Journal of The Korea Institute of Information Security & Cryptology(JKIISC) ISSN 1598-3986(Print)
VOL.23, NO.3, June 2013 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2013.23.3.383

데이터 스트림 빈발항목 마이닝의 프라이버시 보호를 위한 더미 데이터 삽입 기법*

정 재 열,† 김 기 성, 정 익 래‡
고려대학교 정보보호대학원

Dummy Data Insert Scheme for Privacy Preserving Frequent Itemset Mining in Data Stream*

Jay Yeol Jung,† Kee Sung Kim, Ik Rae Jeong‡
Graduate School of Information Management and Security, Korea University

요 약

데이터 스트림 마이닝 기술은 실시간으로 발생하는 데이터를 분석하여 유용한 정보를 얻는 기술이다. 데이터 스트림 마이닝 기술 중에서 빈발항목 마이닝은 전송되는 데이터들 중에서 어떤 항목이 빈발한지 찾는 기술이며, 찾은 빈발항목들은 다양한 분야에서 패턴분석이나 마케팅의 목적으로 사용된다. 기존에 제안된 데이터 스트림 빈발항목 마이닝은 악의적인 공격자가 전송되는 데이터를 스니핑할 경우 데이터 제공자의 실시간 정보가 노출되는 문제점을 가지고 있다. 이러한 문제는 전송되는 데이터에서 원본 데이터를 구별 못하게 하는 더미 데이터 삽입 기법을 통해 해결가능하다. 본 논문에서는 더미 데이터 삽입 기법을 이용한 프라이버시 보존 데이터 스트림 빈발항목 마이닝 기법을 제안한다. 또한, 제안하는 기법은 암호화 기법이나 다른 수학적 연산이 요구되지 않아 연산량 측면에서 효과적이다.

ABSTRACT

Data stream mining is a technique to obtain the useful information by analyzing the data generated in real time. In data stream mining technology, frequent itemset mining is a method to find the frequent itemset while data is transmitting, and these itemsets are used for the purpose of pattern analyze and marketing in various fields. Existing techniques of finding frequent itemset mining are having problems when a malicious attacker sniffing the data, it reveals data provider's real-time information. These problems can be solved by using a method of inserting dummy data. By using this method, a attacker cannot distinguish the original data from the transmitting data. In this paper, we propose a method for privacy preserving frequent itemset mining by using the technique of inserting dummy data. In addition, the proposed method is effective in terms of calculation because it does not require encryption technology or other mathematical operations.

Keywords: Data Stream, Privacy Preserving, Frequent Itemset mining

접수일(2013년 3월 28일), 수정일(2013년 4월 30일), 게재확정일(2013년 5월 1일)

* 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (20120007037)

† 주저자, blue7angels@korea.ac.kr
‡ 교신저자, irjeong@korea.ac.kr(Corresponding author)

I. 서 론

최근 인터넷의 발달과 스마트폰의 보급으로 인해서 우리는 장소에 구애받지 않고 많은 정보들을 접하고 있다. 접하는 정보에는 실시간 검색어, 주식 거래 현황, 교통 정보, 전력 사용량 등과 같은 실시간 정보들이 있다. 이러한 정보들을 통해서 지금 가장 이슈가 되는 뉴스와 주식 거래 흐름, 도로 교통 정보, 가정에서 사용 중인 전력 정보 등 여러 가지 정보들을 얻을 수 있다. 이러한 실시간으로 발생하는 데이터를 분석하기 위해서는 데이터 스트림 마이닝(data stream mining) 기법이 필요하다[1-9]. 데이터 스트림 마이닝 기법 중 빈발항목 마이닝(frequent itemset mining)은 실시간으로 전송되는 데이터들 사이에 어떤 데이터가 가장 많이 발생했는지 찾거나 일정 빈도 이상으로 발생한 데이터를 찾는 기법이다[2-6]. 이러한 데이터 스트림 빈발항목 마이닝을 통해서 가정에서 많이 사용하는 전자 제품의 정보와 증권 거래소에서 많이 거래되는 주식의 정보 등을 알 수 있다. 가정에서 많이 사용되는 전자 제품의 정보로는 외출 중에 집에서 사용되는 제품의 정보를 확인하여 전기를 절약하거나 외부에서의 침입 여부를 예측할 수 있다. 증권 거래소에서도 많이 거래되는 주식 정보를 통해서 최근의 주식 동향을 파악해서 제때에 매도와 매수를 통해서 손해를 줄이고 이익을 볼 수 있다. 이러한 데이터 스트림 빈발항목 마이닝은 전송되는 데이터가 노출 될 경우 데이터 제공자의 프라이버시가 위협을 받게 된다. 예를 들어 가정에서의 가전제품 사용 정보를 타인이 알게 될 경우 CCTV를 통해 감시한 것처럼 데이터 제공자가 가정에서 무엇을 하는지 알게 된다. 그리고 증권거래소에서의 거래 내역이 노출이 될 경우 A라는 사람이 B 주식을 매수해서 보유하고 있으면 B 주식이 필요한 사람들은 A 사람을 찾아와서 주식을 팔라고 강요하거나 협박하는 등 범죄에 악용될 위험이 존재한다. 그래서 이러한 경우를 해결하기 위해서 프라이버시 보존 데이터 스트림 빈발항목 마이닝 기법이 필요하다. 하지만 학계에서는 프라이버시 보존 데이터 스트림 빈발항목 마이닝 기법을 중점적으로 연구하기 보다는 데이터 스트림 빈발항목 마이닝 기법과 프라이버시 보존 데이터 마이닝 기법이 개별적으로 연구 되고 있다.

