SVM 워크로드 분류기를 통한 자동화된 데이터베이스 워크로드 식별

Automatic Identification of Database Workloads by using SVM Workload Classifier

김소연, 노홍찬, 박상현
연세대학교 컴퓨터과학과

So-Yeon Kim(sykim@cs.yonsei.ac.kr), Hong-Chan Roh(fallsmal@cs.yonsei.ac.kr),
Sang-Hyun Park(sanghyun@cs.yonsei.ac.kr)

요약

데이터베이스 시스템의 응용분야가 데이터웨어하우징에서 전자상거래에 이르기까지 광범위해지면서 데이터베이스 시스템이 대형화되었다. 이로 인해 데이터베이스 시스템의 성능 향상을 위한 튜닝이 중요한 논점이 되었다. 데이터베이스 시스템의 튜닝은 워크로드 특성을 고려하여 수행할 필요가 있다. 그러나 복합적인 데이터베이스 환경에서 워크로드를 식별하기는 어려우므로 자동적인 식별 방법이 요구된다. 본 논문에서는 데이터베이스 워크로드를 자동적으로 식별하는 SVM 워크로드 분류기를 제안한다. TPC-C와 TPC-W 성능 평가에서 자원할당 파라미터 변경에 따른 워크로드 데이터를 수집하여 SVM을 통해 분류한다. SVM의 커널별 커널 파라미터와 오류 허용 임계치 값인 C의 조정을 통하여 최적의 SVM 워크로드 분류기를 선택한다. 제안한 SVM 워크로드 분류기와 Decision Tree, Naïve Bayes, Multilayer Perceptron, K-NN 분류기의 분류 성능을 비교한 결과, SVM 워크로드 분류기가 다른 기계 학습 분류기보다 9% 이상 향상된 분류 성능을 보였다.

■ 중심어 : | 워크로드 분류 | 서포트 벡터 머신 | 데이터베이스 관리 시스템 | 데이터베이스 튜닝 |

Abstract

DBMS is used for a range of applications from data warehousing through on-line transaction processing. As a result of this demand, DBMS has continued to grow in terms of its size. This growth invokes the most important issue of manually tuning the performance of DBMS. The DBMS tuning should be adaptive to the type of the workload put upon it. But, identifying workloads in mixed database applications might be quite difficult. Therefore, a method is necessary for identifying workloads in the mixed database environment. In this paper, we propose a SVM workload classifier to automatically identify a DBMS workload. Database workloads are collected in TPC-C and TPC-W benchmark while changing the resource parameters. Parameters for SVM workload classifier, C and kernel parameter, were chosen experimentally. The experiments revealed that the accuracy of the proposed SVM workload classifier is about 9% higher than that of Decision tree, Naïve Bayes, Multilayer perceptron and K-NN classifier.

■ keyword : | Workload Classification | Support Vector Machine | Database Management System | Database Tuning |

* 본 연구는 2008년 정부(교육과학기술부)의 재원으로 한국연구재단 연구과제로 수행되었습니다.
(KRF-2008-313-D00849)
접수번호 : #091124-004
접수일자 : 2009년 11월 24일
심사완료일 : 2010년 01월 15일
교신저자 : 박상현, e-mail : sanghyun@cs.yonsei.ac.kr

I. 서 론

정보화 사회로 발전하면서 다양하고 복잡한 데이터들이 생겨나게 되었다. 이에 따라 데이터베이스 시스템은 대형화되었고 이러한 대용량 데이터베이스 시스템의 최적화는 매우 중요한 논점이 되었다[1]. 데이터베이스 시스템은 데이터베이스 시스템의 파라미터를 조절하는 튜닝 과정을 통해 최적화 될 수 있다[2]. 데이터베이스 관리자는 적절한 튜닝 과정을 통하여 최적의 데이터베이스 시스템 성능을 유지할 필요가 있다. 데이터베이스 시스템을 효율적으로 튜닝하기 위해서는 데이터베이스 시스템의 자원 사용과 응용 프로그램의 요구사항, 워크로드 특성, 데이터베이스 시스템에 관한 정보가 필요하다[3]. 특히 워크로드의 경우 데이터베이스 시스템의 응용분야가 다양화 되고 복잡해짐에 따라 데이터베이스 관리자가 특성을 식별하기 어려워졌다. 따라서 효과적인 데이터베이스 시스템의 관리를 위해 워크로드를 자동적으로 식별하는 연구가 필요하다.

