https://doi.org/10.14400/JDC.2017.15.6.229

RHadoop 플랫폼기반 CAWFP-Tree를 이용한 적응 빈발 패턴 알고리즘

박인규
중부대학교 게임소프트웨어학과

Adaptive Frequent Pattern Algorithm using CAWFP-Tree based on RHadoop Platform

In-Kyu Park
Dept. of Game Software, College of Engineering Joongbu University

요 약 효율적인 빈발 패턴 알고리즘은 연관 규칙 마이닝이나 융복합을 위한 마이닝 과정에서 필수적인 요소이며 많은 활용성을 가지고 있다. 패턴 마이닝을 위한 많은 모델들이 빈발 패턴에 관한 정보를 추출하여 FP-트리를 이용하여 저장하고 있다. 본 논문에서는 항목들의 무게중심을 이용한 새로운 빈발 패턴 알고리즘(CAWFP-Growth)을 제안하여 항목들이 가지는 가중치와 빈도수를 같이 고려하여 항목간의 중심을 계산하여 기존의 FP-Growth 알고리즘의 효율성을 향상시킨다. 제안한 방법은 하향 폐쇄의 성질을 유지하기 위한 기존의 전역적 최대치 가중치 지지도를 필요로 하지 않기 때문에 자연히 빈발 패턴의 탐색시간이 줄어들고 정보의 손실을 줄일 수 있다. 실험결과를 통하여 제안된 알고리즘이 기존의 동적 가중치를 이용하는 다른 방법과 비교해볼 때, 항목들의 무게중심이 빈발패턴의 정확한 정보를 유지하고 FP-트리의 처리시간을 줄여주기 때문에 제안한 방법의 중요성을 보이고 있다 또한 가상 분산모드에서 맵리듀스 프레임을 기반으로 빅데이터를 모델링하고 향후 완전분산 모드에서 제안한 알고리즘의 모델링이 필요하다.

주제어 : 데이터 마이닝, 가중치 빈발패턴, 무게중심, 하향폐쇄, 맵리듀스

Abstract An efficient frequent pattern algorithm is essential for mining association rules as well as many other mining tasks for convergence with its application spread over a very broad spectrum. Models for mining pattern have been proposed using a FP-tree for storing compressed information about frequent patterns. In this paper, we propose a centroid frequent pattern growth algorithm which we called “CAWFP-Growth” that enhances he FP-Growth algorithm by making the center of weights and frequencies for the itemsets. Because the conventional constraint of maximum weighted support is not necessary to maintain the downward closure property, it is more likely to reduce the search time and the information loss of the frequent patterns. The experimental results show that the proposed algorithm achieves better performance than other algorithms without scarifying the accuracy and increasing the processing time via the centroid of the items. The MapReduce framework model is provided to handle large amounts of data via a pseudo-distributed computing environment. In addition, the modeling of the proposed algorithm is required in the fully distributed mode.

Key Words : Data Mining, Weight Frequent Pattern, Centroid, Downward Closure, MapReduce

“이 논문은 2016년도 중부대학교 학술연구비 지원에 의하여 이루어진 것임“
Received 1 May 2017, Revised 31 May 2017
Accepted 20 June 2017, Published 28 June 2017
Corresponding Author: In-Kyu Park(Dept. of Game Software, College of Engineering, Joongbu University)
Email: fip2441g@gmail.com

ISSN: 1738-1916

1. 서론

최근에 IT 기술의 발전과 더불어 생성되거나 저장되는 기하급수적인 빅 데이터(BigData)를 분석함으로써 새로운 사실이나 의미(Significance)를 발견하려는 빅 데이터 마이닝 기술이 중요해지고 있다. 데이터 마이닝 기법들 중 빈발 패턴 마이닝은 데이터베이스의 트랜잭션(Transaction) 등의 데이터에서 빈번하게 발생하는 아이템들의 집합을 찾아내는 마이닝 기법으로 특히, 연관규칙(Association Rules), 폐쇄 패턴(Closed Pattern), 함수의존도(Functional Dependencies) 등등에서 필수적인 부분이다[1].

