ISSN 1598-0170 (Print)
ISSN 2287-1136 (Online)
http://www.jksii.or.kr

맵리듀스 기반 DFP-Tree를 이용한 클러스터링 알고리즘☆

Clustering Algorithm using the DFP-Tree based on the MapReduce

서 영 원[1] 김 창 수[1*]
Young-Won Seo Chang-soo Kim

요 약

빅 데이터가 이슈화됨에 따라 데이터 분석의 결과를 기반으로 동작하는 많은 응용들이연구되고 왔고, 대표적인 응용들은 전자상거래 시스템의 상품 추천 서비스, 검색 엔진에서의 검색 서비스, 소셜 네트워크 서비스에서의 친구 추천 서비스 등이 있다.

본 논문은 기존의 데이터 마이닝 기법 중 데이터 집합에서 나타나는 유사한 패턴들을 마이닝하는 빈발 패턴 트리와 컴퓨터 과학의 이론에 기초한 결정트리를 결합하여 결정 빈발 트리 알고리즘을 제안한다. 이는 기존의 빈발 패턴 트리 알고리즘은 패튼 트리에서 패턴 생성에 대한 정확성은 보장되나 소셜 데이터처럼 다양한 패턴이 나타는 데이터에 대해서는 많은 수의 패턴들을 생성시켜 분석에 대한 어려움이 있어, 서브트리들과의 수렴 여부를 판단하는 모델로 변형시켜 문제를 개선한다. 또한 맵리듀스로 모델링하여 분산처리를 통한 고속 처리 알고리즘을 제시한다.

☞ 주제어 : 데이터 마이닝, 빈발 패턴 트리, 클러스터링 알고리즘, 분산 처리 시스템, 추천 시스템, 맵 리듀스

ABSTRACT

As BigData is issued, many applications that operate based on the results of data analysis have been developed, typically applications are products recommend service of e-commerce application service system, search service on the search engine service and friend list recommend system of social network service.

In this paper, we suggests a decision frequent pattern tree that is combined the origin frequent pattern tree that is mining similar pattern to appear in the data set of the existing data mining techniques and decision tree based on the theory of computer science. The decision frequent pattern tree algorithm improves about problem of frequent pattern tree that have to make some a lot's pattern so it is to hard to analyze about data. We also proposes to model for a Mapredue framework that is a programming model to help to operate in distributed environment.

☞ keyword : SData Mining, Frequent-Pattern tree, Clustering Algorithms, Distributed Processing System, Recommendation System, Map Reduce

1. 서 론

최근 글로벌 IT기업들은 기계학습 분야에 많은 투자를 하는 동향을 보이고있다[1]. 구글은 영국의 기계학습 관련 업체인 딥 마인드를 인수하고 페이스북은 기계학습 분야의 최고 권위자 중 한명인 뉴욕대학교의 얀 레쿤 교수를 인공지능 연구소 수장으로 영입하였다. 또한 트위터는 이미지 관련 기계학습 전문 기업인 매드비츠를 인수함으로써 업계의 흐름을 따라가고 있다. 이러한 현상은 위 기업들의 특성과 많은 관련이 있다. 구글의 경우 개인이 읽은 웹 페이지들을 분석하여 개인에게 특화된 검색결과와 광고, 뉴스페이지를 보여 주고 있으며[2] 페이스북의 경우 사용자들의 친구 목록과 뷰 페이지를 분석하여 여러 사용자 간의 연관성을 그래프 형태의 구조로 모델링하고 분석하여 친구추천서비스 제공하고 광고를 보여주고 있다. 또한 최근에는 사용자가 업로드한 사진들을 분석하여 사진에 나타나는 사람 얼굴에 대해 자동태깅을 하는 기능 또한 제공하고 있다.[3] 이러한 기술들은 기계학습 분야의 발전과 현재 사용자들에 의해 발생되는 데이터들의 특성과 관련있다. 그리고 이러한 특성을 가진 데이터를 다루는 기술을 빅 데이터[4]라고 부르고 있다.

[1] Department of IT Convergence and Application Engineering, Pukyong National University, Busan, 599-1, Korea
* Corresponding author (cskim@pknu.ac.kr)
[Received 10 September 2015, Reviewed 11 September 2015, Accepted 10 November 2015]
☆ 이 논문은 부경대학교 자율창의학술연구비(2015)에 의하여 연구되었음.

