

Safe와 Non-safe 전력 부하 라인 분석을 위한 TFP트리 기반의 점진적 출현패턴 마이닝

TFP tree-based Incremental Emerging Patterns Mining for Analysis of Safe and Non-safe Power Load Lines

이 종 범* 박 명 호** 류 근 호***
Jong Bum Lee Minghao Piao Keun Ho Ryu

요약 본 논문에서는 특정 지역의 전력 소비 데이터를 이용하여 safe와 non-safe 전력 부하 라인의 차이를 분석하여 정의하고, 출현패턴을 사용하여 잠재되어 있는 non-safe라인을 식별하기 위하여 제한된 메모리에서 효율적으로 패턴을 찾을 수 있는 TFP-tree 기반의 점진적 출현패턴 마이닝 알고리즘을 제안한다. 특히, 두 개의 다른 최소 지지도 값을 사용하여 전력 소비 데이터와 같은 대용량 데이터에서의 마이닝 문제를 해결한다.

키워드 : 출현패턴 마이닝, 점진적 마이닝, TFP-tree 알고리즘

Abstract In this paper, for using emerging patterns to define and analyze the significant difference of safe and non-safe power load lines, and identify which line is potentially non-safe, we proposed an incremental TFP-tree algorithm for mining emerging patterns that can search efficiently within limitation of memory. Especially, the concept of *pre-infrequent patterns* pruning and use of two different *minimum supports*, made the algorithm possible to mine most emerging patterns and handle the problem of mining from incrementally increased, large size of data sets such as power consumption data.

*이 논문은 2010년도 충북대학교 학술 연구 지원 사업으로 수행되었습니다.

* 충북대학교 컴퓨터과학과 석사수료
jongbumlee@dblab.chungbuk.ac.kr

** 충북대학교 컴퓨터과학과 박사재학
bluemhp@dblab.chungbuk.ac.kr

*** 충북대학교 소프트웨어학과 및 컴퓨터공학 교수(교신저자)
khryu@dblab.chungbuk.ac.kr

Keywords : Emerging Patterns, incremental mining, TFP-tree

1. 서론

최근 전력 산업분야에서는 효율적인 산업 운영과 계획을 위하여 고객의 전력 사용 패턴의 특성을 분석하고 예측하는 기술에 대하여 많은 연구들이 진행되어 왔다. 기존에는 통계적인 방법을 많이 사용하였지만, 최근에는 데이터 마이닝 기법들을 적용하는 사례가 늘어나고 있다[9,13]. 최근에 소개된 이동객체의 시간과 공간을 기반으로 시공간 패턴[16, 17]을 추출하는 방법을 전력산업에 응용하여 실시간으로 빈발패턴을 찾아내어 부하 라인을 예측하는 점진적 마이닝 방법이 필요하다.

패턴 기반의 기법들을 사용하면 비슷한 소비 패턴을 가지고 있는 고객들에 대한 분류를 하고, 맞춤형 서비스를 제공하는 등 여러 가지 정책 및 계획 수립에 실용적인 참고가치를 제공할 수 있다. 또한 전력 소비 데이터와 같은 시계열 데이터들은 데이터 사이즈가 한정되어 있는 것이 아니고 계속해서 증가함으로 여러 번의 스캔을 필요로 하는 Apriori [10]기반의 기법보다는 단 한 번의 스캔만을 진행하여 메모리에 기본 정보들을 저장하는 알고리즘이 필요하다. 기존에는 FP-tree 및 FP-Growth 알고리즘[6]과 같은 효율적이고, 빠른 수행 속도를 보여주는 알고리즘을 기반으로 하는 점진적 빈발패턴 마이닝 기법들이 많이 연구 되었다. 그러나 알고리즘은 수행속도 뿐만 아니라 메모리도 효율적으로 관리할 수 있어야 한다.

