네트워크 패킷 데이터 마이닝을 위한 데이터 압축 전처리 기법에 관한 연구

나상혁*°, 이원석**
*연세대학교 컴퓨터과학과 석사과정, **연세대학교 컴퓨터과학과 정교수

The research of preprocessing technique of Data Compaction customized to network packet data

Na Sang-Hyuck, Lee Won-Suk
Department of Computer Science at Yonsei University
E-mail : {mailofnsh, leewo}@database.yonsei.ac.kr

요 약

네트워크(Network) 라우터(Router)와 스위치(Switch) 장치에서 수많은 패킷(Packet)이 통과된다. 네트워크에 연결된 컴퓨터가 20대일 경우에 일일 평균 패킷 전송양은 약 400GB 정도에 이른다. 이러한 패킷 데이터를 분석하기 위해서는 수집된 데이터를 디스크 장치에 저장할 수 있는 대규모의 저장공간과 주기적인 백업이 필요하다.

수집된 데이터 원형에는 사용자가 원하는 정보뿐만 아니라 불필요한 정보가 산재해있다. 따라서 수집된 데이터를 원형 그대로 저장하는 것이 아니라 원하는 정보(Information)와 지식(Knowledge)이 유지되고 쉽게 식별될 수 있도록 데이터를 가공해서 요약된 정보를 유지하는 것이 효과적이다.

전 세계적으로 네트워크를 통과하는 패킷 데이터의 양이 헤아릴 수 없을 만큼 증가하고, 인터넷 보급률이 증가함에 따라서 인터넷 사용자 및 소비자의 정보 분석의 필요성이 부각되고 있다. 본 논문에서는 네트워크에서 수집된 패킷 데이터에 적합한 데이터 전처리 기법(preprocessing)을 제안한다.

1. 서론

네트워크(Network) 라우터(Router)와 스위치(Switch) 장치에서 수많은 패킷(Packet)이 통과된다. 네트워크에 연결된 컴퓨터가 20대일 경우에 일일 평균 패킷 전송 량은 약 400GB 정도에 이른다. 이러한 패킷 데이터를 분석하기 위해서는 수집된 데이터를 디스크 장치에 저장할 수 있는 대규모의 저장공간과 주기적인 백업이 필요하다. 그러나 모든 데이터에 원하는 정보가 존재하지 않는다. 따라서 패턴(Pattern)을 얻을 수 있는 데이터만 추출하여 요약된 형태로 유지할 수 있는 전처리 기술(Preprocessing)이 요구된다.

이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국과학재단의 국가지정연구실사업으로 수행 된 연구임 (No.R0A-2006-000-10225-0)

인터넷의 수요가 증가함에 따라 사용자에게 맞춤형 서비스를 제공하기 위해서 사용자의 접근 패턴이 숨어있는 네트워크 패킷 데이터의 분석이 필연적이다. 네트워크 장비(Network device)로 부터 수집된 패킷을 데이터 마이닝 함수(Data mining function)에 적용하여 얻어진 결과 패턴 집합은 다수의 사용자들을 통해 발견된 정보로서 이를 이용하여 사용자들의 선호도 및 목적에 적합한 서비스를 제공할 수 있다.

전처리 작업 없이 네트워크 패킷 원시 데이터에 대해 마이닝을 수행하는 경우 유용한 결과 패턴 집합을 얻을 수 없다. 수집된 패킷 데이터에 데이터 패킷 페이로드(payload)뿐만 아니라 네트워크 운영을 위해 수많은 프로토콜 메시지 패킷, 경유하는 패킷이 포함 되어있기 때문이다. 그리고 전처리 되지 않은 데이터를 마이닝 하고 얻은 결과 패턴집합에는 불필요한 정보들이 포함되어있어 유용한 마이닝 결과 패턴 집합을 도출하기 어렵고, 별도의 후처리 작업(Post-processing)이 요구된다.

사전에 관심 없는 데이터를 필터링(Filtering) 하고 결과 패턴 집합에 관련된 원시 데이터를 유지하도록 데이터를 가공하여 유지함으로써 마이닝 함수(Mining Function)에서 결과 패턴집합이 쉽게 얻어지도록 전처리 작업이 수행되어야 한다.

