



## 스트림 데이터 마이닝(Stream Data Mining)

서울대학교 심규석\*\* · 김철연\* · 우정철

### 1. 스트림 데이터 마이닝이란?

기존의 응용 프로그램에서는 데이터 원본들이 주로 디스크나 테잎에 데이터가 들어있다는 관점 하에 이들에 대한 순차 접근이나 임의 접근이 가능하다고 보았다. 하지만 최근에 이르러 많은 응용 분야에 있어서 이러한 관점이 더 이상 유효하지 않게 되었다. 따라서 기존의 데이터 원본에 대한 관점이 유효하지 않은 이러한 분야에서는 질의 처리나 데이터 마이닝에 관한 새로운 많은 issue들이 발생하게 되었는데, 특히 데이터 마이닝에 관련된 이러한 issue들에 대한 연구를 스트림 데이터 마이닝이라고 부른다. 이러한 응용 분야의 예로 네트워킹 분야를 들 수 있는데, 이 분야는 라우터와 같은 굉장히 많은 네트워크 구성 요소들과 관련된 프로토콜 및 데이터 교환 트래픽들로 구성되며, 통행량의 조절, 장애 처리, 용량 모니터 및 조절, 침입 탐지, 주문형 회계와 청구서 발송 등의 작업들을 요구한다. 이러한 작업들은 정보들이 네트워크에서 흘러다니는 동안에 빠르게 처리가 되어야만 하나, 라우터 타입에 따라서 통행량의 속도가 다르고 이러한 정보들이 너무 방대하므로 이들을 모두 일정한 저장 공간에 보관하여 처리하기가 불가능하다. 또 다른 응용 분야로는 환경이나 군사 분야를 들 수 있다. 이러한 분야는 온도나, 기압, 혹은 트래픽의 흐름 등과 같은 것을 측정하는 여러 개의 센서들이 도처에 분산되어 있으며, 이러한 센서들에서 수집되는 데이터들이 중앙 사이트로 계속해서 보내진다. 이러한 응용 분야에서도 데이터가 계속해서 끊임없이 도착하는데, 이러한 데이터의 특징은 데이터 도착 속

도가 가변적이며, 많은 소스들이 데이터를 전송하고, 또한 데이터의 사이즈가 아주 거대해서 무한한 데이터 스트림 (infinite data stream)을 형성하기 때문에 디스크나 테잎 같은 곳에 저장을 할 수가 없다는 것이다. 이러한 스트림 데이터를 처리하기 위하여 새로운 스트림 데이터 모델을 제안하게 되었는데 이 모델에서는 데이터의 양이 아주 크거나 또는 무한하며 데이터에 대해 순차적인 액세스만이 가능하다고 가정하고 제한된 용량의 디스크나 메모리 등을 이용해서 질의 처리나 데이터 분석을 처리한다. 또한 이 모델에서는 데이터가 끊임없이 도착하여 제한된 용량에 저장해두지 않은 데이터는 잃어버리게 되는데, 이러한 잃어버린 데이터들은 다시 볼 수가 없다고 가정한다. 그렇기 때문에 스트림 데이터 모델에서는 데이터의 요약 정보만을 저장하고 있다가 질의를 처리할 때 이 요약 정보를 이용하게 된다.

스트림 데이터에 관련된 연구의 효시는 1980년에 Munro와 Patterson이 “Selection and Sorting with Limited Storage”라는 제목으로 발표한 논문이다. 이 논문은 데이터에 대한 중간 값(Median)을 구하기 위하여 데이터 스캔하는 횟수와 이에 필요한 메모리의 최소 스페이스에 대한 연구를 다루었다. 또한 1985년에는 Flajolet와 Martin이 [FM85]에서 데이터에서 서로 다른 원소(element)의 수를 셀 경우, 정확한 수를 세기 위해서는 N개를 세기 위해서  $\log N$  비트의 메모리가 필요하지만, probabilistic counting을 이용하여 훨씬 더 많은 개수를 근사적으로 셀 수 있도록 하는 아이디어를 제안하였다. 그 이후에 [AMS99], [GM99], [HNSS96], [MRL98] 등 많은 논문들이 스트림 데이터 모델에 관하여 연구하였다. 지금까지 나열한 연구들은 모든 데이터에 대해서 문제를 정의하고 해결하는 것이지만 [BW01]과 [GGR01]에서는 항상

\* 학생회원

\*\* 정회원

최근부터 과거의 어느 시점까지의 윈도우(Window)의 데이터에 대해서 질의를 처리해 주는 문제에 대하여 연구하였다. 스트림 데이터 모델과 이를 위한 데이터베이스 시스템에서 발생되는 여러 가지 연구 주제에 관해서는 Stanford 대학의 교수들과 대학원 학생들이 공동으로 발표한 [BBD+02]에 자세히 소개되어 있으므로 관심있는 사람들은 참고하기 바란다.