본 논문에서는 워크로드를 자동적으로 식별하는 SVM 워크로드 분류기를 제안한다. SVM은 기존의 경험적 학습에 의존하는 분류기와는 달리 구조적 최적 분류하는 방법에 근거하기 때문에 학습 패턴의 수와 학습에 필요한 시간 등을 고려하지 않아도 되며 뛰어난 분류 성능을 가진다. 워크로드 데이터 구축을 위해 국제 표준 데이터베이스 성능평가인 TPC-C[4]와 TPC-W[5]을 사용한다. TPC-C는 도매 업체의 재고 관리 시스템을 시뮬레이션하는 OLTP 환경의 워크로드를 제공하고, TPC-W는 인터넷 전자 서점의 전자 상거래 시스템을 시뮬레이션하는 웹 기반 환경의 워크로드를 제공한다. TPC-C와 TPC-W 성능 평가별로 자원할당 파라미터 변경에 따른 15개의 성능 지표 값에 대한 워크로드 데이터를 수집한 후 제안한 SVM 워크로드 분류기를 통해 워크로드를 식별한다. 그리고 제안한 분류기와 다른 기계 학습 분류기와의 분류 성능을 비교한다. 제안한 SVM 워크로드 분류기는 데이터베이스 시스템과 연계되어 작동하고 그 결과는 데이터베이스 시스템 관리자에게 전달된다. 본 연구의 결과는 데이터베이스 튜닝을 비롯하여 자동화된 데이터베이스 시스템 관리에 필요한 워크로드 정보를 정확하게 제공할 수 있다.

2장은 패턴 분류에 적용되었던 기계 학습 알고리즘들을 살펴보고, 기존 연구에 대해서 정리한다. 3장은 제안하는 SVM 워크로드 분류기에 대해 설명한다. 4장은 실험 방법을 설명하고 5장은 제안한 SVM 워크로드 분류기와 다른 기계 학습 분류기와의 분류 성능을 비교한다. 6장은 결론을 맺고 향후 연구 계획을 제시한다.

II. 관련 연구

1. 기계 학습 알고리즘

기계 학습 알고리즘은 패턴 분류를 위해 사용되며 Supervised 기계 학습, Unsupervised 기계 학습, Neural Network 기반의 기계 학습으로 나뉘어진다.

Supervised 기계 학습 알고리즘은 데이터를 기존 학습으로 알게된 그룹으로 분류하는 방법이다. 즉, 분류해야 하는 데이터의 종류를 학습시킨 다음, 새로운 데이터를 학습으로 알게된 그룹 중 하나로 분류하는 것이다. 예로는 Decision Tree, Naïve Bayes, SVM, Hidden Markrov Model, Regression 등이 있다. 특히, SVM은 기계 학습 시스템에서 뛰어난 일반화 성능을 가져 최근 주목받은 알고리즘이다[6-8]. SVM의 분류 정확성(accuracy)과 복잡성(complexity)은 커널 파라미터와 오류 허용 임계치 값인 C의 조정을 통하여 균형을 맞출 수 있다. 따라서 SVM의 성능은 커널 파라미터와 C값을 어떻게 정의하느냐에 달려있다.

Unsupervised 기계 학습 알고리즘은 유사한 데이터들끼리 그룹으로 묶어서 분류하는 방법이다. 새로운 데이터에 대해서 새로운 그룹을 만들어 기존에 그룹으로 묶여지지 않은 데이터에 대해서도 분류가 가능해진다. 예로는 DBSCAN, Expectation Maximization, K-Nearest Neighbor(K-NN), K-Means 등이 있다.