데이터 스트림 빈발항목 마이닝 기법은 마이닝 하는 기본적인 방법에 따라 샘플링(sampling), 카운팅(counting)과 해싱(hashing) 3가지로 나뉜다[7]. 샘플링 기법은 저장된 데이터들 사이에서 확률적으로 데이터들을 선택하고 선택된 데이터와 이 데이터가 빈발항목일 확률을 저장한다. 이 방법은 데이터 집합에서 빈발항목이 뽑힐 확률이 높다는 점에 기인한다. 하지만 이 방법은 빈발항목을 정확하게 뽑기 위해서는 많은 데이터 저장 공간을 필요로 한다는 단점이 있다. 카운팅 기법은 저장된 데이터에 포함된 아이템들의 개수를 세서 빈발항목을 선택하는 것으로 모든 아이템들을 저장하는 것이 아니라 일부의 아이템만을 저장해서 개수가 많은 아이템들의 카운팅은 유지하고 개수가 적은 아이템들은 다른 아이템으로 전환하여 카운팅을 하는 방법이다. 이 방법은 저장 공간을 얼마나 하느냐에 따라서 효율성과 정확성이 조절된다. 저장 공간을 많이 하면 정확성은 올라가지만 효율성이 떨어지고 저장 공간을 적게 하면 효율성은 올라가지만 정확성은 떨어지게 된다. 해싱 기법은 저장된 데이터에 포함된 아이템들을 다른 값으로 변화 시켜서 저장해 개수를 세는 방법이다. 예를 들면 10진수를 2진수로 표현해서 2진수의 각 자리만 저장을 해 개수를 세는 것이다. 1부터 31번까지의 아이템들을 2진수로 변화시켜 10110이 빈발했다면 22번 아이템이 빈발항목인 것이다. 이 방법도 변화를 시키는 방법에 따라서 저장 공간과 정확성이 조절된다.

프라이버시 보존 데이터 마이닝에서 프라이버시를 보호하기 위한 기법은 두 가지로 나눌 수 있는데 주로 사용되는 방법은 원본 데이터를 변환하거나 암호화 기법을 사용하는 방법이다[10-12]. 다른 하나는 원본 데이터 집합에 더미 데이터를 삽입하는 방법이다[13, 14]. 두 번째 방법은 많이 사용되지는 않지만 연산 효율성을 위해서 사용되는 경우가 있다. 두 방법 모두 데이터 스트림 마이닝에 적용할 수 있지만 첫 번째 방법은 연산량이 많아서 실시간 데이터 처리가 중요한 데이터 스트림 마이닝에서는 사용하기 어렵다. 두 번째 방법은 원본 데이터 집합에 더미 데이터를 삽입해서 어떤 데이터가 원본 데이터인지 알 수 없게 만들어 프라이버시를 보호하는 것으로 많은 연산량이 요구되지 않아 실시간 처리가 가능해 데이터 스트림 마이닝에 적합하다. 따라서 본 논문에서는 더미 데이터를 삽입하여 프라이버시를 보호하는 프라이버시 보존 데이터 스트림 빈발항목 마이닝 기법을 제안한다.

II. 관련 연구

데이터 스트림 빈발항목 마이닝 기법 연구는 다음과 같다. 1982년에 M.J. Fischer와 S.L. Salzberg[15]가 과반수 아이템(majority item)을 정의하였

고 그 후에 과반수 아이템 찾는 문제와 k개의 빈발항목 찾는 문제를 해결하는 알고리즘들이 제안되었다[2-6]. G. S. Manku와 R. Motwani [4]가 제안한 샘플링 기반의 스티키 샘플링(sticky sampling) 알고리즘은 샘플링 한 아이템과 그 아이템의 예측한 빈발 정도를 저장하여 조건에 맞는 아이템을 찾는 것으로 예측한 빈발 정도가 실제의 빈발 정도와 다른 경우가 발생하기 때문에 데이터 분석결과의 정확도가 떨어진다는 문제점이 있다. E. Demaine 등[5]이 제안한 카운팅 기반의 빈발(frequent) 알고리즘 역시 실제 빈발항목을 찾기 어렵다는 단점이 존재한다. A. Metwally 등[6]이 제안한 공간 절약(space saving) 알고리즘은 E. Demaine 등[5]이 제안한 빈발 알고리즘을 보완한 것으로 실제로 발생한 빈발항목 보다 적은 빈발항목이 나올 수는 있지만 잘못된 빈발항목이 나타날 확률은 적다. G. Cormode와 S. Muthukrishnan [2]가 제안한 해싱 기반의 그룹 테스팅(group testing) 알고리즘은 전송되는 데이터를 2진수로 변환하여 저장하는 것으로 본 논문에서 사용하는 기법이다. 이 알고리즘과 비슷한 C. Jin 등[3]이 제안한 h카운트(hcount) 알고리즘은 데이터의 해시와 최솟값을 이용한 것이다. 하지만 위의 알고리즘 모두 프라이버시 노출 문제를 가지고 있다.

프라이버시 보존 데이터 마이닝 기법 연구는 다음과 같다. 프라이버시 보호를 위해 주로 사용되는 기법은 원본 데이터를 변화 시켜서 프라이버시를 보존하는 기법이다[10-12]. S. Oliveira와 O. Zaïane [10]가 제안한 회전 기반 변환기법은 군집화에만 적합한 방법이고 알려진 원본 데이터로 공격이 가능하기 때문에 데이터 스트림 마이닝에 적합하지 않고 B. Goethals 등[11]이 제안한 준동형 암호화(homomorphic encryption) 기법은 모든 데이터를 각각 암호화하기 때문에 많은 연산량을 요구하여 데이터 스트림 마이닝에 적합하지 않다. M.A. Ouda 등[12]이 제안한 기법도 RSA 암호화(RSA encryption)와 준동형 암호화를 사용하기 때문에 같은 문제를 가지고 있다. 그 외에는 원본 데이터에 더미 데이터를 삽입해서 사용하는 기법이 있다[13, 14]. 더미 데이터를 삽입하는 기법은 원본 데이터를 변화하는 것보다 높은 효율성을 가지고 있다. R. Agrawal와 R. Srikant [13]가 제안한 기법은 원본 데이터에 더미 데이터를 혼합하여 혼합한 데이터를 마이닝 하는 것으로 더미 데이터를 균등분포(uniform distribution)나 정규분포(normal distribution)로 생성하여 데이터 분석의 유사성을 통해 프라이버시를 보호하면서 유사한 결과를 얻을 수 있고, P.K. Fong와 J.H. Weber-Jahnke [14]가 제안한 기법은 의사결정 나무에 더미 데이터를 삽입하여 분석하는 것으로 민감한 데이터의 프라이버시를 보호하면서 원본 데이터와 유사한 분석 결과가 나타난다.