초기의 Apriori 알고리즘은 불필요한 후보패턴의 생성과정에서 패턴의 양이 증가하여 FP-트리를 이용한 FP-Growth 알고리즘을 활용하여 이러한 문제점을 극복하여 성능면에서 많은 개선을 가져오게 되었다. 이러한 방법들은 각 항목의 중요도를 모두 같은 값으로 고려한 방법들이기 때문에 항목간의 중요도가 서로 상이한 환경에서는 적용이 어려워진다. 따라서 빈발항목의 추출에서 중요도의 시변성(Time-Variance)에 관한 많은 연구가 진행되어 왔다. 일반적으로 임의의 항목이 가지는 의미는 시간에 따라서 변하기 마련이다[2]. 실제적으로 많은 사례에서 보면 항목이 가지는 의미는 시간에 종속적인 경우가 존재한다. 가령 소매시장에서 값 비싼 항목은 자주 거래가 이루어지지 않지만 전체적인 매출에서 보면 커다란 비중을 차지하고 있다. 또한 이러한 방법들은 하향 폐쇄의 성질을 만족하기 이해서는 전역임계값이나 국부적임계값을 최대치 임계값으로 설정하고 있다. 이로 인해 항목간의 가중치를 고려할 경우에 정보의 손실이 발생할 수 있다. 그리고 기존의 적응 빈발 패턴 마이닝에서는 항목의 가중치만을 고려하고 빈도수는 고려하지 않고 있다.

본 논문에서는 임계값의 최대치를 설정하지 않고 항목의 가중치와 빈도수를 같이 항목간의 정확한 관계를 설정할 뿐만 아니라, 하향 폐쇄의 성질을 유지할 수 있는 새로운 중심 기반 적응 가중치 빈발 패턴 (CAWFP: Centroid based Adaptive Weight Frequent Pattern) 알고리즘을 제안한다. 기존의 동적 가중치 빈발 패턴(WFP: Weighted Pattern Mining) 에서와 같이 시간적으로 변하는 중요도를 고려할 수 있고 단 한 번의 데이터베이스 스캔을 통하여 처리되므로 스트림 데이터 환경과 같은 실시간 처리가 필요한 환경에서 적용이 가능하다.

2. 관련연구

2.1 빈발 패턴 마이닝

$I=\{i_1, i_2, ..., i_m\}$는 항목들의 집합이고 $T=\{T_1, T_2, ..., T_n\}$는 트랜잭션들의 집합으로 구성된 데이터베이스는 D라고 한다. 트랜잭션 $T_i \in T$ 는 I의 부분집합으로 구성된다. 항목집합 X 가 $X \subset T$ 일 경우, “트랜잭션 T 가 아이템 집합 X 를 포함 한다”라고 한다. D에 속하는 s퍼센트의 트랜잭션들이 아이템 집합 X를 포함한다면 우리는 X의 지지도를 s라고 한다. 또한 우리는 미리 정의된 최소 지지도 (Minimum Support)이상을 갖는 항목집합을 빈발 항목집합이라 한다. 그리고 k개의 항목들로 이루어진 빈발 항목집합을 빈발 k-항목 집합 (Frequent k-Itemset)이라 한다. 빈발 패턴 마이닝 문제는 사용자가 지정한 최소 지지도 이상의 지지도를 갖는 모든 빈발 아이템 집합들을 구하는 것이다. 빈발 패턴 알고리즘으로 초기에 Apriori 알고리즘이 제안되었으나 불필요한 후보 패턴의 발생으로 인하여 잘 활용되지 못하였다. 그 후 FP_트리를 이용한 FP-Growth 알고리즘은 이러한 문제를 극복하여 많은 성능적인 개선을 가져오게 되었다.

2.2 가중치 빈발 패턴 마이닝

MINWAL에서는 하향 폐쇄 성질(Download Closure Property)을 만족하기 위하여 k-지지도를 유지하고 있지만, 빈발 항목의 정확도에서 효율성이 떨어지고 Apriori 알고리즘을 기반으로 하고 있기 때문에 k-지지도를 유지하기 위한 시간 손실이 크다[3]. WARM에서는 지지도와 하향 폐쇄에 각각 가중치를 적용하여 하향폐쇄가 깨지는 문제를 해결하였다[4]. 그러나 WARM에서 “ab” 항목의 가중치 지지도는 모든 트랜잭션의 가중치에 대한 “a” 와 “b”를 포함하는 트랜잭션의 가중치의 비율로써 Apriori 알고리즘을 기반으로 하고 있다. WAR에서는 가중치를 고려하지 않고 빈발 패턴을 발생하고 연관 규칙을 과정에서 가중치를 고려하는 방법이다[5]. 역시 Apriori 알고리즘을 기반으로 하고 있다. Apriori 알고리즘을 기반으로 하고 있는 알고리즘들은 후보 패턴을 발생하고 조사