빅 데이터는 대용량 데이터의 수집, 관리, 저장, 분석을 포괄하는 기술이다. 이미 빅데이터를 위한 많은 도구들이 개발되어 왔으며 경영, 의료, 기계등의 산업분야와 융합됨으로써 가치를 입증하였다. 거의 모든 서비스들은 인터넷을 통해 사용가능해짐과 더불어 유비쿼터스 컴퓨팅의 발전으로 언제어디서나 인터넷에 대한 접근성이 좋아져 빅데이터는 빠른 속도로 발전하였다. 그리고 웹에서 발생하는 로그, 사용자들이 업로드하는 텍스트, 사진 등의 데이터는 빅데이터의 발전에 큰 역활을 하고 있다.

본 논문은 여러가지 데이터 분석 기법 중 유사한 데이터를 군집화 시키는 클러스터링 기법을 제시한다. 클러스터링은 비교사 학습 기법[5]으로 크게 분할 접근과 계층적 접근으로 구분 할 수 있는데 분할 접근은 범주 함수를 최적화시키는 K개의 분할 영역을 결정해 나가는 방법으로 유클리드 거리 측정법에 기반한다. 계층적 접근은 각 데이터를 하나의 클러스터로 입력하고 각 클러스터의 유사도를 계산하여 합병 혹은 분할해 나가는 방식이다. 본 논문에서는 데이터의 집합에서 자주 나타나는 패턴을 검색하는 기존의 데이터 마이닝 기법중 하나인 FP-Tree를 이용하여 계층적 접근의 새로운 클러스터링 알고리즘을 제시한다. 또한 대용량 데이터에 대한 처리를 위해 맵리듀스 프로그래밍 모델로 이를 병렬처리할 수 있는 형태로 구현한다. 이후 다른 클러스터링 알고리즘과 결정 빈발 패턴 트리의 성능비교를 통하여 수행 시간 및 클러스터링 성능을 평가한다.

2. 관련연구

2.1 빈발 패턴 마이닝

데이터 마이닝에서 빈발 패턴을 마이닝하는 기법들은 활발히 연구되는 분야 중 하나이다. 이는 데이터 집합에 대한 본질과 특성들을 찾는 방법으로 이어지기 때문인데 대표적으로 Apriori 알고리즘[6]이나 빈발 패턴 트리 알고리즘이 있다. Apriori 알고리즘은 그래프 알고리즘 중 넓이 우선 탐색 방식에 기반하여 동작하는 알고리즘으로 데이터 집합에서 빈번하게 나타나는 패턴이 존재할 때 그 패턴의 서브패턴 또한 빈번하게 나타난다는 원리를 이용하여 빈발 패턴을 마이닝 한다. 또한 이 원리를 반대로 생각하여 빈번하게 나타나지 않는 패턴을 가지는 슈퍼패턴은 빈발 패턴이 아니다 라는 원리를 이용하여 연산량을 감소시킨다. 빈발 패턴 트리 알고리즘[7,11]은 Aprioir의 그래프 스트럭처의 데이터들을 트리형태로 모델링한 알고리즘으로 Apriori 알고리즘과 다르게 깊이 우선 탐색의 방식으로 동작하고 보다 빠르게 동작하고 알고리즘으로 알려져 있다. 이는 Apriori 알고리즘에서 서브패턴들의 빈도를 계산할 때 계속해서 이루어지는 전체 데이터에 대한 스캔연산을 한번만 수행하기 때문인데 트리를 생성하는 과정에서 서브 패턴을 고려하지 않고 한번의 입력으로 빈발 패턴을 찾을 수 있기 때문이다.

2.2 클러스터링

클러스터링 알고리즘은 데이터 마이닝 또는 머신러닝 분야에서 데이터 분류를 위한 도구 중 비지도 학습방법에 속한 모든 알고리즘을 총칭하는 용어이다. 입력되는 데이터를 기반으로 유사한 데이터의 서브셋을 찾는 데이터 분석을 위한 가장 기초적이면서도 중요한 부분을 차지하고 있으며 단순 혹은 다중선형회귀법, K-means, DBSCAN등의 알고리즘들[8]이 존재한다. 크게 분활적 클러스터링과 계층적 클러스터링으로 나눌수 있다. 분할적 클러스터링은 데이터 분류에 대해 최적화시키는 K개의 분할 영역을 결정해 나가는 방법으로 유클리드 거리등의 측정법에 기반한다. K-mean은 분할적 클러스터링의 대표적 알고리즘으로 원하는 서브셋의 수인 K를 입력으로 데이터 거리의 평균이 가장 작은 k의 분할을 찾을 때 까지 반복적으로 동작하는 알고리즘이다. 계층적 클러스터링은 초기 각 데이터를 하나의 클러스터로 설정한 후 다른 데이터와의 거리를 기반으로 분할 및 합병해 나가는 상향식방식을 뜻한다. 모든 점들이 하나의 대형 클러스터에 속하게 될 때까지 그 히스토리 정보를 유지해 나가게 되고 이것은 agglomerative hierarchical clustering 이라 하며 유클리드 거리를 기반으로 가까운 데이터끼리 클러스터링 하는 방법을 나타낸다. 클러스터링은 패턴인식, 영상처리 등의 응용분야에서도 데이터 마이닝에서의 만큼 높은 중요도를 가지는 알고리즘 중 하나이다.