스트림 데이터에 관련된 연구가 어떤 것인지 이해를 돋기 위하여 간단한 예를 보이기로 한다. N개의 숫자가 하나씩 도착할 때 정확하게 그 중의 중간 값(Median)을 찾는 문제를 생각해 보자. 이 문제를 풀려면 어떤 정도의 스페이스가 최소한 필요한지를 생각해보자. N개의 숫자가 있을 때  $N/2$ 개의 숫자를 저장하지 못한다면 중간 값을 정확하게 찾을 수 없다. 다시 말해서 공간 복잡도(Space Complexity)가 데이터 사이즈 N에 대하여 Linear한 함수로 나타나기 때문에 정확한 중간 값을 구하려면 많은 스페이스가 필요함을 알 수 있다. 만일 제한된 작은 스페이스를 이용하여 중간 값을 찾으려면 할 수 없이 근사적인 중간 값을 구하여야만 하는데, 실제로 사용할 수 있는 알고리즘을 만들기 위해서는 근사 값의 신뢰도가 명확히 계산되어야 한다. 이러한 이유로 스트림 데이터 연구는 컴퓨터 이론에 관한 많은 지식을 요구하게 된다.

## 2. 스트림 데이터 마이닝 알고리즘 개발에 사용되는 기법들

그러면 스트림 데이터에 관련된 알고리즘을 만들 때 사용되는 기본적인 방법에 대하여 소개한다. 그 방법들을 나열하면, (1) 표본 추출 기법(Sampling), (2) 요약된 계산한 상태를 보관하는 방법, (3) Divide-and-Conquer 방법, (4) Embedding 방법 등이 있다.

### 2.1 표본 추출 방법

이 방법은 데이터가 계속해서 들어올 때 새로 들어온 데이터를 표본에 넣어야 할지 말아야 할지를 결정하는 것이다. 이 방법의 핵심은 그 동안 들어온 모든 데이터의 분포가 모집단의 분포와 유사한 샘플이 만들어지도록 하는 것이다. 만일 이러한 성질이 보장된다면, 이 표본 추출된 데이터를 이용해서 처리될 수 있으므로, 필요한 모든 질의나 분석은 이 작은 표

본 데이터를 가지고 하면 적은 양의 메모리만을 이용해서 큰 데이터에 대해 처리할 수 있게 된다. 이미 오래전에 이러한 표본 추출 기법을 사용하는 알고리즘이 [Vitter85]에서 제안되었다.

표본 추출 기법이 어떻게 스트림 알고리즘에서 사용될 수 있는지 I-번쨰의 수를 찾는 간단한 예제를 통해서 생각해 보자. 전체 데이터의  $S = O((\log n)^{2/d})$ 에 해당되는 샘플 데이터가 있다면, 이 샘플을 소팅을 한 후에 그 샘플에서  $i/S$ -번쨰 수를 구하면 이 수는 전체 수의 개수가 N인 원래 스트림 데이터에서  $i-d*N$ 과  $i-d*N$ 번째 사이의 수가 될 확률이 아주 높음이 [MRL98]에서 증명되어 있다. 표본 추출 기법은 모든 문제에 항상 사용될 수 있는 것은 아니다. 간단한 예로, maximum이나 minimum을 찾는 문제 표본 추출 기법을 사용하는 것은 적절하지 못한 방법이라고 할 수 있다.

### 2.2 요약된 정보를 보관하는 기법

이것은 전체 데이터를 있는 그대로 다 저장하는 것이 아니라 가능한 필요한 만큼의 데이터만을 가지고 있는 방법이다. 예를 들어 [GK01]을 보면 Quantile들의 제한된 스페이스를 가지고 요약 정보를 계산하는 온라인 알고리즘이 소개되어 있다. 이 방법은 트리 구조를 이용해서 요약 정보를 보관하는데 스페이스가 충분하면 항상 I-번쨰 수를 정확히 계산할 수 있지만 스페이스가 부족하게 될 때는 너무 자세하게 나타낸 부분을 트리에서 버리는 방법이다. 또 한가지 예는 [GKS01]에 나와 있는 스트림 데이터에 대한 히스토그램을 만드는 알고리즘에서 사용된 방법이다. V-Optimal 히스토그램 알고리즘은 다이나믹 프로그래밍 기법을 이용한 알고리즘인데, 이 알고리즘에서 계산된 테이블에서 적당한 크기의 에러를 보장할 수 있을 만큼을 잘라냄으로써 사용하는 기억 용량을 줄이면서, 동시에 속도를 빠르게 하는 방법이었다.