하는 방식으로 이루어지기 때문에 데이터베이스를 여러 번 스캔해야 하는 단점을 가지고 있다. WFIM에서는 FP-트리 기반의 초기의 가중치 빈발 패턴 마이닝은 WFIM이다. 이는 두 번의 데이터베이스 스캔을 필요로 하고 최소 가중치와 가중치의 범위를 가진다[6]. 항목은 고정된 가중치를 가지고 있다. WIP에서는 가중치 신뢰도(Weight Confidence)를 정의하고 가중치의 범위를 이용하여 가중치의 그룹을 결정하고 h-신뢰도를 이용하여 우수한 지지도를 가지는 패턴을 확보한다[7]. WLPMiner에서는 가중치와 지지도에 대해 제약을 두어서 소수의 우수한 빈발 패턴을 확보하게 되었다[8]. 지금까지의 모든 가중치 빈발 마이닝 알고리즘은 항목들의 가중치가 고정된 값을 가지는 경우를 다루었고 항목들의 가중치 값이 시간에 따라 변하는 환경도 고려되었다. 여러 가지 알고리즘을 통하여 알 수 있는 것은 빈발 패턴 마이닝에서 적용하던 하향 폐쇄의 성질을 가중치 빈도수에는 적용할 수 없는 것이 중요한 문제점으로 지적되어 왔다[9]. 이러한 일환으로 본 논문에서는 무게중심(Centroid)을 이용하여 가중치 빈도수를 고려하는 방법을 제안한다.

2.3 맵리듀스 프레임워크

맵리듀스의 실행은 클러스터내의 노드간의 데이터 전달에 따라서 맵(Map)과 리듀스(Reduce)의 두 단계로 구성되어지며 모두 키-값 쌍 (Key-Value Pair)의 형태로 동작하고 있다. 맵 단계의 각 맵퍼 (Mapper)는 하나의 스플릿 (Split)을 입력으로 받고, 사용자가 정의한 맵 함수를 수행한 후에 키-값 쌍으로 이루어진 맵출력이 노드에 저장된다. 이 출력 데이터들은 키의 순서대로 정렬/분류되어 키에 따라 특정 리듀서로 전송된다. 리듀스 단계에서는 각 키에 대해 값들의 리스트로 구성하여 사용자가 정의한 리듀스 함수를 수행한다.

프레임워크의 구성은 하나의 마스터 잡 트래커와 복수의 태스크 트래커(Task Tracker)로 되어 있으며 클러스터의 각 노드는 태스크 트래커로 동작한다. 마스터는 입력 데이터의 분할, 태스크 스케쥴링(Scheduling), 태스크간의 통신과 모니터링을 담당하며 태스크의 실행은 슬레이브(Slave)가 담당한다. 맵리듀스는 아파치 분산처리 솔루션인 하둡(Hadoop)의 기본모델로 빅데이터를 위한 여러 가지의 확장성을 가지고 있다[10].

3. 제안된 적응가중치 빈발패턴 마이닝

3.1 정의

[정의1] 패턴 p에 대한 적응 가중치 지지도(AWS: Adaptive Weighted Support)는 식(1)과 같다[11].

$$AWS(X) = \sum_{i=1}^{N} Weight(X,i) \times Support(X,i) \qquad (1)$$

여기서 N은 배치의 개수를 말하고 Weight(X,i)는 X에 속하는 i번째 배치에 있는 X의 가중치이고, Support(X,i)는 i번째 배치에 있는 X의 지지도이다.

[정의2] 패턴 x의 AWS(p)의 값이 주어진 최소 임계값보다 크거나 같을 때 패턴 x를 적응 가중치 빈발 패턴이라 한다. 만일 최소 임계값이 1.2라면 <Table 1>에서 패턴 “bd”는 적응 가중치 빈발 패턴이다. 패턴 “be”의 적응 가중치 지지도 (AWS(be))는 ((0.7 + 0.5) / 2) * 2 + ((0.4 * 0.6) / 2) * 1 = 1.7이다. 여기서 최소임계(minimum threshold) 값이 1.4라면 패턴 “be”는 적응가중치 빈발 패턴이다.