2.3 맵리듀스

맵 리듀스[9]는 구글이 제시한 분산 처리 환경을 위한 프로그래밍 모델로서 좀 더 심플하고 간결하게 병렬 프로그래밍을 할 수 있도록 지원하는 프레임워크이다. 크게 맵과 리듀스로 나누어져 있으며 각 수행 주체를 매퍼와 리듀서라고 부른다. 수행동작을 살펴보면 먼저 매퍼는 DBMS나 HDFS같은 저장소에 저장되어 있는 데이터를 라인 대 라인방식으로 읽어온다. 이 후 데이터를 분석하여 키와 값을 만드는데 이때 만들어진 키는 처리될 리

듀서를 구분하는 고유값이 된다. 즉 리듀서는 매퍼에서 생성된 키의 개수와 같은 개수로 만들어지고 각 키에 해당하는 값 집합을 입력받는다. 실제 동작에서는 매퍼와 리듀서 사이 컴바인같은 동작으로 인하여 정렬 등의 작업을 추가할 수있다. 리듀서는 결과값에 대한 키와 값을 출력하므로써 맵리듀스의 전체 작업을 마무리한다. 현재 맵리듀스는 아파치의 분산처리 솔류션인 하둡[10]의 메인 프로그래밍 모델로 자리잡았다. 하둡을 지원하는 여러 패키지를 통해 자바, C++, 파이썬, 스칼라 등의 프로그래밍 언어를 사용하여 구현할 수 도있고 최근 스파크 같은 인메모리 기반 처리 시스템에서의 맵리듀스 또한 고속 처리를 위한 주요 수단으로 많은 주목을 받고 있다.

3. 결정 빈발 패턴 트리

빈발 패턴 트리는 2.1과 같이 데이터 집합에서 빈번하게 나타나는 패턴을 찾는 데이터 마이닝 기법 중 하나이다. 하지만 기존의 빈발 패턴 트리는 새로운 데이터에 대한 트리 확장 과정에서 다른 서브트리의 데이터집합은 고려하지 않고 오직 현재 입력된 데이터의 트랜젝션 값 하나와 트리에 나타나는 트랜젝션 값 하나만을 비교하여 트리의 확장 여부를 판단한다. 이는 패턴에 대한 마이닝은 정확 할 수 있지만 패턴의 형태가 다양하게 나타나는 소셜 데이터의 경우 데이터의 크기가 커지면 커질수록 과도한 수의 패턴이 생성되어 패턴 결과를 분석할 수 없거나 혹은 분석 시 발생하는 연산에 대한 오버헤드가 크다는 단점을 가지고 있다. 본 논문에서는 빈발 패턴 트리의 생성 과정에 각 서브트리에 나타나는 패턴들과 새로 입력된 데이터의 패턴에 대한 비교를 통해 트리의 확장 여부를 판단해 나가는 결정 빈발 패턴 트리를 제시한다. 이는 수렴 여부를 판단하는 과정이 결정트리의 재귀적 특성과 닮아 붙혀진 이름이다. 기존 패턴으로의 수렴 혹은 새로운 패턴생성 여부는 새로 입력된 데이터의 패턴과 기존의 패턴들과 유사도를 측정하여 판단된다. 이러한 동작방식은 패턴을 찾는 것 뿐만 아니라 클러스터링에 대한 결과도 보임으로서 보다 많은 응용에 적용할 수 있는 장점이있다. 또한 본 논문에서는 단일 머신이 아닌 대용량 데이터 혹은 분산 처리를 위한 맵리듀스 프로그래밍 모델을 적하여 고속 처리가 가능한 분산에서의 맵리듀스 기반의 결정 빈발 패턴 트리를 보인다.