Neural Network 기반의 기계 학습 알고리즘은 생물학에서의 Neural Network를 계층적인 모델로 표현한 방법이다. Neural Network는 입력층, 은닉층, 출력층 방향으로 연결되어진 모델로 데이터의 패턴을 찾아내

는 기능을 한다. 예로는 Radial Base Function (RBF) Network와 Multilayer Perceptron (MLP) 등이 있다.

2. 데이터베이스 워크로드에 대한 기존 연구

데이터베이스 워크로드 특성은 데이터 접근 방식(access method), 자원 할당량(resource usages), 질의 실행 계획(query execution plan) 등에 따라 다양하게 변화될 수 있다[9]. 데이터베이스 관리자는 이러한 데이터베이스 시스템의 워크로드 특성의 변화를 인식하여 데이터 접근 방식, 물리적 구조, 자원 할당량 등을 효율적으로 조율하여 데이터베이스 시스템의 성능을 향상시켜야 한다[10]. 그러나 데이터베이스 응용 분야의 다양화와 복잡화에 따라 워크로드를 식별하는 것은 어려워지고 있다. 이로 인해 워크로드를 자동적으로 식별하려는 연구가 진행되었다.

[11]은 관계형 데이터베이스 시스템에서 트랜잭션 및 쿼리의 워크로드 특징을 분석하는 REDWAR(Relational Database Workload Anaylzer)를 개발하였다. [12]는 전자 상거래 시스템에서 세 개의 응용 분야에 대한 워크로드 특징을 분석하고 QoS 요구사항들을 정립하였고, Quarermaster 시스템을 제시하였다. [13]은 Decision Tree를 이용하여 워크로드를 식별하는 연구를 수행하였다. 총 9개의 성능지표를 이용하여 워크로드 데이터를 수집하였으며 DB2 Intelligent Miner을 이용해 워크로드 모델을 생성하고 워크로드 식별을 수행하였다. 그러나 기존 연구는 데이터베이스 시스템의 자원할당 파라미터가 고정된 상태로 워크로드 데이터를 수집하여 분류하였다는 한 계가 있다. 이로 인해 실제 데이터베이스 시스템에서 발생되는 워크로드와 차이가 존재하여 분류 결과의 정확도가 떨어진다. 또한 [13]에서 사용한 Decision Tree는 연속적인 데이터를 처리하는 능력이 다른 기계 학습 알고리즘에 비해 떨어지고 모형을 구축하는데 사용되는 표본의 크기에 지나치게 민감하다는 문제점을 가지고 있다.

본 논문에서는 기존 연구의 문제를 해결하기 위해 실제 데이터베이스 시스템에서 발생되는 워크로드와 유사한 환경을 제공하는 TPC-C와 TPC-W을 사용한다. TPC-C는 총 9개의 테이블과 5개의 트랜잭션으로 구성되며, 웨어하우스(warehouse) 1개당 10개의 터미널(사용자)이 생성되는 다중 사용자용이다. TPC-W는 8개의 테이블에 대해 14개의 웹 상호작용을 수행하고 가상 브라우저(Emulated Browser)의 수를 통해 시스템 부하를 조절한다. 성능 평가별로 자원할당 파라미터 변경에 따른 워크로드 데이터를 수집한 후 SVM을 이용해 워크로드를 식별하고자 한다.

Ⅲ. 제안하는 SVM 워크로드 분류기

1. 제안하는 SVM 워크로드 분류기의 구조

제안하는 SVM 워크로드 분류기의 구조도를 [그림 1]을 통해 나타내었다. 데이터베이스 시스템과 SVM 워크로드 분류기가 연계되어 작동하고 그 결과를 데이터베이스 시스템 관리자에게 전달하도록 구성되었다. 현재 데이터베이스 시스템의 자원할당 파라미터에 따른 15개의 성능 지표 값에 대한 워크로드 데이터를 SVM 워크로드 분류기를 통해 분류한다. 데이터베이스 시스템 관리자는 분류 결과를 통해 워크로드의 종류가 변경되었으면 데이터 접근 방식, 물리적 구조, 자원 할당량 등을 효과적으로 조율하여 데이터베이스 시스템의 성능을 향상시키도록 한다.