### 2.3 Divide-and-Conquer 방법

이 방법은 데이터 전체를 이용하지 않고 들어오는 데이터에 대해서만 일정한 블록단위로 모으다가 데이터가 블록에 다 차면, 그 블록안의 데이터와 그 전 까지 계산된 것을 함께 사용해서 incremental하게 다시 계산하는 것을 말한다. 예를 들어서, [GMMO00]에서는 스트림 데이터에 대한 군집화(Clustering) 알고

리즘을 디자인하는데 이 기법이 사용되었다.

## 2.4 Embedding 기법

이 방법은 우리가 원래 풀려는 문제를 훨씬 쉽게 풀릴 수 있는 다른 문제로 변환하는 것을 말한다. 이 방법을 사용하려면, 주어진 데이터를 다른 문제가 처리할 수 있는 형태로 매핑(mapping)을 시켜 주어야 하며 그 변환된 다른 문제가 매핑된 데이터를 이용해서 답을 구한 후에 그 답을 원래의 문제에 대한 답으로 다시 매핑(mapping) 해 주어야만 한다. 예를 들어서 자연수가 계속해서 도착할 때에 각 수들의 frequency들의 제곱을 해서 다 더한 값을 구하는 문제를 풀려면, 이 답은 순서에 상관없는 계산을 하면 되므로, 스케치(Sketch)라 불리는 히스토그램을 가지고만 있으면 원하는 답을 쉽게 구할 수 있다. 이러한 기법을 이용한 연구들은 [AMS], [FKSV99], [Indyk00] 등이 있다.

## 3. 최근에 발표된 연구들

여기서는 최근에 스트림 데이터 마이닝에 관하여 발표된 연구들을 살펴보기로 한다. 먼저 군집화 알고리즘에 대해서 살펴보면, 기존의 데이터 마이닝 분야의 군집화 알고리즘들 중에도 그대로 스트림 데이터의 군집화에 사용 가능한 것들이 있다. 우선 Guha, Rastogi, Shim이 발표한 CUR[GRS98]에서는 데이터를 일정한 크기로 분할한 후에 각각의 분할에 대하여 군집화를 하고 모든 분할의 군집화된 결과들을 가지고 다시 군집화를 수행해서 전체 데이터를 군집화 하는데 이러한 아이디어도 스트림 데이터를 군집화 하는데 사용될 수 있다.

또한 BIRCH[ZRL96]에서는 전체 데이터를 스캔하면서 메인 메모리에 들어갈 정도의 크기로 전체 데이터의 요약 정보를 만든 후 (pre-clustering) 이 요약 정보를 이용해서 다시 전체 데이터의 군집화를 수행하는데 이러한 아이디어도 사용될 수 있다. 그 외에도 스트림 데이터의 군집화를 위한 다른 기존의 여러 가지 알고리즘이 있지만 이런 것들 모두 사용 가능한 스페이스를 제한할 때는 어떤 quality의 결과를 보장할 수 없기 때문에 스트림 데이터 마이닝에 사용하는데는 많은 어려움이 있다. 스트림 데이터에 관한 군집화 알고리즘으로는 [GMM00]에서 소개

된 알고리즘이 있는데 이것은 원하는 크기의 에러를 보장하면서  $O(n)$ 까지 시간 복잡도가 보장된다. 그 외에도 [OMM+02] [COP03]에서 스트림 데이터의 군집화 알고리즘에 관하여 발표되었다.