3.1 클러스터링

기존의 빈발 패턴 트리와 다르게 결정 빈발 패턴 트리는 트리 생성과정에서 빈발 패턴 생성과 더불어 클러스터링을 함께 수행한다. 이는 데이터의 각 아이템 개별 요소들을 기준으로 패턴 생성 여부를 판단하는 것이 아니라 요소의 집합, 즉 데이터가 가지는 아이템의 리스트 전체 기반으로 패턴 생성 여부를 판단하고 다른 패턴들과 비교하기 때문이다. 리스트에 대한 다차원 데이터를 적용시키기 위해 코사인 유사도 공식을 사용하여 비교를 해나간다. 여기서는 이러한 동작방식을 설명하기 위해 임의의 데이터[7]에 대한 결정 빈발 패턴 트리 생성 과정을 보인다. 초기 데이터에 대해 스켄이 이루어지고 아이템의 빈발 요소등을 계산하는 작업은 기존의 빈발 트리와 똑같이 이루어진다.

(표 1) 아이템 트랜젝션 데이터
(Table 1) Item Transaction Data

TID	list
1	1(4), 2(2), 5(5)
2	2(5), 4(5)
3	2(4), 3(4)
4	1(4), 2(4), 4(5)
5	1(5), 3(3)
6	2(2), 3(1)
7	1(5), 3(4)
8	1(5), 2(3), 3(2), 5(5)
9	1(4), 2(3), 3(3)

표 1은 아이템 리스트를 가지고 있는 데이터 집합을 보여준다. 기존의 빈발 패턴 트리와 유사하게 데이터 스켄을 통해 초기 데이터 리스트에 나타나는 아이템의 빈도수를 계산한다. 표 2 는 각 아이템의 빈도수가 계산되어 내림차순으로 정렬된 해더 테이블을 나타낸다.

(표 2) 헤더 테이블
(Table 2) Header Table

Item	Frequency
2	7
1	6
3	6
4	2
5	2

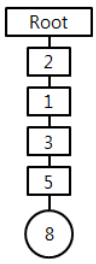
이 후 minsupport를 적용하여 프루닝을 진행한다. 이는 기존의 빈발 패턴 트리와 같이 빈번하게 나타나지 않는 아이템에 대해 패턴을 생성하지 연산의 연산량 감소와 함께 패턴의 정확성을 지키도록 한다. minsupport는 빈발

패턴 트리 뿐만아니라 다른 마이닝 알고리즘에서도 중요한 요소 중 하나인데 실제 minsupport의 계수가 높으면 프루닝에 대한 연산량의 감소와 함께 빈번하게 나타나는 패턴을 정확하게 찾는 역할을 한다. 하지만 특정 목표에 따라 희소 패턴을 찾거나 모든 데이터에 대한 정확한 패턴을 얻고 싶을 경우 minsupport의 계수 설정은 보다 엄격하게 진행되어야 한다. 최근에는 다중 minsupport[1]를 지원하는 프래임워크를 적용하여 유연하게 동작하는 알고리즘들이 많이 구현되고 있다. 다음에 나타나는 표 3은 표 1에 각 아이템별 빈도순을 내림차순으로 정렬하고 minsupport를 1로 설정한 결과를 아이템 개수에 따라 내림차순 한 결과를 나타낸다. 실질적으로 1이하의 빈발도를 갖는 데이터는 없으므로 어떤 덴디터도 증명도지 않는 결과를 보인다.

(표 3) 빈도수를 기준으로 정렬된 아이템 트랜젝션 데이터
(Table 3) Item Transaction Frequency Data Sorted by Frequency

TID	list
8	2(3), 1(5), 3(2), 5(5)
1	2(2), 1(4), 5(5)
4	2(4), 1(4), 4(5)
9	2(3), 1(4), 3(3)
2	2(5), 4(5)
3	2(4), 3(4)
5	1(5), 3(3)
6	2(2), 3(1)
7	1(5), 3(4)

표 3의 데이터를 완성으로 모든 전처리 단계가 끝났다. 이제 부터는 패턴을 생성하는 과정을 설명한다. 표 3의 데이터를 순차적으로 입력하여 트리를 생성하는데 다음은 초기 8의 TID를 입력한 형태의 트리이다.

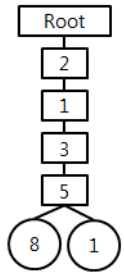


(그림 1) TID=8이 입력된 초기 트리
(Figure 1) Initial Tree that included item of TID 8

네모는 아이템에 대한 패턴을 나타내고 원은 TID에 대한 클러스터를 나타낸다. 이후 TID=1의 데이터가 입력되는데 기존 빈발 트리와 다르게 유사도 계산을 통하여 패턴 수렴여부를 계산한다. 하지만 이때 유사도 계산 보다 먼저 포함 여부를 계산한다. 그림 1을 보면 2,1,3,5로 이루어진 패턴 하나가 존재한다. 그리고 새로 입력되는 TID=1의 데이터는 2,1,5의 아이템을 가지고 있는데 이는 다음과 같은 공식에 True의 결과를 나타낸다.

$$\sum_{i=1}^{n} f(X, T_i) = \begin{cases} 1, \text{if } X \subseteq T_i \\ 0, otherwise \end{cases}$$

이러한 공식에 새로입력된 패턴은 기존의 패턴에 수렵하고 다음과 같은 트리가 만들어 진다.