III. 배경지식

3.1 데이터 전송 형태

데이터 전송은 원본 데이터를 전송하는 경우와 더미 데이터가 함께 전송되는 경우로 나뉜다. 원본 데이터는 일정시간동안 동일한 양의 데이터가 동일한 간격 전송되고 더미 데이터와 함께 전송되는 경우에는 전송되는 원본 데이터의 개수에 추가된 더미 데이터의 수만큼 늘어난 데이터가 전송된다. 예를 들어 1초에 원본 데이터 100개를 전송하는 데이터 스트림이 있다고 하자. 더미 데이터 50개를 삽입하면 이 데이터 스트림은 1초에 150개의 데이터를 전송하고 더미 데이터 100개를 삽입하면 1초에 200개의 데이터를 전송하는 것이다. 각 경우의 데이터가 전송되는 간격은 다르며 전송되는 데이터의 간격을 통해서 원본 데이터와 더미 데이터를 구별 할 수 는 없다.

3.2 데이터 스트림 모델(data stream models)

데이터 스트림 모델은 크게 시계열(time series), 금전 등록기(cash register), 턴스타일(turnstile) 3가지로 나눌 수 있다[1]. 시계열 모델은 입력되는 데이터를 각각의 출력데이터로 나타내는 것으로 시간대별 IP 트래픽 분석과 주식에서의 거래내역 분석 등에 사용된다. 금전 등록기 모델과 턴스타일 모델은 입력되는 데이터를 계속 저장해 가면서 출력 데이터로 나타내는데 두 모델의 차이점은 금전 등록기 모델의 입력되는 데이터는 모두 양수이고 턴스타일 모델은 양수와 음수인 것이다. 금전 등록기 모델은 IP별로 서버에 접속해서 발생하는 패킷 분석 등에 사용되고 턴스타일 모델은 지하철에서의 출입 분석이나 마트의 물건 구매 분석 등에 사용된다. 본 논문에서 입력되는 데이터가 양수와 음수이기 때문에 턴스타일 모델을 사용한다.

3.3 빈발항목 마이닝(frequent itemset mining)

빈발항목 마이닝은 데이터 집합이 있을 때 어떤 데

이터가 많은지 분석하는 것으로 빈발항목 마이닝 알고리즘에는 2 가지가 있는데 과반수 아이템을 찾는 알고리즘과 k 빈발 항목 찾기 알고리즘이 있다. 과반수 아이템 찾는 알고리즘은 데이터 전체에서 차지하는 비율이 $\frac{1}{2}$보다 큰 데이터를 찾는 알고리즘이고 k개의 빈발항목 찾기 알고리즘은 데이터 전체에서 차지하는 비율이 $\frac{1}{(k+1)}$보다 큰 데이터를 찾는 알고리즘이다.

IV. G. Cormode와 S. Muthukrishnan의 연구[2]

G. Cormode와 S. Muthukrishnan는 그룹 테스팅을 이용한 두 가지 빈발항목 마이닝 기법을 제안하였다[2]. 하나는 과반수 아이템 찾기 알고리즘이고 다른 하나는 k개의 빈발항목 찾기 알고리즘이다.

4.1 과반수 아이템 찾기 알고리즘(finding majority item algorithm)

과반수 아이템 찾기 알고리즘은 전송되는 아이템들을 2진수로 변환하여 변환한 2진수들을 이용해 과반수 아이템을 찾는 알고리즘이다. 과반수 아이템 찾기 알고리즘의 특징은 전송되는 아이템의 개수가 m개일 경우에 $(\log_2 m)+1$개의 저장 공간을 필요로 하기 때문에 적은 저장 공간이 요구되는 것이다.

x는 전송되는 아이템을 의미하며, $c[\,]$은 실제로 개수를 세는 배열이다. $trans$는 아이템의 추가와 삭제를 나타내며, $bit(x,y)$는 x를 2진수로 나타냈을 때 y번째 자리의 값을 나타내는 것으로 예를 들어 $bit(7,2)$의 경우에는 7은 2진수로 111이기 때문에 $bit(7,2)$는 111의 2번째 값인 1이다. 과반수 아이템 찾기 알고리즘은 전송되는 아이템을 $c[\,]$에 업데이트하는 알고리즘과 실제 과반수 아이템을 찾는 알고리즘으로 이루어져있고 [표 1]과 같다.

[표 1] 과반수 아이템 찾기 알고리즘

```
UpdateCounters (x, trans, c[0,...,log2 m])
    If(trans = insertion) then
        d = 1
    else
        d = -1
    c[0] = c[0] + d
    for j = 1 to log2 m
        c[j] = c[j] + (bit(x,j) × d)

FindMajority (c[0,...,log2 m])
    x = 0, t = 1
    for j = 1 to log2 m do
        if (c[j] > c[0]/2) then
            x = x + t
        t = 2 × t
    return x
```

[표 1]에서 $c[0]$는 추가한 아이템의 수에서 삭제한 아이템의 수를 뺀 것으로 실제로 존재하는 아이템의 개수를 나타낸 것이다. 그래서 실제로 과반수 아이템을 찾는 알고리즘을 보면 $c[1...\log_2 m]$의 값을 $c[0]$를 2로 나눈 값과 비교하여 크면 1, 작으면 0으로 표현한 2진수를 찾는다. 찾은 2진수를 10진수로 변환한 값이 과반수 아이템이다.

4.2 k개의 빈발항목 찾기 알고리즘(finding k hot items algorithm)

k개의 빈발항목 찾기 알고리즘은 과반수 아이템 찾기 알고리즘처럼 전송되는 아이템을 2진수로 변환하여 변환한 2진수를 이용하는 것은 같지만 과반수 아이템을 찾는 것이 아니기 때문에 저장 공간을 3차원 배열로 늘려서 더 많은 저장 공간을 이용하여 빈발항목을 찾는다. k개의 빈발항목 찾기 알고리즘의 특징은 무조건 k개의 빈발항목을 찾는 것이 아니라 전체를 $k+1$로 나눈 값보다 큰 아이템들을 찾는 것이다.