<Table 1> An adaptive weighted database

Batch	TID	Trans.	Weight				
1st	T_1 T_2 T_3	b c e a b e b c	a 0.3	b 0.7	c 0.5	d 0.2	e 0.5
2nd	T_4 T_5 T_6	a c c d e c e	a 0.6	b 0.2	c 0.5	d 0.3	e 0.5
3rd	T_7 T_8 T_9 T_{10}	a b e c e a c d c d e	a 0.6	b 04	c 0.7	d 0.4	e 0.6

이러한 적응 빈발 패턴은 하향 폐쇄의 성질을 유지할 수 없다. AWS(d) = 0.3*1 + 0.4*2 = 1.1이고, AWS(cd) = ((0.5 + 0.3) / 2) * 1 + ((0.7 + 0.4) / 2) * 2 = 1.5로 패턴 “d”는 빈발 패턴에 해당되지 않는다.

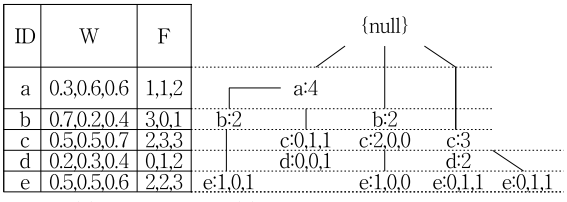
[정의3] 패턴 p에 대한 무게중심에 의한 적응 가중치 지지도(CAWS: Centroid based Adaptive Weighted Support)는 식(2)와 같이 정의한다.

$$CAWS(P) = \frac{\sum_{i=1}^{N} Weight_i(P) * Support_i(P)}{\sum_{i=1}^{N} Weight_i(P)} \quad (2)$$

본 논문에서 정의된 식(2)에 의하여 항목 “d”에서 CAWS는 (0.3*1 + 0.4*2) / (0.3 + 0.4) = 1.55가 된다. 또한 항목 “c”에서 CAWS는 (0.5*2 + 0.5*3 + 0.7*4) / (0.5 + 0.5 + 0.7) = 3.12가 된다. 패턴 “cd”는 패턴 “c”와 패턴 “d”의 선형결합이므로 항목 “cd”의 CAWS는 (1.55 + 3.12) / 2 = 2.35가 되어 하향 폐쇄의 성질을 유지한다.

3.2 트리의 생성

이 절에서는 적응 가중치를 갖는 항목들로 이루어진 트랜잭션들의 내용을 저장하는 CAWFP-트리의 구조 및 생성 방법을 설명한다. CAWFP-트리는 루트노드와 일련의 Prefix-트리로 구성되고 헤더 테이블에는 항목 id, 가중치와 지지도가 포함되어 있다. 각 배치의 트랜잭션들이 삽입되어 CAWFP-트리가 사전식으로 구성되어 진다. 또한 FP-트리 와 같이 항목들의 배치에 따른 빈도수 정보를 통합하기 위한 단말노드(Tail Node)와 배치별 지지도 정보를 가지는 일반노드로 구성되어 진다. 따라서 모든 노드는 두 가지의 노드로 구성되어 진다. 예를 들어 세 개의 배체에서 두 번째 배치에서 테일 노드 “b”가 처음으로 존재하면 노드의 구성은 b:0,1,0 이 된다. [Fig. 1]은 <Table 1>의 데이터베이스를 이용하여 구성한 트리의 모습을 보여준다.

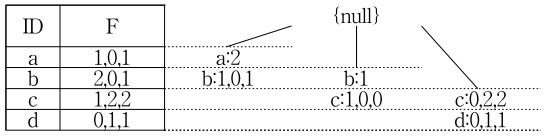


(a) Header Table (b) Conditional CAWFP-Tree

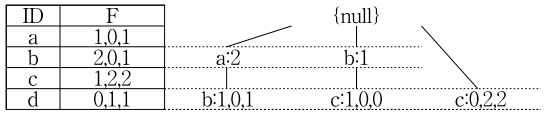
[Fig. 1] CAWFP-tree of <Table 1> database

3.3 마이닝 과정

제안하는 CAWFP 알고리즘은 FP-Growth처럼 트리의 상향 탐색을 통하여 수행된다. [정의 3]에서 언급하였던 것처럼 식(2)에 의하여 항목집합의 가중치와 지지도의 중심을 측정하여 하향 폐쇄의 여부를 결정한다. 항목들이 가지는 가중치에만 의존하는 기존의 방법과 달리 가중치와 빈도수를 같이 고려하여 보다 정확한 적응 가중치를 측정함으로써 탐색 시간과 정보의 손실을 최소화 할 수 있게 된다. <Table 1>을 이용하여 구성한 [Fig. 1]의 트리를 가지고 최소 임계값이 1.4일 경우 마이닝하는 과정을 살펴보면 우선 식(2)에 의한 적응 가중치 지지도는 <a:1.4, b:2.72, c:2.7, d:1.22, e:2.375> 가 된다. 여기서 d 항목은 최소 임계치보다 작기 때문에 빈발 패턴의 가능성이 있는 항목은 a, b, c, e 이다.