(그림 2) TID=8,1이 입력된 트리
(Figure 2) Tree that included item of TID 8,1

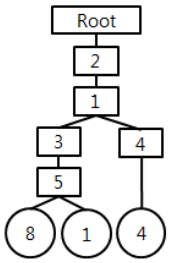
이는 TID=1과8이 하나의 클러스터로 그룹화 되었다는 의미로 2,1,3,5와 2,1,5의 비교이후 2를 제외한 1,3,5와 1,5의 비교, 3,5와 5의 비교를 통해 이루어진다. 다음은 이러한 비교를 위한 코사인 유사도 공식이다.

$$similarity(A,B) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

위의 식을 이용하여 새로입력되는 데이터와 기존의 패턴들 사이의 유사도를 계산하여 수렴 여부를 결정한다. 위 트리에서 보면 초기 2,1,3,5는 2,1,5를 가지므로써 기존의 패턴에 수렴하게 된다. 이후 해당노드인 2를 제거하고 1,3,5와 1,5를 비교하는데 위 공식을 대입하면 다음과 같은 값을 가진다.

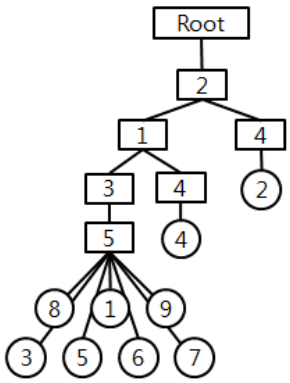
$$Set\ A = 1(5), 3(2), 5(5)$$
$$Set\ B = 1(4), 5(5)$$
$$similarity(A,B) = \frac{45}{7.34 \times 6.4} = 0.957$$

즉 유사도가 0.957로 거의 비슷하다는 의미로 해석할 수 있다. 경우에 따라 이러한 해석을 다르게 할 수 있지만 본 논문서에 제시하는 결정 빈발 패턴 트리를 위해 측정하는 유사도의 경우 0.6 ~ 0.7 이상으로 나타날 시 신뢰할 수 있는 클러스터링 결과를 보인다. 다음은 이후 TID=4의 데이터를 입력한 결과를 나타낸다.



(그림 3) TID=8,1,4이 입력된 트리
(Figure 3) Tree that included item of TID 8,1,4

TID=8,1과 4는 다른 클러스터로 그룹화 되었다. 여기서 중요한 점은 2와 1을 가진 데이터 집합에 대해 3,5 그리고 4를 추천할 수 있는 기반이 마련되었다는 것이다. 실제 클러스터링에서 차원축소를 통해 진행하면 이러한 데이터에 대한 유지를 할 수없게 되는데 빈발 패턴 트리의 경우 데이터 손실 없이 트리 생성 및 데이터 복구가 가능해 클러스터링을 위해 보다 유용한 모델을 생성하기 적합하다. 아래의 그림 4는 모든 데이터에 대해 클러스터링을 수행한 결과에 대한 그림를 나타낸다.



(그림 4) 완성된 결정 빈발 패턴 트리
(Figure 4) Decision Frequent Pattern Tree that are built completely

3.2 구현

본 절은 3.1에서 설명한 결정 빈발 패턴 트리의 생성 프로그래밍 모델에 대해 설명한다. 먼저 유사도 계산에 대한 연산의 의사코드를 나타낸다. 병합 정렬의 분할과 정복 동작방식과 거의 유사하다.

```
function similarity(inputA[],inputB[])
setFirst(inputA)
setFirst(inputB)
while next(inputA) != null &&next(inputB) != null:
    if id(inputA) == id(inputB):
        up += rate(inputA) * rate(inputB)
        bottomA += power(rate(inputA))
        bottomB += power(rate(inputB))
        doNext(inputA)
        doNext(inputB)
    else if id(inputA) < id(nextB):
        bottomA += power(rate(inputA))
        doNext(inputA)
    else:
        bottomB += power(rate(inputB))
        doNext(inputB)
```

```
while next(inputA) != null:
        bottomA += power(rate(inputA))
        doNext(inputA)
while next(inputB) != null:
        bottomB += power(rate(inputB))
        doNext(inputB)
return (up/(sqrt(bottomA)*sqrt(bottomB)))
```