TFP-tree 알고리즘[2]은 메모리 사용 효율이 뛰어난 빈발패턴을 찾는 알고리즘 중 하나이다. 또한, 본 논문에서는 전력 부하 라인이 safe 라인 혹은 non-safe라인인지 명확하게 구분하기 위하여 출현패턴 기반의 마이닝[3] 기법을 사용한다. 빠른 실행 속도 뿐만 아니라 효율적인 메모리 관리 전제하에 전력 부하 패턴에 대한 정확한 분석을 위하여 본 논문에서는 TFP-tree 기반의 점진적 출현패턴 마이닝 알고리즘을 제안하였으며, 이상 패턴을 분석하여 정전 등 사고를 유발할 수 있는 non-safe 전력 부하 라인을 구별하기 위하여 출현패턴 기반의 분류작업을 진행하였다.

2. 관련 연구

2.1 출현패턴의 정의

출현패턴(Emerging Patterns)이란, 주어진 클래스별 데이터 집합들의 차이를 최대화하여, 두 개의 분할된 데이터 집합을 명확하게 구분해 줄 수 있는 빈발패턴의 특수한 형태이다.

정의 2.1.1: 두 개의 서로 다른 클래스에 해당되는 두 데이터 집합 D_1 , D_2 에 대해, 패턴 X 의 성장률(GR : growth rate)은 다음과 같이 계산한다.

$$GR(X) = \begin{cases} 0 & \text{if } \sup_{D_1}(X) = 0 \text{ and } \sup_{D_2}(X) = 0 \\ \infty & \text{if } \sup_{D_1}(X) = 0 \text{ and } \sup_{D_2}(X) > 0 \\ \frac{\sup_{D_2}(X)}{\sup_{D_1}(X)} & \text{otherwise} \end{cases} \quad (1)$$

여기서, $D_1(D_2)$ 을 배경 데이터 집합, $D_2(D_1)$ 를 목표 데이터 집합이라고 하며, 성장률이 주어진 경계값을 만족시키는 패턴을 목표 데이터 집합 $D_1(D_2)$ 의 출현패턴이라고 한다.

출현패턴 X 의 strength(Str)는 식 2와 같이 정의되며, CAEP[4]에서는 분류를 위하여 출현패턴들의 strength(Str)를 기반으로 하는 score(SC)를 계산하여 분류기준으로 하는 기법을 소개하였다.

$$Str(X) = \frac{GR(X)}{GR(X) + 1} \times \sup(X) \quad (2)$$

정의 2.1.2: 인스턴스 T 와 각 클래스 C_i 에서 발견된 출현패턴의 개수 $E(C_i)$ 가 주어졌을 때, 각 클래스에서 T 의 score는 다음과 같이 계산할 수 있다.

$$SC(T, C_i) = \begin{cases} \sum_{X \in T, X \in E(C_i)} \frac{GR(X)}{GR(X) + 1} \times \sup c_i(X), GR > \rho \\ \sum_{X \in T, X \in E(C_i)} \sup c_i(X), GrowthRate = \infty \\ 0, GR = 0 \text{ or } GR < \rho \end{cases} \quad (3)$$

2.2 전처리 기법

본 논문에서는 새로운 데이터가 추가될 때 생성되는 빈발패턴들에 대한 효과적인 처리를 위하여 FUP[1]에서 소개된 처리 기법을 사용한다.

정리 2.2.1 : 기존 마이닝된 빈발패턴 집합에는 존재하지 않지만, 새로 추가되는 데이터 집합에서 빈발하는 k-itemsets은 기존 데이터베이스가 업데이트된 후에도 빈발할 가능성이 있다[1].

이 정리를 기반으로 원본 데이터베이스를 업데이트

하기 전에 새로 추가되는 데이터 집합에서 패턴들에 대한 지지도를 카운트하는 것만으로 쉽게 후보 집합들에 대한 pruning을 할 수 있다.