스트림 데이터를 요약해서 보관하는 방법으로는, 히스토그램과 웨이블릿(wavelet)이 대표적이라고 할 수 있다. 히스토그램중에서 많이 사용되는 V-Optimal 히스토그램 알고리즘은 전체 데이터 사이즈가  $n$ 이고  $B$ 개의 버킷을 사용하는 최적한 히스토그램을 만들때 수행시간 복잡도가  $O(n^2B)$ 이기 때문에 스트림 데이터에서는 사용될 수가 없다. Guha와 Koudas 와 Shim이 [GKS01]에 최적 에러의  $(1 + \epsilon)$ 이내의 에러를 보장하면서  $O(B^2\epsilon^{-1}\log n)$  스페이스를 사용하고  $O(B^3n\epsilon^{-1}\log n)$  시간이 걸리는 알고리즘을 발표하였다. 또한 [GK02]에서 좀더 개선된 알고리즘을 발표하였다. 그 외에도 웨이블릿이나 Fourier 계수로 스트림 데이터를 요약하는 여러 가지 방법들이 [GGI+02] 제안되었다.

스트림 데이터에 관한 분류 알고리즘은 [DH00]에서 제안되었다. 여기서는 처음에 들어오는 일부 튜플(tuple)을 가지고 루트노드의 분할 조건을 결정하고 또 그 다음에 들어오는 튜플들을 가지고 또 리프노드의 분할 조건을 결정하는 방식으로 수행한다. 몇 개의 튜플을 사용해야 정확하게 각 노드의 분할 조건을 잘 맞추게 될지 알아야 알고리즘이 정확히 수행되는데 이를 위해 Hoeffding bound라는 것을 이용하여 최소한 몇 개의 튜플을 각각의 분할 조건을 찾는데 사용해야 할지를 결정한다.

스트림 데이터를 위한 연관 규칙 알고리즘도 [MM03] 최근에 발표되었다. 이 논문에서는 최소 지지도를 만족하는 아이템 셋(item set)을 계산하되 결과 중에는 사용자가 허락하는 범위 내의 오차가 있는 결과도 포함될 수 있다. 즉, 이 오차를  $\epsilon$ 이라고 하고, 최소 지지도를  $s$ 라고 하고, 전체 데이터를  $N$ 이라고 했을 때, 결과 값에서는 나타나는 횟수가  $sN$  이상이 되는 것을 모두 찾아주고, 동시에  $(s-\epsilon)N$  이상이 되는 데이터들 중에서 일부도 찾아주게 된다. 이를 위해서 이들은 ' $7/\epsilon$ ' 만큼의 메모리를 사용하는 카운팅 알고리즘(Lossy counting)을 제안한다. 그리고 최소 지지도를 만족하는 아이템 셋을 생성하기 위해서 적당한 크기의 버퍼를 사용해야 하는데, 이 버퍼를 많이 쓸수록 더 빠른 수행속도를 보인다.

여러 개의 시계열 (time series) 스트림 데이터를 위한 알고리즘은 [ZS02]에서 소개되었는데 여기서는 시계열 데이터 간에 여러 가지 통계적 상관도 (correlation)를 자동적으로 계산해 주는 것이 목적이이다. 이 논문에서는 금융 시계열 스트림 데이터의 특징을 DFT(Discrete Fourier Transform) 계수 (Coefficient)를 이용하여 원하는 정확도에 따라서 나타내는데 데이터가 계속 들어올 때 이 계수들을 점진적 (incremental)으로 계산하도록 해준다.