3.1의 데이터와 같이 각 아이템에 대한 가중치를 가지고 있어 같은 아이디의 아이템이 존재하면 아이템 아이디가 아니라 해당 가중치를 연산하므로 유사도를 구할 수 있다. 이는 분자에서 $O(N)$와 분모에서 $O(N)$의 성능을 보여 정확하게는 $O(2N)$이지만 $O(N)$로 수렴 시킬 수 있다. 하지만 모든 대용량 데이터집합에 대해 이러한 연산이 반복적으로 수행되면 굉장히 많은 데이터의 입출력이 발생하게 되는데 최근 인메모리기반의 고속 처리 가능한 프레임워크들로 이러한 데이터 입출력에 대한 부담을 줄이고 성능을 개선해 나가는 연구가 많이 이루어지고 있고[1] 특히 빅데이터 분야에서 가장 주목받는 기술이기도 하다. 다음은 실제 결정 빈발 패턴 트리생성에 대한 의사코드를 보여준다. 결정 빈발 패턴 트리는 사이클이 없는 트리형태이므로 보다 쉽게 반복적 연산을 사용하여 구현할 수 있다.

```
function makeDFPTree(dataSet[],tetha)
root = 0
setFirst(dataSet)
while next(dataSet) != null:
    data = dataSet
    while pattern in root:
        if pattern include data:
            input(pattern,data)
        else if similarity(data,pattern) > tetha:
            input(pattern,data)
    if not processing:
        input(root,pattern)
    doNext(dataSet)
return root
```

```
function input(dataA[],dataB[],tetha)
    while next(dataA) != null&&next(dataB) != null:
    if similarity(set(dataA),set(dataB)) > tetha:
        if id(dataA)==id(dataB):
            doNext(dataA)
            doNext(dataB)
        else:
            break
concatnation(dataA,dataB)
```

3.3 맵리듀스 기반의 모델링

결정 FP-Tree를 분산시스템에 적용하기 위하여 맵리듀스로 모델링한다. 맵리듀스는 분산환경에서의 프로그래밍을 좀더 심플하게 할 수 있도록 도와주는 도구로서 프로그래밍에 대한 코드적 관점 뿐만 아니라 여러 가지 부가 기능이 포함되어있다. 먼저 빈도수 계산을 위해 트랜잭션 데이터 집합을 맵으로 입력한다. 이는 각 독립된 요소를 연산, 즉 여기서는 카운팅을 통해 독립된 아이템에 대한 증가연산을 사용하는데 이미 이에 해당하는 많은 논문과 관련 자료가 인터넷에 존재한다.

이 후 데이터를 정렬하고 트리 생성을 준비한다. 정렬의 경우 맵리듀스의 컴바인을 통해 동작 시킬수 있으므로 부가적인 동작방식의 이해가 필요없다. 맵리듀스를 이용한 기존의 빈발 패턴 트리 생성은 여러가지 방법으로 구현되어 왔지만[11][12] 본 논문에서 제시하는 결정 빈발 패턴 트리의 경우 맵의 입력으로 빈발도를 키, 벨류를 각 아이템으로, 출력을 각 아이템에 대한 하위 노드로 모델링한다. 이후 아이덴티티 리듀서를 사용하고 맵리듀스에 대한 반복 호출을 통해 수행을 완성한다.

4. 성능 분석

결정 빈발 패턴 트리의 성능평가를 위하여 아마존의 메타데이터를 이용하여 클러스터링을 수행한다. 실험은 vCore 4, Memory 4GB기반의 단일 머신에서 이루어졌으며 맵리듀스 기반의 클러스터링을 수행하여 클러스터의 개수 및 알고리즘 완료 시간을 측정하여 이루어졌다. 실험에서 사용된 아마존의 메타데이터[16]는 스탠포드대학의 SNAP 그룹[15]에서 제공하는 것으로 아마존에서 판매하는 상품 아이디, 그룹, 카테고리, 유사한 아이템, 구매한 사용자로 구성되어 있으며 본 실험에서는 사용자에 대해 구매 아이템을 기반으로 클러스터링을 수행하여 전체 사용자 집합에서 유사한 사용자를 분류한다. 여기서 나타나는 상품에 대한 아이디는 아마존에서 아이템에 부여하는 고유 번호인 ASID를 나타낸다.

4.1 데이터 전처리

먼저 실험을 위하여 아이템 기반으로 저장되어있는 아마존 메타데이터에 대해 전처리를 진행하여 사용자 기반의 데이터로 변환한다. 상품 아이디를 기준으로 저장된 데이터를 맵리듀스를 통해 사용자 ID와 해당 상품에 대한 평점을 기준으로 구성된 사용자 데이터셋을 생성한다. 카테고리, 판매순위 등의 데이터는 본 실험과 무관함과 동시에 불필요한 데이터를 제거함으로써 데이터 크기 감소로 인한 성능 증가를 도출하였다.

