2 실험 결과

5.1장에서 제시한 바와 같이 실험을 한 결과 다음과 같은 결과 값을 얻을 수 있었다. 우선 <표 1>에서는 시계열 데이터의 특징을 보이는 로그인 데이터를 가상 트랜잭션을 생성한 결과이다.

<표 1> 실험1 결과

구분	트랜잭션 생성기			
	타임윈도우값	제한값	생성된 트랜잭션수	트랜잭션수가 0인 사용자
실험A	2	3	1254	135
실험B	3	3	1302	120
실험C	4	3	1281	122
실험D	3	2	2121	71
실험E	3	4	870	189

<표 2>에서는 생성된 가상 트랜잭션에 대하여 가중치를 주었을 때 평균 접속 빈도율의 값을 보여준다.

<표 2> 실험2 결과

구분	사용자 접속 빈도율		
	접속 수에 대한 가중치	접속유지 시간에 대한 가중치	평균 접속 빈도율
실험1-A	50%	50%	0.000372
실험1-A	0%	100%	0.000360
실험1-A	100%	0%	0.000383
실험1-B	50%	0%	0.000411
실험1-B	0%	100%	0.000407
실험1-B	100%	0%	0.000415
실험1-C	50%	50%	0.000871
실험1-C	0%	100%	0.000845
실험1-C	100%	0%	0.000897
실험1-D	50%	50%	0.000361
실험1-D	0%	100%	0.000348
실험1-D	100%	0%	0.000374

아래 <표 3>은 어떤 특정 시간에 대한 최고 접속 빈도율 보여준다.

<표 3> 실험3 결과

구분	특정 시간 동안의 접속 빈도율			
	접속 수에 대한 가중치	접속유지 시간에 대한 가중치	최고 접속 빈도율	시간
실험1-A	50%	50%	0.022	Jun 4 13:00 - 15:00
실험1-B	50%	50%	0.0226	Jun 4 13:00 - 15:00
실험1-C	50%	50%	0.248	Jun 4 13:00 - 15:00

<표 4>는 가중치 값을 50%,50%로 주었을 때 평균 접속 빈도율을 기준으로 해서 비슷한 사용자들끼리 그룹화 하였으며 각 그룹들이 접속한 네트워크에 대하여 연관 규칙을 보여준다.

<표 4> 실험4 결과

구분	사용자 그룹별 분류				연관규칙 생성기				
	그룹 A	그룹 B	그룹 C	그룹 D	지지도	그룹별 항목집합개수			
						그룹 A	그룹 B	그룹 C	그룹 D
실험 2-A	202	145	73	64	3%	1	1	2	3
실험 2-B	206	138	76	64	3%	1	1	2	3
실험 2-C	343	106	21	14	3%	1	1	1	2
실험 2-D	201	147	81	55	3%	2	2	2	4
실험 2-E	218	102	74	90	3%	1	1	2	5

5.3 결과 분석

5.2 실험 결과에서 보듯 본 시스템에 약 한달 동안 가상 트랜잭션이 생성되지 않은 사용자가 171~189명에 이르는 것을 볼 때 시스템의 사용 빈도가 굉장히 낮은 시스템으로 판단된다. 또한 평균 접속 빈도율 놓고 볼 때도 마찬가지로 사용자의 접속 빈도 면에서는 시스템의 사용 빈도가 굉장히 낮음을 볼 수 있다. 그룹 D가 가장 많이 사용한 사용자로서 그 비율은 전체 사용자의 2~18%이며, 그룹 A와 그룹 B를 합친 시스템을 많이 사용하지 않는 사용자가 66~92%가 되는 것으로 짐작된다. 또한 특정 시간 동안의 접속 빈도율을 보면 타임 윈도우 크기를 늘려도 가장 접속 빈도율이 높은 시간대가 같음을 볼 수 있다.

그러나 그룹에 속한 사용자에 대하여 분석한 결과 같은 그룹 내에서도 많은 편차가 존재함을 알 수 있었다. 위와 같은 결과는 첫째로, 실험 샘플 데이터의 생성기간이 한 달이 채 못되기 때문에 각 사용자에 대하여 정확한 분석이 이루어지지 않은 결과 일 수도 있다. 둘째로, 가중치를 적용함에 있어서, 현재의 시스템의 특성을 고려하지 않았기 때문이라 생각한다.

6. 결론

본 논문에서는 데이터 마이닝의 연관 규칙과 분류화 기법을 이용하여 시스템 사용자 로그인 데이터를 분석하였다. 시계열 이벤트의 특징을 보이는 로그인 로그 데이터를 가공하여 가상 트랜잭션을 생성함으로 해서 기존의 트랜잭션 데이터를 이용하는 모든 알고리즘에 적용 가능하기 때문에 다른 분석 방법을 통한 결과 값의 추출 또한 가능하다.