점진적 빈발패턴 마이닝에서 또 하나의 효과적인 전처리 기법은 *pre-minimum support*[15]를 사용하여 *pre-pruning*을 하는 것이다. 즉, 아이템 X 의 지지도가 *pre-minimum support*와 같거나 크다면 아이템 X 는 *pre-frequent item*이며, 반대로 아이템 X 의 지지도가 *pre-minimum support*보다 작다면, 아이템 X 는 *pre-infrequent item*이다. 본 논문에서는 *pre-minimum support* 개념을 TFP-tree에 접목함으로써 정보의 손실을 줄이고, 빈발패턴을 효과적으로 찾을 수 있게 하였다.

2.3 점진적 출현패턴 마이닝의 문제점

minimum support 와 *pre-minimum support* 두 지지도 임계값을 사용하여 점진적 출현패턴 마이닝을 진행 할 시 그림 1에서와 같은 9가지의 경우 중 하나에 속하게 된다[11]. 여기서 Pre-large itemset 은 *pre-frequent itemset*을 말한다.

		New transactions		
		Large itemsets	Pre-large itemsets	Small itemsets
Original database	Large itemsets	Case1	Case2	Case3
	Pre-large itemsets	Case4	Case5	Case6
	Small itemsets	Case7	Case8	Case9

그림 1. 가능한 9가지 Case

출현패턴을 마이닝 할 때 Case 1, 5, 6, 8, 9 등 5가지 경우는 쉽게 처리할 수 있지만, Case 2, 3, 4, 7 등 4가지 경우에는 빈발할 수 있는 아이템임에도 불구하고 빈발하지 않는 아이템이 될 수 있다. 또한, 데이터베이스가 계속 업데이트됨에 따라서 새로 추가되는 트랜잭션의 개수가 기존 트랜잭션에 비해 상대적으로 매우 적게 되므로, Case7과 같은 경우 빈발 가능성이 있는 아이템도 빈발하지 않는 아이템이 될 수 있다. 따라서 새로 추가되는 데이터베이스의 트랜잭션의 수가 식 4의 t 보다 작게 되면 빈발할 수 있는 아이템이 줄어들기 때문에 식 4를 사용하여 새로 추가되는 트랜잭션의 개수를 체크하여 후보 아이템 집합에 대한 빈발여부를 최종 체크한다.

$$t \geq \frac{(S_h - S_l)d}{1 - S_l} \quad (4)$$

이때, t 는 새로운 트랜잭션의 수, S_h 는 높은 임계값, S_l 은 낮은 임계값을 나타내며, d 는 원본 트랜잭션의 수이다. 새로운 트랜잭션의 수가 경계치 t 를 만족 할 때 후보 아이템 집합에 대한 빈발 여부를 계산한다.

3. TFP-tree 기반의 점진적 출현패턴 마이닝

본 논문에서는 TFP-tree기반의 점진적 출현패턴 마이닝 알고리즘을 IEP-TFP (TFP tree-based Incremental Emerging Patterns mining algorithm) 라 명하고, P-tree와 P-tree table 및 T-tree 의 노드 구조를 출현패턴 마이닝이 가능하도록 확장하였다.

3.1 원본 데이터베이스로부터의 P-tree와 P-tree table 구축

본 논문에서 제안하는 IEP-TFP알고리즘의 수행 과정은 다음과 같은 샘플 데이터 집합을 이용하여 설명한다. $DB = \{D_1: a,c,d\}, \{D_2: b,c,d\}, \{D_3: a,b,c\}, db = \{D_1: a,b,d\}, \{D_2: b,c,d\}$;

먼저 샘플 데이터 집합에서 DB 를 스캔하여 P-tree를 구축한다. 트리는 각 후보 아이템들을 기반으로 최상위 레벨을 우선 구축한다. 첫 번째 트랜잭션 $\{D_1: a,c,d\}$ 은 최상위 레벨 {a}노드에 대한 자식노드로 아이템 집합 {c, d}가 P-tree에 추가하게 된다. 트랜잭션의 클래스 라벨이 D_1 이기 때문에 해당 아이템 노드는 D_1 에 대한 지지도 1을 증가시킨다. 다음으로, 추가되는 트랜잭션에 대하여 이미 생성되어 있는 아이템 노드이면 해당 클래스의 지지도를 증가시킨다. 반대로, 노드가 생성되어 있지 않은 아이템에 대해서는 아이템 노드를 생성한 후 해당 클래스의 지지도를 증가하게 된다. 그림 2는 DB 로부터 구축된 최종 P-tree이다.