$m, x, trans$는 과반수 아이템 찾기 알고리즘과 같은 의미로 사용되고 k는 찾고자하는 빈발항목의 개수이다. $1-\delta$는 k개의 빈발항목 찾기 알고리즘을 수행했을 때 정확한 결과가 나올 확률을 의미하고 $T, W, (\log_2 m)+1$은 각 차원의 최댓값으로 $T=\log_2(\frac{k}{\delta})$, $W=2k$으로 표현한다. P는 난수 $a[i], b[i]$를 뽑을 때 사용하는 값으로 W보다 큰 소수이고 $h_i(x)$는 아이템 x를 난수를 이용해 해시한 것으로 $h_i(x)=(((a_i \times x)+b_i) \bmod P) \bmod W$로 표현한다. k개의 빈발항목 찾기 알고리즘은 $c[][][]$과 난수 $a[i], b[i]$를 초기화 해주는 알고리즘, 전송되는 아이템을 $c[][][]$ 에 업데이트하는 알고리즘과 실제 빈발항목을 찾는 알고리즘으로 이루어지고 [표 2]와 같다.

[표 2]의 ProccessItem$(x, trans, T, W)$에서 사용되는 UpdateCounters$(x, tt, c[i][h_i(x)])$은 과반수 아이템 찾기 알고리즘에서 사용하는 것과 유사한

〔표 2〕 k개의 빈발항목 찾기 알고리즘

```
Initialize (T, W)
   n=0
   for i=1 to T
      for j=0 to W-1
         for k=0 to log2m
            c[i][j][k]=0
      a[i]=Random(0, P-1)
      b[i]=Random(0, P-1)

ProccessItem (x, trans, T, W)
   If(trans== insertion) then
      n=n+1
   else
      n=n-1
   for i=1 to T
     UpdateCounters (x, trans, c[i][h_i(x)])

GroupTest (T, W, k, c[][][])
  for i=1 to T
    for j=0 to W-1
      r=1, t=n/(k+1), x=0
      if c[i][j][0]>t then
        for l=1 to log2m do
          p=c[i][j][l], q=c[i][j][0]-p
          if ((p≤t∧q≤t)∨(p>t∧q>t))
            then skip to next value of i
          else if (p>t) then
            x=x+r
          r=2×r
        output x
```

것으로 다른 점은 1차원 배열이 아니라 3차원 배열에 값을 저장하는 것이다. $c[i][h_i(x)][0]$은 아이템 x를 추가한 개수에서 삭제한 개수를 뺀 값이고, 카운터 값들은 i의 값에 따라서 다양한 곳에 저장된다. 예를 들어 T가 3이고 $h_1(x)=2$, $h_2(x)=5$, $h_3(x)=1$이면 아이템 x의 값을 $c[1][2][0,...,\log_2 m]$, $c[2][5][0,...,\log_2 m]$, $c[3][1][0,...,\log_2 m]$에 업데이트 시켜주는 것이다. 이렇게 하는 이유는 해시를 할 경우 다른 아이템들이 같은 해시 값을 가질 확률이 존재하기 때문에 각 아이템들의 개수를 분산 저장해서 정확한 빈발항목들을 찾으려하는 것이다. 그래서 T값을 크게 하면 할수록 다른 아이템들 사이에 같은 해시 값을 가지는 경우를 줄여서 정확한 결과를 알 수 있다. 빈발항목은 마지막 알고리즘에서 모든 $c[i][j][0]$값을 전체 데이터 개수를 $k+1$로 나눈 값과 비교하여 클 경우 $c[i][j][0,...,\log_2 m]$을 이용해서 찾는다. 빈발항목은 최초 0개에서 최대 k개가 나올 수 있다.

V. 더미 데이터 삽입을 통한 프라이버시 보호 기법

G. Cormode 등〔2〕이 제안한 빈발항목 마이닝 기법에서 발생하는 프라이버시 문제는 더미 데이터를 삽입하는 프라이버시 보존 기법으로 해결가능하다. 제안하는 방법은 전송되는 원본 데이터 사이에 더미 데이터들을 삽입하여 전송되는 데이터에서 원본 데이터를 구별 못하게 함으로써 프라이버시를 보호하는 것이다. 하지만 일반적인 데이터를 더미 데이터라 하여 원본 데이터에 삽입을 하면 원본 데이터 분포에 변화를 주어서 원본 데이터를 분석한 결과와 다른 결과가 나타날 수 있다. 그래서 더미 데이터를 원본 데이터의 아이템들에서 균등하게 선택한다. 특히 제안하는 기법의 데이터 스트림 모델은 턴스타일 모델이기 때문에 더미 데이터가 원본 데이터의 아이템들과 부호에서 모두 균등하게 선택되면 추가되는 더미 데이터만큼 삭제가 된다. 그래서 원본 데이터 분포에 영향을 주지 않기 때문에 정확한 분석을 할 수 있다.

그리고 삽입하는 더미 데이터 값뿐만 아니라 더미 데이터를 삽입할 위치도 중요한데 더미 데이터를 균등한 간격으로 삽입을 할 경우 악의적인 공격자에 의해서 삽입된 더미 데이터의 정보가 노출되면 차후에 삽입되는 더미 데이터의 정보도 노출된다. 그러면 악의적인 공격자는 자신이 공격한 이후에도 삽입되는 데이터가 원본 데이터인지 더미 데이터인지 알 수 있다. 그래서 원본 데이터 다음에 삽입되는 더미 데이터의 개수를 조절하여 더미 데이터의 위치 노출을 최소화할 수 있다.