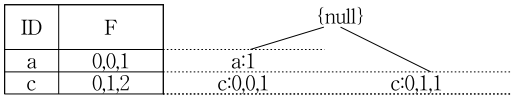
(a) CAWFP-tree and conditional-pattern base of “e”



(b) CAWFP-tree and conditional-tree of “e”

[Fig. 2] CAWFP-pattern base and tree of item “e”

따라서 네 개의 항목에 대하여 트리의 상향 탐색을 통하여 적응 가중치 빈발 패턴을 찾게 된다. 먼저 헤더 테이블의 가장 밑에 있는 항목 “e”에 대한 CAWFP-트리를 구성하기 위해서는 [Fig. 2]의 CAWFP-트리에서 “e” 항목을 포함하는 가지들을 빈도수를 고려하여 추출하면 <ab:1,0,1, bc:2,0,1, cd:1,2,2, c:0,1,1> 이 된다. 여기서 불필요한 항목을 추출하기 위하여 항목의 가중치 빈도수는 <a:1,0,1, b:2,0,1, c:1,2,2, d:0,1,1> 가 되어 이를 이용하여 적응 가중치 빈도수를 구하면 <a:1.4, b:1.34, c:1.58, d:0.78> 이다. 따라서 항목 a 와 d는 Conditional CAWFP-트리에서 제거된다.



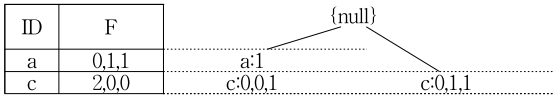
(a) CAWFP-tree and conditional-pattern base of “d”

(b) CAWFP-tree and conditional-tree of “d”

[Fig. 3] CAWFP-pattern base and tree of item “d”

이 과정에서 “be” 와 “ce” 가 후보 패턴으로 추출되고 “bce”의 적응 가중치 지지도는 1.0이 되어 역시 Conditional CAWFP-트리에서 제거되어 추출된 패턴은 <a, b, c, e, be, ce> 이다.

같은 방법으로 항목 “d”에 대한 CAWFP-트리와 Conditional CAWFP-트리를 구성하면 [Fig. 3]과 같다. “d” 항목을 포함하는 가지들을 빈도수를 고려하여 추출하면 <ac:0,0,1, c:0,1,1> 이 된다. 단일 항목의 가중치 빈도수는 <a:0,0,1, c:0,1,2> 가 되어 이를 이용하여 적응 가중치 빈도수를 구하면 <a:1.0, c:1.72> 이다. 따라서 a는 제거되고 “cd” 의 적응 가중치 지지도가 “c”와 “d”의 선형결합으로 (1.58 + 1.57) / 2 = 1.57로서 후보 패턴으로 추출된다. 지금까지의 추출된 패턴은 <a, b, c, e, be, ce, cd> 이다.



(a) CAWFP-tree and conditional-pattern base of “c”

(b) CAWFP-tree and conditional-tree of “c”

[Fig. 4] CAWFP-pattern base and tree of item “c”

Input: DB contains batches of transactions with adaptive weights, minimum threshold(δ)
Output: Adaptive weighted frequent patterns

1. Forach each transaction T_i in DB Do
2. Insert T_i into Tree byexicographical order
3. If isTailNode(item) Then
4. theNode ← item.freq in each batches
5. Else
6. theNode ← sum(tailNode.freq as its children)
7. L=∅
8. Foreach item α_i in the bottom of headerTable Do
9. If CAWS(ai) 〉 δ Then
10. Create CAWFP-tree for item α_i
11. Call Mining(PT_i, α_i)
12. Generate candidate 1-itemset C_1
13. Procedure Mining(T, α)
14. Create conditional tree CDT of α by deciding
15. each item i from T having CAWS(d_i) 〈 δ
16. Foreach item β_i in the headerTable of CDT Do
17. Call Test_Candidate(αβ)
18. Create CAWFP-tree T_β for itemset αβ
19. Calculate CAWS
20. Call Mining(T_β, αβ)
21. Procedure Test_Candidate(X)
22. CAW_X ←Center adaptive weight support of X
23. If $CAW_X \geq \delta$ Then L=L∪CAW_X