본 논문에서는 네트워크 장치에서 수집된 패킷 데이터에 대한 데이터 마이닝 수행 시 데이터의 결과 패턴 집합의 정확도의 손실 없이 패킷 데이터를 효과적으로 전처리하는 기법을 제안한다

2. 본론

2.1 실험 데이터

데이터는 2개의 스위치 장비에서 패킷 데이터를 152일(약 5개월) 동안 수집하였고 수집된 패킷의 카디널리티(Cardinality)는 11,143,662개를 데이터베이스에 표1과 같은 테이블로 저장하였다.

표 1. 패킷 데이터의 스키마

Attribute Name	Type	Sample data
Time stamp	Timestamp	20080529195811
Source IP	Varchar2(15)	127.0.0.1
Source port	Integer	56
Destination IP	Varchar2(15)	999.0.0.1
Destination port	Integer	80
Number of packet	Number(5)	1
Packet capacity	Number(20)	359
Protocol type	Number(5)	17

2.2 전처리의 필요성

네트워크에서 수집된 패킷 데이터의 전처리 작업을 거치지 않고 데이터 마이닝 함수에 입력해 유용한 결과 패턴 집합을 얻는 것은 불가능하다.

첫 번째 이유는 네트워크 장치 간에는 안정적인 패킷의 교환을 위해 불규칙적으로 나타나는 제어 패킷(control packet) 때문이다. 제어 패킷은 전체 데이터 분포에서 27.47% 가량 차지하며, 각 프로토콜 타입 별로 낮은 밀도의 분포를 가진 희박한 데이터(Sparse data)이기 때문에 마이닝결과 패턴집합을 얻기 어렵다.

그림1에서 이러한 패킷 데이터들의 프로토콜 타입(Protocol Type) 별 분포를 나타내었다. 프로토콜 타입 6은 LAN의 접속을 위해 사용되는 IEEE802 이며, 17은 HDLC로 데이터 프레임 내에 흐름제어(Flow control)와 오류정정(Error correction)을 위해 사용되는 것으로 일반적인 데이터를 전송하는 패킷과 밀접히 연관된다. [2][3]

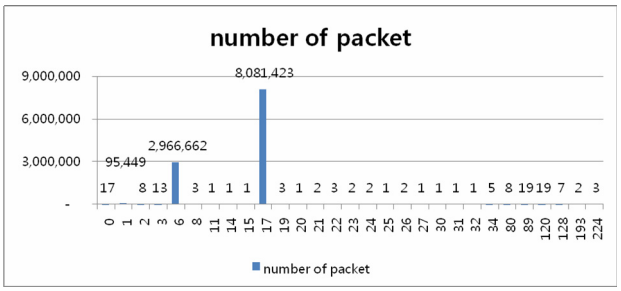

그림 1. 프로토콜 타입에 따른 패킷 분포

표 2. 동일한 목적의 패킷의 중복 출현

Timestamp	Source IP	Destination IP	Protocol Type
000000	165.0.0.1	255.0.0.4	5
000001	165.0.0.1	255.0.0.4	5
000600	165.0.0.1	255.0.0.4	5
000620	165.0.0.5	255.0.0.4	6
004600	165.0.0.1	255.0.0.4	5
005000	165.0.0.1	255.0.0.4	5
010800	165.0.0.5	255.0.0.4	6

프로토콜 타입 17인 패킷을 제외한 나머지 프로토콜 타입 패킷은 LAN 장치간에 안정적인 패킷을 전송하기 위해 전송되는 제어 패킷이거나 네트워크 장치를 경유하는 패킷이다. 예를 들어 그림1의 프로토콜 타입 1은 대상 장치가 네트워크에 연결되어 있는지 여부를 확인하는 패킷 집합이다.

결과적으로 사용자의 의도에 의해 전송된 패킷의 분석을 위해서 사전에 프로토콜 타입에 따라 선택 연산(Selection Operation)을 수행해야 한다.