본 논문에서 제안하는 원본 데이터에 더미 데이터를 삽입하는 기법은 〔표 3〕과 같다. t시간 동안 전송되는 원본 데이터의 수를 s이라고 하고, U는 더미 데이터 분포, V는 사용자가 선택한 더미 데이터 삽입 분포이다. $x[1,...,s]$는 t시간 동안 전송될 원본 데이터 배열, $trans[1,...,s]$는 t시간 동안 전송될 원본 데이터의 부호 배열이다. $x'[1,...,l]$은 t시간 동안 전송될 원본 데이터에 더미 데이터가 삽입된 배열이고 $trans'[1,...,l]$는 $x'[]$의 부호 배열이다. l, r, y, tt은 정수로 l은 원본 데이터에 더미 데이터를 삽입했을 때의 전체 데이터 수이고 r은 더미 데이터 선택 분포 V에서 뽑은 더미 데이터 삽입 개수, y는 더미 데이터 분포인 균등 분포 U에서 뽑은 더미 데이터, tt는 0과 1 중 하나로 0일 경우 삽입, 1일 경우 삭제를 나타낸다. 예를 들어 3개의 아이템 1, 2, 3이 있고 원본 데이터

〔표 3〕 더미 데이터 삽입 알고리즘

```
Input  : s, U, V, x[], trans[]
Output : x'[], trans'[]
DummyInsert (s, U, V, x[], trans[] )
  l=0
  for i=1 to s
    l=l+1
    x'[l]=x[i], trans'[l]=trans[i]
    select randomly r in V
    if (r==0) then
      skip to next value of i
    else
      for j=1 to r
        l=l+1
        select randomly y in U
        select randomly tt in 0 or 1
        x'[l]=y, trans'[l]=tt
      End for
    End if
  End for
End DummyInsert
```

와 부호 배열이 {1, -1, 3, 1, 2, 1, 2, 1, 1, -2} 라고 가정하자. 〔표 3〕의 알고리즘을 이용하여 더미 데이터를 삽입한 데이터와 부호 배열이 {1, 2, -1, 1, 3, 1, -1, 2, 3, 1, -2, 2, 1, -3, 1, -2} 일 때 삽입한 더미 데이터만큼 삭제되기 때문에 데이터의 개수는 늘어났지만 빈발항목 분석에 사용되는 데이터는 동일하다 그래서 원본 데이터를 제안한 더미 데이터 삽입 알고리즘을 통해서 변경하여 변경한 데이터를 빈발항목 마이닝 기법을 이용하여 분석하면 프라이버시를 보호하면서 빈발항목을 찾을 수 있다.

VI. 안전성 정의 및 분석

본 장에서는 원본 데이터를 실시간으로 생성하여 생성한 원본 데이터에 다양한 삽입 분포를 가지는 더미 데이터들을 삽입하여 그에 따른 분석을 하였다. 실시하는 실험은 C를 이용하여 정규분포를 따르는 원본 데이터와 다양한 삽입 분포를 따르는 더미 데이터를 생성하고 빈발항목을 분석하였다. 실험에 이용한 알고리즘은 k개의 빈발항목 찾기 알고리즘이고 사용한 변수는 〔표 4〕와 같다. 실험에 사용한 더미 데이터 삽입 분포는 정규분포와 균등분포로 하였다. 평균(μ)이 50이고 표준편차(σ)가 10인 정규분포와 표준정규분포표를 이용하여 평균이나 평균보다 작은 값이 선택될 경우에는 더미 데이터의 삽입 개수를 0(확률 50%), 50보다 크고 58.4보다는 작거나 같은 경우에는 1(확률 30%), 58.4보다 큰 경우에는 2(확률 20%)로 하여서 실험하였다. 그리고 같은 정규분포를 이용하여 52.54와 같거나 작은 경우에는 더미 데이터 삽입개수를 0(확률 60%), 52.54보다 크고 62.8보다 같거나 작은 경우에는 1(확률 30%), 62.8보다 큰 경우에는 2(확률 10%)로 한 경우와 55.25보다 같거나 작은 경우에는 더미 데이터 삽입개수를 0(확률 70%), 55.25보다 크고 62.8보다 같거나 작은 경우에는 1(확률 20%), 62.8보다 큰 경우에는 2(확률 10%)로 한 경우에서도 실험을 하였다. 즉 정규분포를 이용하여 0:1:2의 비율이 50:30:20, 60:30:10과 70:20:10인 3가지 실험을 하였다. 균등분포를 사용한 경우에는 0, 1이 균등하게 나오는 것과 0, 1, 2가 균등하게 나오는 것 2가지 실험을 하여 총 5가지 분포에 대해서 실험하였다. 실험 컴퓨터는 Pentium(R) Dual-Core 2.93Ghz, RAM 4GB이고 운영체제로는 Windows 7을 사용 하였다.

〔표 4〕 실험 변수

변 수 명	변 수 값
원본 데이터의 아이템 수(m)	128
생성한 원본 데이터의 전체 개수	10,000
빈발 항목 파라미터(k)	3
1초에 전송하는 원본 데이터 수 (s)	10
첫 번째 빈발항목 저장 공간(T)	4
두 번째 빈발항목 저장 공간(W)	6
난수 생성 시 사용하는 값(P)	19

제안하는 기법은 실험을 통해 안전성, 정확성, 효율성 등 3가지 측면에서 분석하였다.

6.1 프라이버시의 정의

제안하는 기법의 안전성은 원본 데이터에 더미 데이터를 삽입하여 전송되는 데이터를 봤을 때 어떤 데이터가 원본 데이터이고 더미 데이터인지 파악 할 수 없다는 점이다.

악의적인 공격자가 데이터가 전송되는 과정을 지켜보면서 전송되는 모든 데이터를 저장하여 균등 분포를 따르는 데이터들을 제거하면 원본 데이터와 유사한 형태의 데이터를 유추할 수 있다. 하지만 유추한 데이터

들을 통해서 실시간으로 전송되는 데이터가 원본 데이터인지 더미 데이터인지 구별하기는 어렵다. 그래서 이 기법의 프라이버시 레벨($PL(Privacy\,Level)$)은 전송되는 데이터를 통해서 더미 데이터를 얼마만큼의 확률로 구별 할 수 있는 가로 정의한다.