항목 “c”에 대한 CAWFP-트리와 Conditional CAWFP-트리를 구성하면 [Fig. 4]와 같다. “c” 항목을 포함하는 가지들을 빈도수를 고려하여 추출하면 <a:0,1,1, b:2,0,0> 이 된다. 이에 대한 적응 가중치 빈도수를 구하면 <a:1.0, b:2.0> 이다. 따라서 a는 제거되고 “bc” 의 적응 가중치 지지도가 “b”와 “c”의 선형결합으로 ((1.4/0.7) + (1.0/0.5)) / 2 = 2.0로서 후보 패턴으로 최종적인 적응 가중치 빈발패턴은 <a, b, c, e, be, ce, cd, bc> 가 된다. 지금까지 설명한 마이닝 과정을 알고리즘으로 기술하면 아래와 같다.

3.4 맵리듀스 기반의 모델링

이상과 같이 구성한 CAWFP-트리를 Jar화일을 맵리듀스의 Job으로 연결하여 아래와 같이 가상분산 모드의 하둡환경에서 모델링한다. 본 논문에서 제시하는 무게중심에 의한 빈발 패턴트리와 기존의 두 가지의 다른 방법들간의 비교우위를 논하기 위하여 각각의 트리에 해당하는 각 알고리즘의 Jar화일을 구성하여 맵리듀스 Job에 연결하여 수행한다[12].

```
public class CAWFPDriver extends Configured implements
Tool{
 @Override
 public int run(String[] args) throws Exception {

   Configuration conf = new Configuration();
   conf.set("mapreduce.job.jar","CAWFP.jar");

   Job job = Job.getInstance(conf);
   job.setMapperClass(CountCAWFPMapper.class);
   job.setReducerClass(CountCAWFPReducer.class);
   job.setMapOutputKeyClass(Text.class);
   job.setMapOutputValueClass(Text.class);
   job.setOutputKeyClass(Text.class);
   job.setOutputValueClass(Text.class);
   FileInputFormat.addInputPath(job, new Path(args[0]));
   FileOutputFormat.setOutputPath(job,                new
Path(args[1]));

   return job.waitForCompletion(true) ? 0 : 1 ;
 }
}
```

트랜잭션 데이터의 각 항목에 대하여 키-값의 쌍으로 구성하여 CAWFPMapper 로 입력하고 빈도수 계산을 수행한다. 이후 CAWFP 리듀서를 통하여 최소지지도를 유지하는 빈발패턴을 출력한다.

4. 실험 결과 및 분석

4.1 실험 환경 및 데이터 집합

실험은 제안된 빈발 패턴 알고리즘의 성능을 FP-Growth를 이용하는 기존의 AWFPM 알고리즘과 AWFIMiner 알고리즘과의 비교분석하기 위해서 하나의 클러스터에 국한하여 수행한다. 노드는 CentOS 64비트 운영체제 , 인텔 i7-7700 3.60GHz, 16GB 메모리로 구성되고, 맵리듀스 프레임워크는 Hadoop 버전 2.7.3 을 사용한다[13,14,15,16,17,18,19,20,21].

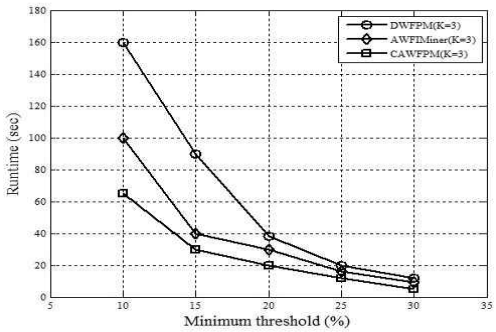
데이터의 특성을 고려하기 위하여 트랜잭션의 수와 길이가 반비례하는 형태로 <Table 2>와 같이 구성하였다. Mushroom은 조밀한 데이터집합으로 배치별로 기본적으로 3,000개씩 분할하였다. T10I4D100K는 합성데이터로서 트랜잭션의 길이(10), 항목의 개수(1,000), 트랜잭션 수(100,000)를 가지고 있고 배치당 기본적으로 35,000개씩 데이터를 할당하였다. 또한 각각의 전체 데이터를 3세 개의 배치(K=3)로 분할하였고 가중치는 공히 0.1~0.9까지의 값을 추가하여 Java로 구현하여 수행하였다.