두 번째 이유는 데이터 전송이 발생하는 한 사건(Event)에 대해 동일한 패킷이 중복되어 출현한다. 표2의 timestamp가 000000부터 005000까지 source IP address는 165.0.0.1, destination은 255.0.0.4인데 사용자의 한 요청이 반복되어서 나타났다. 이와 같은 중복된 패킷은 데이터 전송 시 일정한 크기의 패킷으로 나누어서 전송된 것이다. 하지만 이런 중복된 투플은 마이닝 함수에서 여과 없이 입력되면 단일 트랜잭션으로 여겨져 결과 패턴 집합이 그림2와 같이 편향(Skew)된 형태로 나타난다.

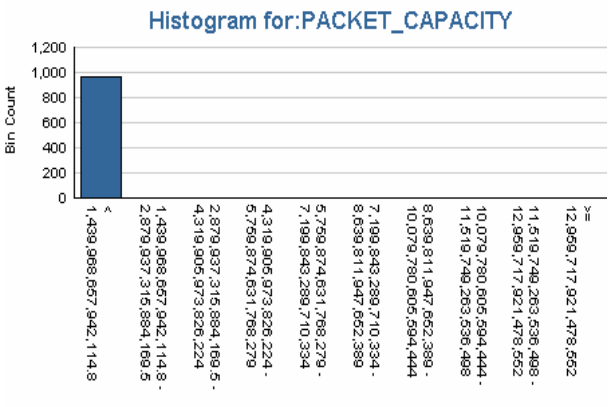

그림 2. Packet Capacity의 데이터 분포

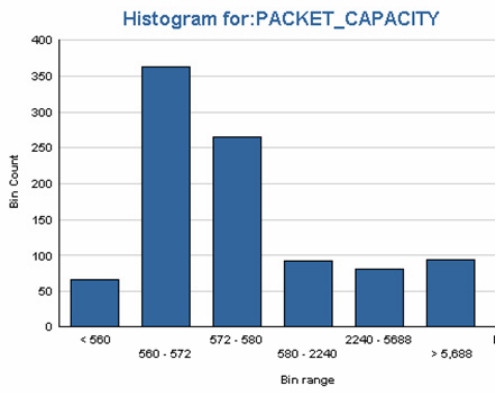

그림 3. packet capacity의 6부류 그룹화

2.3 전처리 방법

2.3.1 편향된 데이터의 이산화(Discretizing)

그림 2는 Packet Capacity 값의 분포를 10개의 그룹으로 나타낸 것이다. 대부분 값의 값이 가장 좌측 그룹에 분포 되어있다는 것을 알 수 있다. 그리고 Packet Capacity 값이 최대 14자리에 달하고, 이로 인해 작은 값을 가진 Packet Capacity에서 발견할 수 있는 패턴 결과 집합을 놓칠 수 있다.

$$Packet\_capacity = \left\{ x = packet\_capacity \left\| \frac{x - Min(x)}{Max(x) - Min(x)} * 100 \right\rceil \right\} = [0,1]$$

위 식을 이용해서 Packet Capacity의 전체 분포에서 중간 값 이상 Packet Capacity 투플만 추려내어 1 값으로 정규화 한다. 1인 값인 투플을 집계연산을 수행 후 그룹화 하여 그림3과 같이 6개의 그룹으로 표현하였다

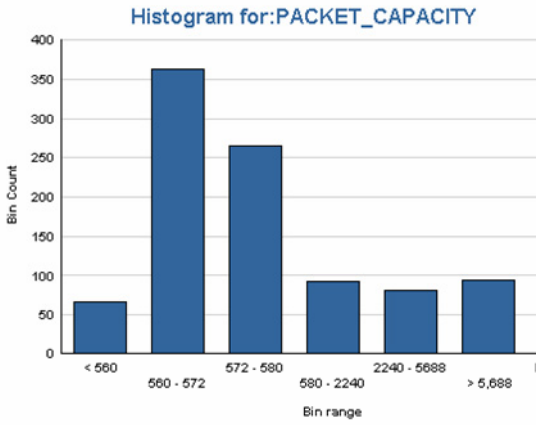

그림 4. 전처리 후 packet capacity의 분포

표 3. 중복된 트랜잭션의 단일화

Timestamp	Source IP	Destination IP	Protocol Type
000000	165.0.0.1	255.0.0.4	5
000620	165.0.0.5	255.0.0.4	6