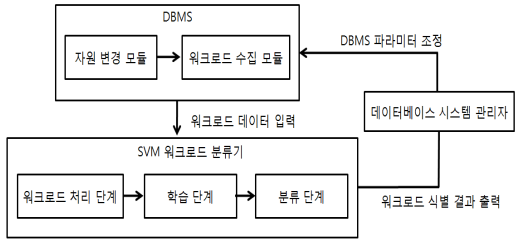

그림 1. 제안하는 SVM 워크로드 분류기의 구조

2. 제안하는 SVM 워크로드 분류기의 작업 흐름도

제안하는 SVM 워크로드 분류기는 [그림 2]에 나타난대로 워크로드 처리 단계와 학습 단계를 거쳐 생성되며, 분류 단계를 통해 실제 워크로드 데이터베이스의 워크로드를 자동으로 식별한다.

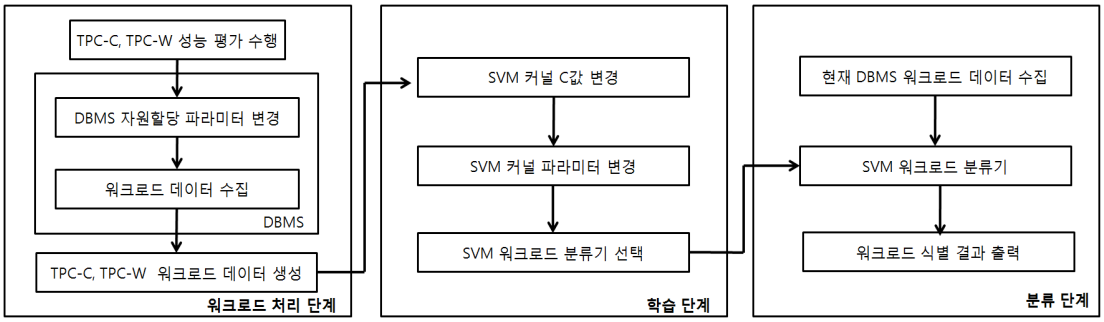

그림 2. 제안하는 SVM 워크로드 분류기의 작업 흐름도

2.1 워크로드 처리 단계

SVM의 학습 데이터로 사용하기 위해 TPC-C와 TPC-W 성능 평가별로 자원할당 파라미터를 변경해가면서 15개의 성능 지표 값에 대한 워크로드 데이터를 수집한다. TPC-C 성능평가에서 수집된 학습 데이터의 클래스는 TPC-C로 정하고, TPC-W 성능평가에서 수집된 학습 데이터의 클래스는 TPC-W로 정한다.

2.2 학습 단계

SVM의 대표적인 4개의 커널들의 커널 파라미터와 오류 허용 임계치 값인 C를 조절해가며 학습 데이터의 분류 정확도를 10-fold 교차 검증을 사용하여 측정한다. 10-fold 교차 검증은 하나의 데이터 set을 10등분하여 그 중 하나를 테스트를 위한 데이터로 사용하고 나머지 9개를 학습을 위한 데이터로 사용한다. 실험 결과를 통해 최적의 파라미터로 설정된 SVM 워크로드 분류기를 선택한다.

2.3 분류 단계

학습 단계에서 생성한 SVM 워크로드 분류기를 사용하여 현재 데이터베이스 시스템의 워크로드 데이터를 자동적으로 식별한다. 식별한 결과를 데이터베이스 관리자에게 전달한다.