원본 데이터의 개수가 N개 이고 더미 데이터의 개수가 D개이면 전체 데이터의 개수는 $N+D$개 이다. 더미 데이터의 확률은 전체 데이터의 개수와 더미 데이터의 개수로 계산이 가능하다. 그래서 프라이버시 레벨은 다음과 같이 표현한다.

$$PL = \frac{D}{N+D} \quad (1)$$

프라이버시 레벨이 높을수록 원본 데이터를 알아낼 확률이 적다고 할 수 있다.

6.2 정확성 정의

제안하는 기법의 정확성은 생성한 원본데이터를 분석한 빈발항목과 더미 데이터를 삽입한 데이터를 분석한 빈발항목을 비교해서 일치한 정도로 정의한다. 빈발항목 비교 알고리즘은 〔표 5〕와 같다. $x[], x'[], trans[], trans'[]$는 5장에서의 정의와 같고 F는 $x[], trans[]$가 전송된 횟수로 tF는 데이터 전송에 걸린 전체 시간이다. $X[], X'[]$는 tF시간 동안의 $x[], x'[]$들의 집합으로 $X[] = \{x_1[], \cdots, x_F[]\}$, $X'[] = \{x'_1[], \cdots, x'_F[]\}$를 나타내고, $Trans[]$, $Trans'[]$ 또한 $X[], X'[]$와 같은 방법으로 나타낸다. FrequentItemset($x[], trans[]$)은 〔표 2〕의 ProccessItem 알고리즘과 GroupTest 알고리즘을 이용하여 $x[], trans[]$의 빈발항목 $f[]$을 구하는 알고리즘이다. d_n은 $f[], f'[]$를 비교했을 때 같았던 횟수이고 c_n은 $f[], f'[]$를 비교한 횟수이다. t시간 마다 전송되는 원본 데이터 배열과 더미 데이터 삽입 배열을 이용하여 전체시간 tF의 d_n, c_n을 구한다. 일치 정도($CL(Conformity\,Level)$)를 다음과 같이 표현한다.

$$CL = \frac{d_n}{c_n} \quad (2)$$

이 알고리즘은 t시간 마다 전송되는 데이터들에서 빈발항목을 찾아 원본 데이터에서와 더미 데이터를 삽입한 경우를 비교하여 전체 비교횟수에서 얼마나 일치하는지 찾는 알고리즘이다. 일치 정도가 높을수록 제안하는 기법에서 정의한 정확성이 높다고 할 수 있다.

〔표 5〕 빈발항목 비교 알고리즘

```
Input : X[], X'[], Trans[], Trans'[]
Output : d_n, c_n
Conformity(X[], X'[], Trans[], Trans'[])
  d_n = 0, c_n = 0
  for i = 1 to F
    f_i[] ← FrequentItemset(x_i[], trans_i[] )
    f'_i[] ← FrequentItemset(x'_i[], trans'_i[] )
    if (f_i[] == f'_i[])
      d_n = d_n + 1
    end if
    c_n = c_n + 1
  end for
end Conformity
```

6.3 효율성 정의

제안하는 기법의 효율성은 업데이트 시간과 저장 공간으로 분석하였다.

4장에서 설명한 기법에서 과반수 아이템 찾기 알고리즘의 경우에 업데이트 시간은 $O(\log_2 m)$이고 저

〔표 6〕 효율성 비교

	그룹 테스팅	제안하는 기법	비교
과반수 알고리즘 업데이트 시간 (각 데이터)	$O(\log_2 m)$	$O(\log_2 m)$	동일함
과반수 알고리즘 업데이트 시간 (전체 데이터)	$N \times O(\log_2 m)$	$(N+D) \times O(\log_2 m)$	$\frac{N+D}{N}$ 배 증가
과반수 알고리즘 저장 공간 (전체 데이터)	$O((\log_2 m)+1)$	$O((\log_2 m)+1)$	동일함
k 빈발항목 알고리즘 업데이트 시간 (각 데이터)	$O(\log_2(\frac{k}{\delta}) \times \log_2 m)$	$O(\log_2(\frac{k}{\delta}) \times \log_2 m)$	동일함
k 빈발항목 알고리즘 업데이트 시간 (전체 데이터)	$N \times O(\log_2(\frac{k}{\delta}) \times \log_2 m)$	$(N+D) \times O(\log_2(\frac{k}{\delta}) \times \log_2 m)$	$\frac{N+D}{N}$ 배 증가
k 빈발항목 알고리즘 저장 공간 (각 데이터 마다)	$O(k \times \log_2(\frac{k}{\delta}) \times \log_2 m)$	$O(k \times \log_2(\frac{k}{\delta}) \times \log_2 m)$	동일함
처리하는 데이터 수	N	$N+D$	

〔표 7〕 다양한 삽입 분포를 가지는 더미 데이터를 이용한 실험 결과

더미 데이터 삽입 분포(V)	삽입된 더미 데이터의 수	프라이버시 레벨(PL)	원본 데이터와의 일치 정도(CL)	업데이트 효율성
정규분포(0:50%, 1:30%, 2:20%)	5781	0.37	0.9795	1.58배
정규분포(0:60%, 1:30%, 2:10%)	4869	0.33	0.9826	1.47배
정규분포(0:70%, 1:20%, 2:10%)	3918	0.28	0.9881	1.39배
균등분포 (0, 1)	5116	0.34	0.9843	1.51배
균등분포 (0, 1, 2)	10109	0.50	0.9771	2.01배

장 공간은 $O((\log_2 m)+1)$이다. 그리고 k개의 빈발항목 찾기 알고리즘의 경우에는 업데이트 시간은 $O(\log_2(\frac{k}{\delta})\times\log_2 m)$이고 저장 공간은 $O(k\times\log_2(\frac{k}{\delta})\times\log_2 m)$이다.