표 4. 중복된 트랜잭션의 집계함수 적용

Source IP	Rank	Protocol Type	Packet Capacity
0.0.0.0	1	109501	4.4029E+13
0.0.0.0	2	31	32522492
0.0.0.0	3	1	76
10.0.0.113	1	22	205020
10.0.0.12	1	4	5472
10.0.0.13	1	3	5472
10.0.1.2	1	9	41382

2.3.2 데이터 그룹화

데이터를 전송할 시 전달되는 패킷은 분할되어 전송되어 중복된 패킷이 존재한다. 표3과 같이 모든 속성이 중복된 경우 Group by문을 이용하여 표2와 같이 중복을 제거해서 단일 트랜잭션으로 변환한다.

표4와 같이 모든 속성이 중복되지 않은 경우는 Packet Capacity를 기준으로 순위를 매겨 정렬하고 같은 그룹의 패킷집합 중에서 가장 큰 Packet Capacity값을 선택하거나 집계 함수를 적용한다.

2.3.3 계층 특징 구체화

스위치 장치에서는 패킷 데이터의 집합이 백만 초(ms) 단위 시간대 별로 수많이 통과된다. 앞의 두 가지 전처리 기법이 수행된 후 투플 간 발생시간의 간격은 백만 초 단위로 남아있다.

트랜잭션 발생 시점을 분석하고자 할 때 표5 데이터의 Timestamp 속성을 표6과 같이 분석 시각

표 5. 시간 별로 발생된 패킷 집합

Timestamp	Source IP
2008-05-29 PM 7:58:11.000000	200.0.0.1
2008-05-29 PM 7:58:11.000000	200.0.0.2
2008-05-29 PM 7:58:11.000000	200.0.0.2
2008-05-29 PM 7:58:11.000000	200.0.0.3
2008-05-29 PM 7:58:11.000000	200.0.0.4

표 6. 트랜잭션의 발생 시간순서로 구체화

Source IP	0-3	4-7	8-11	12-15	16-19	20-24	Packet Capacity
200.0.0.1	0	0	0	0	1	0	359
200.0.0.2	0	0	0	0	1	0	296
200.0.0.2	0	0	0	0	1	0	69
200.0.0.3	0	0	0	0	1	0	69
200.0.0.4	0	0	0	0	1	0	22154

별로 속성을 나누어 만들고 트랜잭션의 발생한 시각에 따라 트랜잭션의 수를 집계하여 표6과 같이 트랜잭션을 구체화(Characterization) 한다.

3. 결론

본 논문에서는 네트워크 장치에서 수집된 패킷 데이터로부터 데이터 마이닝 함수를 통해 유용한 패턴 집합을 얻을 수 있도록 패킷 데이터를 효과적으로 전처리 하는 기법을 제안하였다. 제안된 접근 방법은 중복(duplicate), 편향(Skew), 희박한(Sparse) 패킷 데이터 집합을 그룹화와 중복된 패킷 데이터 투플 제거를 통해 단일 트랜잭션화 하고 특성을 최대화 하여 마이닝 함수에서 결과 패턴 집합이 나타나도록 전처리 하였다.

[참고문헌]

[1] C. Carter and H. Hamilton. Efficient attribute-oriented generalization for knowledge discovery from large databases. IEEE Trans. Knowledge and Data Engineering, 10:193-208, 1998.

[3] Plummer, D., "An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48-bit Ethernet Addresses for Transmission on Ethernet Hardware", STD 37, RFC 826, MIT-LCS, 1982.

[4] Tanasa D and Trousse B. “Advanced data preprocessing for intersites Web usage mining”. IEEE Intelligent System, 2004.