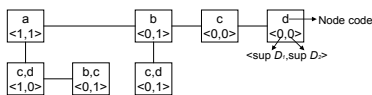


그림 2. DB로부터 P-tree 구축

P-tree의 구축이 끝나게 되면 P-tree table을 구

축하게 된다. 그림 3은 그림 2의 P-tree로부터 구축된 P-tree table이다.

P-tree table에는 P-tree의 모든 정보가 저장되어 있기 때문에 메모리에 저장하여 사용하게 되면 알고리즘의 계산 시간을 단축 할 수 있다.

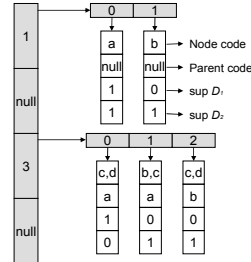


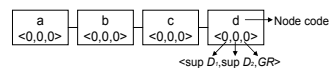
그림 3. P-tree table

3.2 출현패턴 마이닝을 위한 T-tree 구축

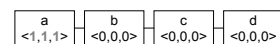
본 논문에서 구축하는 EP-T tree(T-tree for mining emerging patterns)는 [2]에서 소개된 T-tree 구축과정과 같은 방법으로 그림 4(a)와 같이 후보 아이템들을 기반으로 최상위 레벨을 구축한다. 원래의 T-tree 구축과정과 다른 점은 T-tree 구축 과정에서 빈발하지 않는 아이템이 새로 추가되는 아이템 집합 때문에 빈발할 수도 있다는 점을 고려하여 가지치기를 하지 않는다는 점이다. 다음은 T-tree 구축과정에 대한 설명이다.

1) P-tree table을 스캔하여 EP-T tree의 최상위 레벨을 생성한다. 그림 4(a)와 같이 확장된 각 아이템 노드에 아이템 코드와 클래스별 지지도, Growth Rate (GR)를 저장할 수 있다.

2) P-tree table의 레벨 1부터 스캔이 시작되며, 첫 번째 스캔된 아이템 노드 코드는 {a}이며, 아이템의 지지도(D_1 에 대한 지지도 1, D_2 에 대한 지지도 1)와 GR 를 EP-T tree 최상위 레벨에 저장하며 그림 4(b)와 같다.



(a) 최상위 레벨



(b) P-tree 테이블, 레벨 1, 노드 1

그림 4. EP-T tree $_{DB}$ 의 최상위 레벨의 구축

같은 방법으로 P-tree table에 있는 모든 아이테에 대하여 스캔을 하게 되면 그림 5와 같이 DB로부터 가지치기를 하지 않은 최종 EP-T tree_{DB}를 구축할 수 있다.

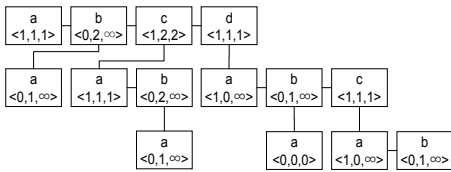


그림 5. EP-T tree_{DB} 구축

DB에 대한 T-tree 구축 후, 새로 추가되는 db는 식 4를 사용하여 트랜잭션의 수가 일정한 조건을 만족하지 않으면 DB와 결합하지 않고 대기상태에 있게 한다. 조건을 만족시키게 되면 db의 트랜잭션은 EP-T tree_{DB} 구축 방법과 동일한 방법으로 EP-T tree_{db}를 구축하게 되고, pre-minimum support의 개념을 사용하여 pruning을 진행한다. 그림 6은 pre-minimum support값을 1.5로 설정하여 구축한 EP-T tree_{db}이다.