본 실험에서는 시스템의 시계열 이벤트의 특징을 보이는 로그인 데이터에 대하여 일차적으로 가상 트랜잭션을 생성하였다. 그 결과에 대하여 타임 윈도우 값을 주어 임계값을 넘어서는 트랜잭션을 분류하였으며, 생성된 트랜잭션에 대하여 접속 수, 접속 유지시간에 가중치를 주어 각 사용자마다 일정한 시스템 접속 빈도율을 계산하였다. 계산된 값에 대하여 데이터 마이닝의 분류화 기법을 적용하여 각 결과 값이 유사한 사용자를 하나의 그룹으로 정의하였으며 또한 그 그룹들에 대하여 연관 규칙 알고리즘을 사용하여 접속한 네트워크간에 연관된 항목 집합들을 찾아보았다. 마찬가지로 가장 접속이 빈발한 시간대에 포함되는 사용자들을 그룹화하고 접속한 네트워크간에 관련 항목 집합들을 찾을 수가 있었다.

그러나 본 논문에서는 위에서 설명한 가중치에 대한 연구가 깊이 이루어지지 않았기 때문에 각 네트워크와 시스템의 특성이 고려되지 않은 일반적인 경우의 결과 값이다. 따라서 추후 좀더 특징 있는 데이터 값을 추가하고 이것에 대해 가중치를 어떻게 주는가 함으로 해서 좀 더 각 도메인에 적합한 결과를 보여 줄 수 있을 것이다. 또한 본 논문에서는 데이터 마이닝의 연관 규칙과 분류화 기법만을 적용하였으나 다른 기법의 적용도 고려 될 수 있다고 본다.

참고문헌

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. "Mining association rules between sets of items in large databases." In Proc. of the ACM SIGMOD Conference on Management of Data, pages 207--216(May 1993).

[2] Rakesh Agrawal and Ramakrishnan Srikant. "Fast Algorithms for Mining Association Rules." In Proc. of the 20th Int'l Conference on Very Large Databases,(September 1994)

[3] H. Mannila, H. Toivonen, and A. I. Verkamo. "Discovery of frequent episodes in event sequences." Data Mining and Knowledge Discovery, 1(3)(1997).

[4] Jiong Yang, Wei Wang, and Philip Yu, " Mining asynchronous periodic patterns in time series data" Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 275-279(2000).

[5] C. Bettini, X. Sean Wang, and S. Jajodia "Mining temporal relationships with multiple granularities in time sequences." Data Engineering Bulletin, 21:32--38,(1998)

[6] J. Han, G. Dong, and Y. Yin. "Efficient mining of partial periodic patterns in time series database." In Proc. 1999 Int. Conf. Data Engineering(ICDE'99), Sydney, Australia, (April 1999).

[7] J. Han, W. Gong, and Y. Yin. "Mining segment-wise periodic patterns in time-related databases." KDD'98(August).

[8] Rakesh Agrawal and Ramakrishnan Srikant. "Mining Sequential Patterns." In Proc. of the 11th Int'l Conference on Data Engineering(1995).

[9] Srikant, R., & Agrawal, R. "Mining sequential patterns: Generalizations and performance improvements," . EDBT(1996).

[10] F. Masseglia, F. Cathala, and P. Poncelet. "The PSP Approach for Mining Sequential Patterns." PKDD'98, LNAI, Vol. 1510, pages 176-184(1998).

[11] R. J. Bayardo. "Efficiently mining long patterns from databases," In Proc. ACM-SIGMOD Int. Conf. "Management of Data," pages 85--93, Seattle(June. 1998).

[12] Sheng Ma, Joseph L.Hellerstein. "Mining partially periodic event patterns," IEEE(2001)

[13] Myra Spiliopoulou. "Managing interesting rules in sequence mining." PKDD'99, number 1704 in LNAI, pages 554-560, (Sept. 1999).

[14] B. Lent, A. Swami, and J. Widom. "Clustering association rules." In Proc. 1997 Int. Conf. Data Engineering (ICDE'97), pages 220--231, Birmingham, England, April 1997.

[15] S. Ramaswamy, S. Mahajan, and A. Silberschatz. "On the discovery of interesting patterns in association rules." VLDB, pages 368--379(1998).

[16] Ramakrishnan Srikant and Rakesh Agrawal. "Mining Generalized Association Rules." VLDB(1995).

[17] S. Sarawagi, S. Thomas, and R. Agrawal. "Integrating association rule mining with relational database systems" ACM-SIGMOD, pp 343--354(1998).

저자소개

이창언

2001 영동대학교 컴퓨터공학과 학사

2001~현재 성균관대학교 전기전자 및 컴퓨터공학과 석사과정

관심분야: 데이터 마이닝, 데이터베이스 보안

김응모

1977.3~1981.2 성균관대학교 수학과, 학사

1983.8~1986. 5 Old Dominion University, 전산학과 석사

1986 ~1990.2 Northwestern University, 전산학과 박사

관심분야: 웹 데이터베이스(Web Database)
데이터베이스 보안(Database Security)
데이터 마이닝(Data Mining), 데이터웨어하우징
지형정보시스템(GIS)