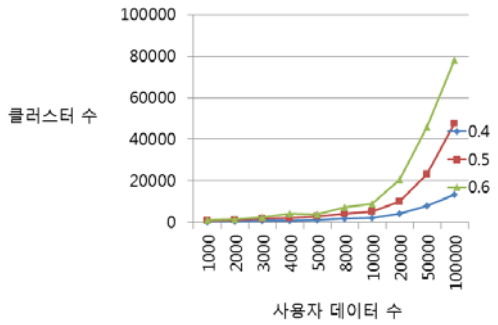
```
Id:   15
ASIN: 1559362022
  title: Wake Up and Smell the Coffee
  group: Book
  salesrank: 518927
  similar: 5  1559360968  1559361247  1559360828  1559361018  0743214552
  categories: 3
   |Books[283155]|Subjects[1000]|Literature & Fiction[17]|Drama[2159]|United States[216
   |Books[283155]|Subjects[1000]|Arts & Photography[1]|Performing Arts[521000]|Theater[
   |Books[283155]|Subjects[1000]|Literature & Fiction[17]|Authors, A-Z[70021]|( B )[700
  reviews: total: 8  downloaded: 8  avg rating: 4
    2002-5-13  cutomer: A2IGOA66V6O8TQ  rating: 5  votes:   3  helpful:   2
    2002-6-17  cutomer: A2OIN4AUH84KNE  rating: 5  votes:   2  helpful:   1
    2003-1-2  cutomer: A2HN382JNT1CIU  rating: 1  votes:   6  helpful:   1
    2003-6-7  cutomer: A2FDJ79LDU4O18  rating: 4  votes:   1  helpful:   1
    2003-6-27  cutomer: A39QMV9ZKRJXO5  rating: 4  votes:   1  helpful:   1
    2004-2-17  cutomer:  AUUVMSTQ1TXDI  rating: 1  votes:   2  helpful:   0
    2004-2-24  cutomer: A2C5K0QTLL9UAT  rating: 5  votes:   2  helpful:   2
    2004-10-13  cutomer:  A5XYF0Z3UH4HB  rating: 5  votes:   1  helpful:   1
```

(그림 5) 아마존 메타 데이터의 구성요소
(Figure 5) Component of Amazons' Meta Data

4.2 클러스터링 성능

실제 성능 평가는 클러스터의 수로 평가하였다. 임계도를 0.4,0.5,0.6으로 두고 사용자 수에 따른 클러스터의

수는 데이터의 밀집도를 나타낸다. 실제 k-means 알고리즘과 비교하여 맵리듀스가 수행시간에서 우수하다는 결론은 많은 논문[13][14]에서 제시되었는데 여기서는 밀집도에 대한 정도를 계산하여 성능을 평가하였다.



(그림 6) 데이터 크기에 따른 클러스터 개수
(Figure 6)The Number of Cluster as Number of Data size

5. 결 론

본 논문에서는 빈발 패턴 트리를 이용하여 클러스터링을 수행하는 결정 빈발 패턴 트리 알고리즘을 보였다. 기존의 빈발 패턴 트리의 패턴 생성과정에서 발생하는 트리의 많은 패턴 생성으로 인하여 클러스터링 및 데이터 분석 시 발생하는 오버헤드를 최소화 하기 위해 유사도 계산 기반으로 서브트리에 대한 수렴 및 확장 여부를 판단한다. 또한 이를 맵리듀스로 모델링하여 보다 고속처리 가능한 모델을 선보이고 다른 클러스터링 알고리즘과 성능을 비교하였다.

현재 많은 기업 혹은 기관에서 소셜 데이터에 대한 클러스터링은 굉장히 중요한 요소로서 실제 클러스터링 기반의 추천 시스템은 특정 기업들의 매출에 상당한 영향을 미친다. 또한 단순 서비스를 넘어 여러 가지 데이터 분석을 기반으로 범죄를 예측, 질병 진단 등의 중요한 문제에 대한 서비스를 아이티 기반의 기술로 수행함으로써 데이터 분석의 정확도 개선으로 신뢰도를 향상시키는 것은 매우 중요한 일이 되었다. 하지만 분석의 정확도나 성능을 개선하는 것은 쉽지않은 연구이고 실시간으로 이러한 기능을 사용자에게 지원한다는 것은 더욱 어려운 일이다. 본 논문은 보다 빠르고 정확한 알고리즘으로 사용자의 요구에 맞게 데이터 분석을 하는 기반 마련에 중점을 두었다. 향후 연구로는 트리형태로 모델링하는 알고리즘 뿐만 아니라 다양항 데이터 구조, 즉 그래프 혹은 다차원의 데이터 구조에 대한 빠른 클러스터링 및 알고리즘을 위하여 컴퓨터 사이언스의 다양한 알고리즘을 클러스터링 같은 기존 마이닝 문제에 적용시키고 맵리듀스로 모델링하여 고속처리 가능한 알고리즘을 개발에 대한 연구가 필요하다.