다음으로 EP-T tree_{db}와 EP-T tree_{DB}를 결합한다. 이때 지지도 계산은 $X.supp_{UD} = X.supp_{db} + X.supp_{DB}$ 방법으로 계산하게 된다. 여기서 $X.supp(UD,db,DB)$ 는 아이테 X에 대한 UD, db, DB의 지지도이다. 이 과정은 새로운 db가 결합될 때마다 반복된다. 마지막으로 EP-T tree_{UD}에서 빈발하지 않는 아이테들을 pruning한다. 그림 7은 minimum support를 2로 설정하여 완성된 최종 EP-T tree_{UD}이다.

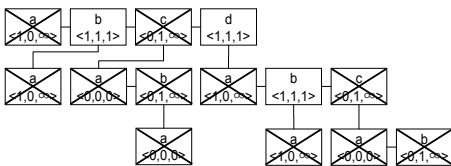


그림 6. 구축된 EP-T tree_{db}

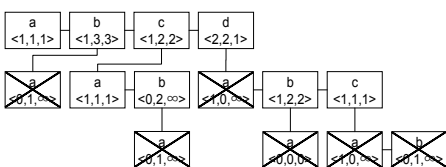


그림 7. EP-T tree_{DB} ∪ EP-Ttree_{db}

4. 실험 평가 및 분석

본 논문에서는 IEP-TFP알고리즘을 사용하여 safe 및 non-safe 라인 식별을 위하여 한국 전력 연구원(KEPRI)에서 제공받은 특정지역의 전력 사용량 데이터 11,228건을 사용하였다. 데이터 구성은 하루를 15분 간격의 전력 사용량을 수치로 나타내어진 데이터로, 차원 축소를 위하여 7가지의 전력 구간(하루 24시간을 F1, F2- 23:00~07:00(8시간), F3- 12:00~15:00(3시간), F4- 00:00~07:00(7시간), F5- 09:00~12:00(3시간), F6- 13:00~17:00(3시간), F7- 19:00~23:00(4시간))으로 나누었다. 그리고 하루 동안 사용한 전력 데이터에 두 개의 속성(average와 peak)값을 추가하였다. 그 후, 전력 데이터는 연속형 속성 값을 갖고 있기 때문에 출현패턴 마이닝 수행을 위해서 엔트로피 기반의 이산화 기법[12]을 적용하였다.

본 논문에서는 분류의 정확성을 평가하기 위하여 5가지의 분류 알고리즘(CMAR[14], Bayesian Network[8], Naïve Bayesian[8], Support Vector Machine[5], C4.5[7])과 비교 하였다. 표 1은 IEP-TFP 알고리즘을 포함한 6가지의 분류알고리즘의 실행 시간과 정확성의 결과를 보여준다.

표 1. 분류의 정확성

Classifiers	Accuracy(%)	Time(sec)
IEP-TFP	93.59	0.02
CMAR	82.94	0.42
Bayesian Network	95.13	0.11
Naive Bayesian	95.14	0.05
SVM	95.5	80.39
C4.5	95.79	0.06

표 2는 전력 데이터를 사용하여 클래스별 분류 정확도를 보여주고 있다.

표 3에서는 서로 다른 시간 간격 동안 클래스별 전력 부하 라인 중 비교적 많이 나타난 패턴을 샘플로 보여준다. 예를 들어 non-safe 전력 부하 라인은 최대값이 (382.75-3354)로 나타날 때, 19:00과 23:00사이 동안 27.3%의 전력 부하 라인이 안전하지 못하다는 판별이 나왔고, 같은 시간대에 평균이 (204.56-2783.55)일 때 31.9%가 전력 부하 라인이 안전하지 못하다는 판별이 나왔다.