<Table 2> Datasets characteriscics

Datasets	#Trans.	#Items	Avg. of Length
Mushroom	8,124	119	23
T10I4D100K	100,000	892	10.1

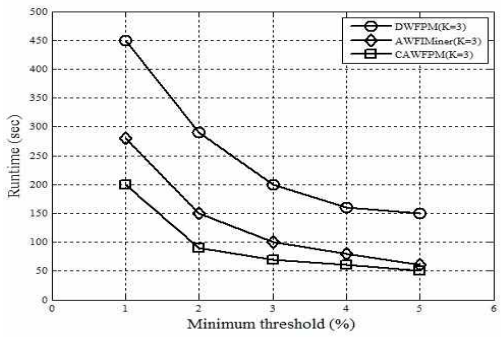
4.2 성능 분석

배치의 개수가 세 개로 분할된 각 데이터 집합에 대하여 [Fig. 5]는 Mushroom에 대하여 최소 지지도 값이 10% ~30%로 변화할 때의 맵리듀스 기반 빈발 패턴 마이닝의 실행시간을 나타낸다. 또한, [Fig. 6]에서는 최소 임계치의 범위가 1% ~ 5%인 T10I4D100K데이터 집합의 실행시간을 나타낸다. 동적 가중치를 사용한 DWFPM의 경우보다 AWFIMiner이 양호한 결과를 보였다. 이는 노드의 종류를 두 가지로 운영하여 데이터베이스의 트랜잭션에 대한 배치정보를 트리에 저장함으로써 메모리의 할당량을 줄일 수 있었다. 그러나 기존의 두 가지의 방법은 모두 하향 폐쇄의 성질을 만족하기 위하여 최대 지지도 임계치를 설정하여 데이터베이스의 탐색 공간을 통하여 트리를 생성하고 있다.



[Fig. 5] Runtime performance for Mushroom

제안된 방법은 무게중심에 의한 방법을 이용하여 최대 지지도 임계치를 사용하지 않고도 하향 폐쇄의 성질을 유지하여 최소 지지도 임계치만에 의한 탐색공간이 확보되기 때문에 자연히 좋은 성능을 보일 수 있다. 또한 보다 빠른 처리 시간이 요구되는 스트림 데이터 처리 환경에서도 잘 적용될 수 있을 것이다. 세 개의 배치로 진행된 각 실험에서 할당된 메모리는 Mushroom데이터의 경우 DWFPM 방법이 1.26MB을 보였고 AWFIMiner 방법이 0.94 MB이었다. T10I4D100K 의 경우에는 DWFPM 방법이 17.61MB이고 AWFIMiner 방법은 14.72MB이었다.



[Fig. 6] Runtime performance for T10I4D100K

5. 결론

가중치를 가지는 대부분의 빈발 패턴 마이닝 알고리즘에서 하향 폐쇄의 성질을 유지하기 위하여 최대치 지지도 임계치를 사용한다. 본 논문에서는 최대치 지지도 임계치를 사용하지 않고 하향 폐쇄의 성질을 유지 할 수 있는 무게중심 방법을 제안하였다. 항목이 가지는 원래

의 가중치와 빈도수와의 무게중심을 이용하여 빈발 항목들의 정확성과 변별력을 향상시켰다. 또한 이를 맵리듀스로 모델링을 통하여 최소 지지도 임계치에 의한 탐색 공간에 의한 트리의 생성으로 인하여 실행시간이 자연히 줄어드는 결과를 보임을 알 수 있었고 기존의 가중치를 가지는 빈발 패턴 마이닝 알고리즘과 성능을 비교하였다.

제안된 방법은 스트림 데이터 분석과 같은 보다 빠른 속도와 정확한 결과가 요구되는 환경에 기반을 마련하는 데에 중점을 두었다. 향후 연구로는 완전 분산 맵리듀스 모델링을 통하여 제안한 방법의 고속 병렬처리에의 특성에 대한 연구가 필요하다.

ACKNOWLEDGEMENTS

This paper was supported by Joongbu University Research & Development Fund, in 2016.

REFERENCES

[1] R. Agrawal, R. Srikant, "Fast Algorithm for Mining Association Rules", In: 20th Int. Conf. on Very Large Data Bases, pp. 487~499, 1994.