## 참고문현

- [AMS99] Noga Alon, Yossi Matias, Mario Szegedy: The Space Complexity of Approximating the Frequency Moments. JCSS 58(1): 137–147 (1999)
- [BW01] Shivnath Babu, Jennifer Widom: Continuous Queries over Data Streams. SIGMOD Record 30(3): 109–120 (2001)
- [BBD+02] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, Jennifer Widom: Models and Issues in Data Stream Systems. PODS 2002: 1–16
- [COP03] Moses Charikar, Liadan O'Callaghan, Rina Panigrahy: Better streaming algorithms for clustering problems. STOC 2003
- [DH00] Pedro Domingos, Geoff Hulten: Mining high-speed data streams. KDD 2000
- [FKSV99] Joan Feigenbaum, Sampath Kannan, Martin Strauss, Mahesh Viswanathan: An Approximate L1-Difference Algorithm for Massive Data Streams. FOCS 1999: 501–511
- [FM85] Philippe Flajolet, G. Nigel Martin: Probabilistic Counting Algorithms for Data Base Applications. JCSS 31(2): 182–209 (1985)
- [GGI+02] Anna C. Gilbert, Sudipto Guha, Piotr Indyk, Yannis Kotidis, S. Muthukrishnan, Martin Strauss: Fast, small-space algorithms for approximate histogram maintenance. STOC 2002
- [GGR01] Venkatesh Ganti, Johannes Gehrke, Raghu Ramakrishnan: DEMON: Mining and Monitoring Evolving Data. TKDE 13(1): 50–63 (2001)
- [GM99] Phillip B. Gibbons, Yossi Matias: Synopsis Data Structures for Massive Data Sets. SODA 1999: 909–910
- [GK01] Michael Greenwald, Sanjeev Khanna: Space-Efficient Online Computation of Quantile Summaries. SIGMOD Conference 2001
- [GK02] Sudipto Guha, Nick Koudas: Approximating a Data Stream for Querying and Estimation: Algorithms and Performance Evaluation. ICDE 2002
- [GKS01] Sudipto Guha, Nick Koudas, Kyuseok Shim: Data-streams and histograms. STOC 2001: 471–475
- [GMMO00] Sudipto Guha, Nina Mishra, Rajeev Motwani, Liadan O'Callaghan: Clustering Data Streams. FOCS 2000: 359–366
- [GRS98] Sudipto Guha, Rajeev Rastogi, Kyuseok Shim: CURE: An Efficient Clustering Algorithm for Large Databases. SIGMOD Conference 1998
- [HNSS96] Peter J. Haas, Jeffrey F. Naughton, S. Seshadri, Arun N. Swami: Selectivity and Cost Estimation for Joins Based on Random Sampling. JCSS 52(3): 550–569 (1996)
- [Indyk00] Piotr Indyk: Stable Distributions, Pseudorandom Generators, Embeddings and Data Stream Computation. FOCS 2000: 189–197
- [MM02] Gurmeet Singh Manku, Rajeev Motwani: Approximate Frequency Counts over Data Streams. VLDB 2002
- [MRL98] Gurmeet Singh Manku, Sridhar Rajagopalan, Bruce G. Lindsay: Approximate Medians and other Quantiles in One Pass and with Limited Memory. SIGMOD Conference 1998: 426–435
- [MP80] J. Ian Munro, Mike Paterson: Selection and

- Sorting with Limited Storage. TCS 12: 315-323 (1980)
- [OMM+02] Liadan O'Callaghan, Adam Meyerson, Rajeev Motwani, Nina Mishra, Sudipto Guha: Streaming-Data Algorithms for High-Quality Clustering. ICDE 2002
- [Vitter85] Jeffrey Scott Vitter: Random Sampling with a Reservoir. TOMS 11(1): 37-57 (1985)
- [ZRL96] Tian Zhang, Raghu Ramakrishnan, Miron Livny: BIRCH: An Efficient Data Clustering Method for Very Large Databases. SIGMOD Conference 1996

- [ZS02] Yunyue Zhu, Dennis Shasha: StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time. VLDB 2002

### 김 철 연

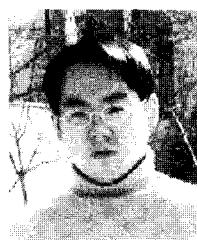


1996 서울대학교 공과대학 컴퓨터공학  
(학사)  
1998 서울대학교 협동과정 인지과학(석  
사)  
1999~2001 (주)아이큐브 연구원  
2001~2002 (주)엔트로스 기술이사  
2003~현재 서울대학교 공과대학 전기  
컴퓨터공학부(박사과정)  
관심분야 : 데이터마이닝, XML, 스트림  
데이터, 인공지능  
E-mail : cykim@kdd.snu.ac.kr

### 심 규 석



1986 서울대학교 공과대학 전기공학(박  
사)  
1988 University of Maryland 전산  
학(석사)  
1993 University of Maryland 전산  
학(박사)  
1994~1996 IBM Almaden 연구소 연구  
원  
1996~2000 Bell 연구소 연구원  
1996~2002 한국과학기술원 전자전산학  
과 조교수  
2002~현재 서울대학교 공과대학 전기컴퓨터공학부 부교수  
관심분야 : 데이터마이닝, XML, 스트림데이터, 퀼리처리 및 최적화,  
OLAP  
E-mail : shim@ee.snu.ac.kr



2002 서울대학교 자연대학 물리학과(학  
사)  
2003~현재 서울대학교 공과대학 전기  
컴퓨터공학부(석사과정)  
관심분야 : 데이터마이닝, XML, 스트림  
데이터, 히스토그램  
E-mail : jcwoo@kdd.snu.ac.kr

### The International Conference on Infor mation Networking(ICOIN) 2004

- 일 자 : 2004년 2월 18~20일
- 장 소 : Marriott Hotel(부산)
- 주 최 : 정보통신연구회
- 상세안내 : <http://www.icoin2004.or.kr>