표 2. 알고리즘 별 safe와 non-safe의 정확성

IEP-TFP		Predicted Class (%)	
		Safe	Non-safe
Actual Class (%)	Safe	99.99	0.01
	Non-safe	18.92	81.03
CMAR		Predicted Class (%)	
		Safe	Non-safe
Actual Class (%)	Safe	83.87	16.13
	Non-safe	20.5	79.5
Bayesian Network		Predicted Class (%)	
		Safe	Non-safe
Actual Class (%)	Safe	97.75	2.25
	Non-safe	9.98	90.02
Naive Bayesian		Predicted Class (%)	
		Safe	Non-safe
Actual Class (%)	Safe	97.77	2.23
	Non-safe	9.93	90.03
SVM		Predicted Class (%)	
		Safe	Non-safe
Actual Class (%)	Safe	98.97	1.03
	Non-safe	11.3	88.7
C4.5		Predicted Class (%)	
		Safe	Non-safe
Actual Class (%)	Safe	97.89	2.11
	Non-safe	8.31	91.69

표 3. 패턴 분석 결과

safety	Time intervals (hour)
Safe	F6-(0.067-0.076), max-(-inf-109.1)
	max-(-inf-109.1), F5-(0.061-0.081)
Non-safe	max-(382.75-3354), F7-(0.069-0.086)
	avg-(204.56-2783.55), F7-(0.069-0.086)
	max-(382.75-3354), avg-(204.56-2783.55), F7-(0.069-0.086)

5. 결론

본 논문에서는 점진적 출현패턴 마이닝을 위한 새로운 알고리즘 IEP-TFP를 이용하여 전력 부하 라인에 대한 분석을 진행하였다. Apriori-TFP 트리 기반으로 *pre-minimum support* 및 *minimum support* 두 가지 지지도 임계값을 사용하여 효과적으로 가능한 많은 출현패턴들을 찾아냈으며, 마이닝

된 출현패턴들을 사용하여 safe 및 non-safe 전력 부하 라인에 대한 분류작업을 진행하였다. 실험 결과, 제안한 IEP-TFP 알고리즘을 사용하여 훈련데이터로부터 추출된 출현패턴들은 충분한 정보를 가지고 있었으며 분류작업을 함에 있어서 safe 부하 라인은 거의 100% 정확하게 구분하였으며 non-safe 부하 라인을 구분함에 있어서도 80% 이상의 정확도를 보여줬다. 다만 훈련데이터가 서울 강남지역의 일부 테스트 지역에만 한정되어 훈련된 알고리즘으로부터 추출된 출현패턴들이 지역적 제한성을 가지게 되었으며, non-safe 부하 라인을 구분하기 위한 훈련데이터가 부족함으로 인해 non-safe 부하 라인을 구분함에 있어서 정확도가 다소 떨어졌다.

참고 문헌

- [1] D. W. Cheung, C. Y. Wong, J. Han, and V. T. Ng, 1996, "Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique," International Conference on Data Engineering, pp. 106-114.
- [2] F. Coenen, 2004, "The TFP Association Rule Mining Algorithm," Department of Computer Science, The University of Liverpool, UK, http://www.csc.liv.ac.uk/~frans/KDD/Software/Apriori_TFP/aprioriTFP.html.
- [3] G. Dong, and J. Li, 1999, "Efficient Mining of Emerging Patterns: Discovering Trends and Differences," ACM SIGKDD Int'l Conference Knowledge Discovery and Data Mining, pp. 43-52.
- [4] G. Dong, X. Zhang, L. Wong, and J. Li, 1999, "CAEP: Classification by Aggregating Emerging Patterns," Discovery Science, pp. 737-751.
- [5] J. C. Platt, 1998, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," Technical Report MST-TR-98-14, Microsoft Research.
- [6] J. Han, J. Pei, and Y. Yin, 2000, "Mining Frequent Patterns without Candidate Generation," ACM Special Interest Group on

Management of Data, pp. 1-12.