제안하는 기법의 경우에도 과반수 아이템 찾기 알고리즘의 업데이트 시간은 $O(\log_2 m)$이고 저장 공간은 $O((\log_2 m)+1)$이다. 또한 k개의 빈발항목 찾기 알고리즘의 업데이트 시간은 $O(\log_2(\frac{k}{\delta})\times\log_2 m)$이고 저장 공간은 $O(k\times\log_2(\frac{k}{\delta})\times\log_2 m)$으로 동일하다.

원본 데이터의 개수가 N개 이고 더미 데이터의 개수가 D개일 때 그룹 테스팅 기법에서의 효율성과 제안하는 기법의 효율성을 비교하면 〔표 6〕과 같다.

〔표 6〕을 보면 저장 공간과 각 데이터의 업데이트 처리시간은 동일하지만 처리하는 데이터 수가 다르기 때문에 그룹 테스팅 기법의 과반수 알고리즘 전체 데이터 업데이트 시간은 $N\times O(\log_2 m)$이고 제안하는 기법의 전체 데이터 업데이트 시간은 $(N+D)\times O(\log_2 m)$로 차이가 있다. 예를 들어 더미 데이터의 개수가 원본 데이터의 개수와 같다면 제안하는 기법의 전체 데이터 업데이트 시간은 그룹 테스팅에서의 업데이트 시간보다 2배 더 걸리는 것이다. 그래서 삽입하는 더미 데이터에 비례하여 업데이트 시간의 효율성을 떨어진다. 하지만 요구되는 저장 공간은 같기 때문에 동일한 공간 효율성을 가진다.

6.4 실험결과분석

〔표 7〕은 원본 데이터에 다양한 삽입 분포를 가지는 더미 데이터를 삽입하여 분석한 결과로 삽입된 더미 데이터의 수와 프라이버시, 원본 데이터 분석결과의 일치 정도, 업데이트 시간의 효율성을 보여준다.

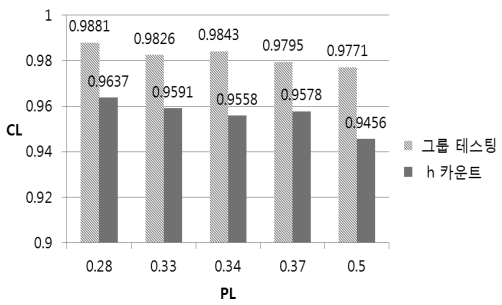
〔표 7〕의 실험 결과를 분석해 보면 삽입된 더미 데이터의 수에 따라서 원본 데이터의 일치 정도가 다소 차이가 있지만 대부분 높게 나온 것을 알 수 있다. 그리고 가장 많은 더미 데이터를 삽입한 실험은 프라이버시 레벨이 가장 높게 나타나고 업데이트 시간의 효율성이 가장 낮게 나타난다. 또한 정규분포(0:60%, 1:30%, 2:10%)와 균등분포 (0, 1)를 비교해보면 모든 부분에서 유사하다.

위 실험 결과를 통해서 제안하는 기법을 사용하는 경우에는 원본 데이터만을 분석한 결과와 유사한 결과를 도출 할 수 있다. 그리고 삽입된 더미 데이터의 수에 따라 프라이버시와 효율성이 반비례 관계로 변화하고 다른 분포를 따르지만 유사한 결과를 도출할 수 있다. 위 결과를 통해서 제안하는 기법은 사용자가 요구한 프라이버시와 효율성, 분포를 만족시키면서 사용할 수 있다.

VII. 다른 빈발항목 알고리즘에서의 기법 적용

균등분포를 따르는 더미 데이터를 삽입하여 프라이버시를 보존하는 방법은 그룹 테스팅 알고리즘 외에도 다른 빈발항목 알고리즘에서 적용이 가능하다.

빈발항목 알고리즘 중에서 샘플링 기반의 스티크 샘플링 알고리즘은 그룹 테스팅 기법과 데이터 타입도 다르고 빈발항목을 구하는 방식 또한 전체 데이터를 분석하는 것이 아니라 일부 데이터를 샘플링해서 분석하기 때문에 제안하는 기법을 사용하여 프라이버시를 보호하기 어렵다. 카운팅 기반의 공간 절약 알고리즘〔6〕은 그룹 테스팅 기법과 데이터 타입은 다르지만 전송되는 데이터들을 특정 공간을 이용하여 빈발항목을 구하는 점이 비슷하여 제안하는 기법을 적용할 수 있다. 하지만 제안하는 기법을 적용하기 위해서는 많은 제약 조건이 필요하다. 첫 번째는 많은 저장 공간이 필요하다. 그룹 테스팅의 경우에는 저장 공간이 $O((\log_2 m)+1)$ 이지만 공간 절약 알고리즘의 경우에는 그룹 테스팅과 동일한 공간으로 빈발항목을 구하게 되면 모든 데이터의 오차율이 급증하기 이 경우 빈발항목을 구할 수 없다. 예를 들어 8개의 아이템 1,



〔그림 1〕 h카운트 알고리즘과 그룹 테스팅 기법 비교

2, 3, 4, 5, 6, 7, 8이 있을 때 저장 공간이 3개이고 1이 가장 많다고 가정하면 전송되는 3개의 데이터 사이에 1이 없을 경우에 1의 오차율이 급증하여 빈발항목에서 제외가 된다. 더미 데이터가 균등하게 삽입될 경우에 원본 데이터에서는 3개의 데이터 사이에 1이 모든 경우에 있었다고 하더라고 더미 데이터가 삽입되면 1일 포함이 안되는 경우가 발생한다. 그래서 아이템 공간에 비례해서 많은 저장 공간이 필요하다, 두 번째는 많은 더미 데이터를 삽입할 수 없다. 첫 번째와 같은 이유로 많은 더미 데이터가 삽입되면 3개의 데이터 사이에 1이 빠질 확률이 증가하기 때문에 적은 수의 더미 데이터를 삽입해야 한다. 하지만 이럴 경우에는 프라이버시 노출 문제가 발생할 수 있다는 문제가 있다. 해싱 기반의 h카운트 알고리즘[3]은 그룹 테스팅 기법과 데이터 타입과 빈발항목을 구하는 방법도 비슷하다, 그래서 이 알고리즘에 균등분포를 따르는 더미 데이터를 삽입하면 추가되는 더미 데이터만큼 삭제가 되기 때문에 원본 데이터 분포에 많은 영향을 주지 않는다. 그러므로 이 알고리즘 역시 제안하는 기법을 이용하면 프라이버시를 보호하면서 빈발항목 마이닝을 할 수 있다. 〔그림 1〕을 보면 h카운트 알고리즘에 제안하는 기법을 적용하더라도 그룹 테스팅 기법처럼 높은 *CL*을 가진다는 것을 알 수 있다.