[2] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, "Mining Weighted Frequent Patterns using Adaptive Weightes", In: Fyfe et al. (Eds.): IDEAL 2008, LNCS 5326, pp. 258~265, 2008.

[3] C. H. Cai, A. W. C. Fu, C. H. Cheng, W. W. Kwong, "Mining Association rules with weighted items", In Proceedings of Intl. Database Engineering and Applications Symposium (IDEAS 1988) , Cardiff, Wales, UK, July pp. 68~77, 1998.

[4] F. Tao, "Weighted association rule Mining using Weighted Support and Significant Framework", In: 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining", pp. 661~666, 2003.

[5] W. Wang, J. Yang, P. S. Yu, "WAR: Weighted Association Rules Item Intensities", Knowledge Information and Systems, No. 6, pp. 203~229, 2003.

[6] U. Yun, J. J. Leggett, "WFIM: Weighted Frequent Itemset Mining with a wieght range and a minimum weight", Society for Industrial and Applied Maathematics, Proceedings of the 2005 SIAM International Conference on Data Mining, pp. 636~640, 2005.

[7] U. Yun, "Efficient Mining of Weighted Interesting Patterns with A Strong Weight and/or Support Affinity", Information Sciences, Vol. 177, pp. 3477~3499, 2007.

[8] U. Yun, "An Efficient Mining of Weighted Frequent Patterns with Length Decreasing Support Constraints", Knowlwdge-Based Systems, Vol. 21, Issue 8, Dec., pp. 741~752, 2008.

[9] S. Zhang, C. Zhang, X. Yan, "Post-Mining: Maintenance of Association Rules by Weighting", Information Systems, Vol. 23, pp. 691~707, 2003.

[10] J. E. Shin, B. H. Jeong, D. H. Lim, "BigData Distribution System using RHadoop", Society of Data Information Science, Vol. 36, No. 5, pp. 1155~1166, 2015.

[11] H. L. Nguyen, "An Efficient Algorithm for Mining Weighted Frequent Itemsets Using Adaptive Weights", I.J. Intellogent Systems and Appillcations, Vol. 11, pp. 41-48, 2015.

[12] K. U. Jeon, M. S. Kim "Frequent Pattern Mining Technique of BigData using MapReduce Framework", Korea Information Processing Society, Vol. 21, No. 3, pp.17-25, 2014..

[13] G. W. Jin, "A Study on the Data Collection Methods based Hadoop Distributed Environment", Korea Convergence Society, Vol. 7, No. 5, pp. 1-6, 2016.

[14] Y. J. Kim, "Convergence of Business Information System Process using Knowledge-based Method", Korea Convergence Society, Vol. 6, No. 4, pp. 65-71, 2015.

[15] J. H. Gu, "A Study on the Machine Learning Model for Product Faulty Prediction in Internet of Things Environment", Convergence Society for SMB, Vol. 7, No. 1, pp. 55-60, 2017.

[16] S. Y. Hong, "New Authentication Methods based

on User's Behavior Big Data Analysis on Cloud", Convergence Society for SMB, Vol. 6, No. 4, pp. 31-36, 2016.

[17] I. K. Park, "An Improvement of the Decision Making of Categorical Data in Rough Set Analysis", Journal of Digital Convergence, Vol. 13, No. 6, pp.157-164, 2015.

[18] I. K. Park, "The Generation of Control Rules for Data Mining", Journal of Digital Convergence, Vol. 11, No. 11, pp.343-349, 2013.

[19] I. K. Park, "Clustering Algorithm for Data Mining using Posterior Probability-based Information Entropy", Journal of Digital Convergence, Vol. 12, No. 12, pp.293-301, 2014.

[20] B. R. Hwang, S. G. Kim, "On Implementing a Learning Environment for Big Data Processing using Raspberry Pi", Journal of Digital Convergence, Vol. 14, No. 4, pp.251-258, 2016.

[21] Apache Hadoop, http://hadoop.apache.org/

박 인 규(Park, In Kyu)

- 1985년 2월 : 원광대학교 전기과 졸업
- 1985년 2월 : 연세대학교 전기과 전자계산기 응용(공학석사)
- 1997년 2월 : 원광대학교 전자과 마이크로 프로세서 응용(공학박사)
- 1997년 3월 ~ 현재 : 중부대학교 게임소프트웨어학과 교수
- 관심분야 : 소프트컴퓨팅, 데이터마이닝
- E-Mail : fip2441g@gmail.com