- [7] J. R. Quinlan, 1993, "C4.5: programs for Machine Learning," Morgan-Kaufmann publishers, San Mateo, CA.
- [8] N. Friedman, D. Geiger, and M. Goldszmidt, 1997, "Bayesian network classifiers," Machine Learning, Vol.29, No.2-3, pp. 131-163.
- [9] Q. Dou, K. Fu, H. Zhu, P. Jiang, and Z. Shi, 2010, "Associated Clustering and Classification Method for Electric Power Load Forecasting," IFIP Advances in Information and Communication Technology, Vol. 340/2010, pp. 112-121.
- [10] R. Agrawal, and R. Srikant, 1994, "Fast algorithm for mining Association rules," International Conference on Very Large Data Bases, pp. 487-499.
- [11] T. P. Hong, C. Y. Wang, and Y. H. Tao, 2001, "A new Incremental Data Mining Algorithm Using Pre-large Itemsets," Intelligent Data Analysis, Vol. 5, No. 2, pp. 111-129.
- [12] U. Fayyad, and K. Irani, 1993, "Multi-Interval discretization of continuous-valued attributes for classification learning," international joint conference on artificial intelligence, pp. 1022-1027.
- [13] W. C. Hong, 2009, "Electric load forecasting by support vector model," Applied Mathematical Modeling, Vol. 33, pp. 2444-2454.
- [14] W. Li, J. Han, and J. Pei, 2001, "CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rule," International Conference on Data Mining, pp. 369-376.
- [15] X. Li, Z. Deng, and S. Tang, 2006, "A Fast Algorithm for Maintenance of Association Rules in Incremental Databases," Advanced Data Mining and Applications, pp. 56-63.
- [16] 박상승, 손호선, 이동규, 지은미, 김희석, 류근호, 2009, "u-City응용에서의 시간 패턴을 이용한 단기 전력 부하 예측," 한국공간정보시스템학회 논문지, 제11권 제2호, pp. 177-181.
- [17] 박지웅, 홍동숙, 김동오, 한기준, 2006, "시공간 이동 패턴 추출을 위한 효율적인 알고리즘," 한국공간정보시스템학회 논문지, 제8권 제2호, pp. 39-52.

논문접수 : 2011.01.14

수정일 : 1차 2011.03.06 / 2차 2011.04.04

심사완료 : 2011.04.19



이 중 범

2009년 청주대학교 컴퓨터정보공학과
공학사

2009년~현재 충북대학교 대학원 컴퓨터
과학과 공학석사 수료

관심분야는 점진적 빈발패턴 마이닝, 스트림데이터처리,
데이터 마이닝



박 명 호

2007년 연변과학기술대학교 컴퓨터과
학과 공학사

2009년 충북대학교 대학원 컴퓨터과학
과 공학석사

2009년~현재 충북대학교 대학원 컴퓨
터과학과 박사 재학

관심분야는 점진적 빈발패턴 마이닝, 스트림데이터처리,
데이터마이닝, 바이오인포매틱스



류 근 호

1976년 숭실대학교 전산학과 이학사

1980년 연세대학교 공업대학원 전산전
공 공학석사

1988년 연세대학교 대학원 전산전공
공학박사

1976년~1986년 육군 군수 지원사 전산실(ROTC 장
교), 한국전자통신연구원(연구원), 한국방송통신대학교
전산학과(조교수)

1989년~1991년 University of Arizona, Research
Staff (TempIS 연구원, Temporal DB)

1986년~현재 충북대학교 소프트웨어학과 교수

관심분야는 시간 데이터베이스, 시공간 데이터베이스,
지식기반 정보검색 시스템, 유비쿼터스컴퓨팅 및 스트
림데이터처리, 데이터 마이닝, 데이터베이스보안, 바이
오&메디컬인포매틱스