그래서 제안하는 기법은 턴스타일 모델과 해싱 기반의 알고리즘에서는 적용가능 하지만 금전 등록기 모델과 카운터 기반의 알고리즘에서는 많은 제약 조건이 따르고 샘플링 기반의 알고리즘에서는 적용하기 어렵다.

VIII. 결 론

본 논문은 G. Cormode 등[2]이 제안한 빈발항목 찾기 알고리즘에 프라이버시 보존 기법 중 더미 데이터 삽입 기법을 접목하여 프라이버시 보존 빈발항목 마이닝 알고리즘을 제안하였다. 제안한 기법을 이용하면 원본 데이터만을 분석했을 때보다 처리속도는 다소 느리지만, 프라이버시를 보존하면서 원본 데이터의 분석 결과와 유사한 결과를 도출할 수 있다. 또한, 그룹 테스팅 알고리즘 외에도 다양한 빈발항목 알고리즘에 적용이 가능한 실용적인 기법이다.

참고문헌

〔1〕 S. Muthukrishnan, Data streams: algorithms and applications, Lightning Source Inc, Jan. 2005.

〔2〕 G. Cormode and S. Muthukrishnan, "What's hot and what's not: tracking most frequent items dynamically," Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART sy- mposium on Principles of database systems, pp. 296-306, Jun. 2003.

〔3〕 C. Jin, W. Qian, C. Sha, J.X. Yu, and A. Zhou, "Dynamically maintaining frequent items over a data stream," Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, pp. 287-294, Nov. 2003.

〔4〕 G.S. Manku and R. Motwani, "Approximate frequency counts over data streams," Proceedings of the 28th International Conference on Very Large Data Bases, pp. 346-357, Aug. 2002.

〔5〕 E. Demaine, A. López-Ortiz, and J. Munro, "Frequency estimation of internet packet streams with limited spa- ce," Proceedings of the 10th Annual European Symposium, pp. 348-360, Sep. 2002.

〔6〕 A. Metwally, D. Agrawal, and A.E. Abbadi, "Efficient computation of frequent and top-k elements in data strea- ms," Proceedings of the 10th Interna- tional Conference on Database Theor- y, pp. 398-412, Jan. 2005.

〔7〕 H. Liu, Y. Lin, and J. Han, "Methods for

mining frequent items in data streams: an overview," Knowledge and Information Systems, vol. 26, no. 1, pp. 1-30, Jan. 2011.

[8] S. Pramod and O.P. Vyas, "Recent frequent itemsets mining over data streams," Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology, pp. 484-489, Oct. 2012.

[9] M. Deypira, M.H. Sadreddinib, and S. Hashemib, "Towards a variable size sliding window model for frequent itemset mining over data streams," Computers & Industrial Engineering, vol. 63, no. 1, pp. 161-172, Aug. 2012.

[10] S. Oliveira and O. Zaïane, "Achieving privacy preservation when sharing data for clustering," Proceedings of International Workshop on Secure Data Management in a Connected World, pp. 67-82, Aug. 2004.

[11] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikäinen, "On private scalar product computation for privacy-pres- erving data mining," In The 7th Ann- ual International Conference in Information Security and Cryptology, pp. 104-120, Dec. 2004.

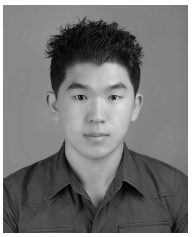
[12] M.A. Ouda, S.A. Salem, I.A. Ali, and E.M. Saad, "Privacy-preserving data mining (PPDM) method for horizon- tally partitioned data," International Journal of Computer Science Issues, vol. 9, no. 5, pp. 339-347, Sec. 2012.

[13] R. Agrawal and R. Srikant, "Privacy-preserving data mining," Proceedings of the 2000 ACM SIGMOD interna- tional conference on Management of data, pp. 439-450, Jun. 2000.

[14] P.K. Fong and J.H. Weber-Jahnke, "Privacy Preserving Decision Tree Learning Using Unrealized Data Sets," IEEE Transactions on Knowle- dge and Data Engineering, vol. 24, no. 2, pp. 353-364, Feb. 2012.

[15] M.J. Fischer and S.L. Salzberg, "Finding a majority among n votes," Research Report 252, Department of Computer Science, University of Yale, Oct. 1982.

〈저자소개〉



정 재 열 (Jay Yeol Jung) 학생회원
2010년 8월: 고려대학교 정보수학과 졸업
2010년 9월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 프라이버시향상기술(PET), 데이터베이스 보안, 비밀 공유 기법

김 기 성 (Kee Sung Kim) 학생회원
2008년 2월: 서울시립대학교 수학과 졸업
2011년 2월: 고려대학교 정보보호대학원 석사 졸업
2012년 3월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 프라이버시향상기술(PET), 데이터베이스 보안, 암호 이론

정 익 래 (Ik Rae Jeong) 정회원
1998년 2월: 고려대학교 전산학과 학사 졸업
2000년 2월: 고려대학교 전산학과 석사 졸업
2004년 8월: 고려대학교 정보보호대학원 박사 졸업
2006년 6월~2008년 2월: 한국전자통신연구원 암호기술연구팀 선임연구원
2008년 3월~2011년 8월: 고려대학교 정보경영공학전문대학원 조교수
2011년 9월~현재: 고려대학교 정보보호대학원 부교수
〈관심분야〉 프라이버시향상기술(PET), 데이터베이스 보안, 암호 이론