참 고 문 헌 (Reference)

[1] Y.Lim "IT's Evolution Scenario as Machine Learning" http://www.zdnet.co.kr/news/news_view.asp?artice_id=20141212161631

[2] A.Das M.Datar and A.Garg "Google News Personalization: Scalable Online Collaborative Filtering" University of Illinois at Urbana Champaign http://www2007.org/papers/paper570.pdf

[3] S.Kim "A Accuracy of Deepspace's Picture Taging System are 97%" http://biz.chosun.com/site/data/html_dir/2014/03/21/2014032103146.html

[4] Wikipedia "Bigdata" ttps://en.wikipedia.org/wiki/ Big_data

[5] Zoubin Ghahramani "Unsupervised Learning" "http://mlg.eng.cam.ac.uk/zoubin/papers/ul.pdf"

[6] Wikipedia "Apriori_algorithm" https://en.wikipedia.org/wiki/Apriori_algorithm

[7] G.Lee and U.Yun, "Analysis and Performance Evaluation of Pattern Condensing Techniques used in Representative Pattern Mining" Journal of Internet Computing and Services, Vol.16, No.2, pp.77-83, 2015

[8] K.Lee, H,Namgoong, E.Kim, K.Lee and H.Kim "Analysis of multi-demensional interaction among SNS users" Journal of Korean Society for Internet Information, Vol.12, No.2, pp.113-121, 2011

[9] Jeffrey Dean and Sanjay Ghemawat "MapReduce: Simplified Data Processing on Large Clusters" Google,Inc.http://static.googleusercontent.com/media/research.google.com/ko//archive/mapreduce-osdi04.pdf

[10] K.Shvachko, H.Kuang, S.Radia and R.Chansler "The Hadoop Distributed File System" Yahoo! Sunnyvale, California USA IEEE 978-1-4244-7153-9 2010

http://zoo.cs.yale.edu/classes/cs422/2014fa/readings/papers/shvachko10hdfs.pdf
[11] D.Cho, K.Chung, K.Rim and J.Lee "Method of Associative Group Using FP-Tree in Personalized Recommendation System" Journal of Korea Contents Association Vol.7 No.10, pp.19-26, 2007
[12] B.Jeong and A.Farhan "Efficient Dynamic Weighted Frequent Pattern Mining by using a Prefix-Tree" Journal of Information Processing Systems D Vol.17-D No.4 pp.253-258 2010
[13] G.Lee, U.Yun, D.Kim, G.Ryang, J.Hwang, B.Yang and C.Jeong "Performence Evaluation and Analysis of Various Techniques on Graph Pattern Mining" Journal of Korean Society for Internet Information, Vol.16, No.1, pp.77-78, 2015
[14] E.Jeong and B.Lee "A strategy of emotional information classification for SNS using Support Vector Machine" ournal of Korean Society for Internet Information, Vol.16, No.1, pp.261-262, 2015
[15] Stanford SNAP Group http://snap.stanford.edu/
[16] Amazon Meta data represented by Stanford SNAP Group http://snap.stanford.edu/data/amazon-meta.html

◐ 저 자 소 개 ◑

서 영 원 (Young-won Seo)
2013~현재 부경대학교 IT융합응용공학과(공학사)
관심분야 : 데이터 마이닝, 기계학습, 빅데이터 등
E-mail : jazz9008@gmail.com

김 창 수 (Chang-soo Kim)
1991년 중앙대학교 컴퓨터공학과 박사
2002년~2003년 미국 UMKC 방문교수
2006년~현재 유비쿼터스 부산 도시협회 방재분과위원장
2013년~현재 미국 콜로라도대학 방문교수
2013년~현재 한국멀티미디어학회 이사
2013년~현재 한국인터넷정보학회 이사
2011년~현재 한국멀티미디어학회 정책자문위원
1992년~현재 부경대학교 IT융합응용공학과 교수
관심분야 : 방재IT, UIS/GIS, 운영체제, 재난관리, 공간 검색, 도시방재 등
E-mail : cskim@pknu.ac.kr