Ⅳ. 실험 방법

복합된 데이터베이스 시스템 환경의 워크로드 데이터 구축을 위해 TPC-C와 TPC-W 성능 평가를 사용하였으며, 데이터베이스 시스템으로는 오라클 9i를 사용하였다. 오라클 9i의 스냅샷(snapshot) 기능을 통해 워크로드 데이터를 수집하였다. 성능 평가별로 네 개의 자원할당 파라미터를 증가시키면서 15개의 성능지표 값을 114회에 걸쳐 워크로드 데이터로 수집하였다. 자원할당 파라미터의 변경은 [표 1]과 같다. 15개의 성능 지표의 종류는 데이터베이스 시스템의 작동 시간, 데이터 변경률, 데이터 버퍼 적중률, 공유 메모리 적중률, 메모리 파싱 비율, 시스템 카탈로그 적중률, 메모리 정렬 비율, 래치 경합 비율, 데이터 버퍼 읽기량, 데이터 비버퍼 읽기량, 데이터 버퍼 쓰기량, 데이터 비버퍼 쓰기량, 체크 포인트를 포함한 디스크 쓰기량, 체크 포인트를 포함하지 않은 디스크 쓰기량, redo 로그량이다.

표 1. 자원할당 파라미터 변경

자원할당 파라미터	초기값	증가값	최대값
db_cache_size (데이터 버퍼의 크기)	32MB	32MB	480MB
shared_pool_size (공유 메모리의 크기)	32MB	32MB	480MB
pga_aggregate_target (개인 메모리의 크기)	20MB	20MB	300MB
dbwr_io (I/O 프로세스의 수)	1개	1개	15개

수집한 워크로드 데이터를 SVM을 이용하여 식별하였다. SVM의 커널은 일반적으로 사용되는 선형 커널, PUK 커널, RBF 커널, 다항 커널을 사용하였다. SVM의 커널별로 커널 파라미터들과 오류 임계치 값인 C를

변경하면서 실험하였다. 각 커널에 C값은 0, 0.005, 0.5, 1, 5, 10, 50, 100 으로 동일하게 변경하였고 각 C값에 따른 커널별 커널 파라미터의 변경 내용은 [표 2]와 같다.

표 2. 커널별 커널 파라미터 변경값

커널 종류	커널 파라미터	변경값
선형 커널	없음	c 값만 변경함
PUK 커널	ω	0, 1, 2, 3, 4, 5, 10, 50, 100.
RBF 커널	σ	0, 0.01, 0.05, 1, 5, 10, 15, 25, 50, 100.
다항 커널	d	2, 3, 4, 5, 6, 7.

실험을 통하여 최적의 파라미터로 설정된 SVM 워크로드 분류기를 선택하였고 다른 기계 학습 분류기와 분류 성능을 비교하였다. 분류 성능 평가를 위해 학습 데이터와 테스트 데이터는 10-fold 교차 검증을 이용하여 9:1의 비율로 나누어 사용하고, 방법으로 정확도(accuracy)를 사용하였다(수식 1). 여기서 TP는 true positive, FN은 false negative, TN은 true negative, FP는 false positive를 의미한다[14].

$$\text{Accuracy} = \frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{FN}+\text{TN}+\text{FP})} \times 100 \qquad (1)$$

V. 실험 결과

1. 자원할당 파라미터에 따른 SVM 분류 성능 비교

SVM을 이용해 자원할당 파라미터에 따라 수집한 워크로드 데이터를 분류하였다. SVM의 커널 파라미터와 오류 허용 임계치 값인 C의 조정에 따른 분류 성능 결과를 평가하였다. [그림 3]은 자원할당 파라미터를 고정한 워크로드 데이터를 사용하여 분류한 결과이다. 각 커널의 파라미터를 최적으로 설정하였을 때, C값에 따른 SVM의 정확도를 나타낸다. 실험 결과, RBF 커널 파라미터가 σ=1, C=50으로 설정되었을 때 78.80%의 최고 정확도를 보였다. [그림 4]는 자원할당 파라미터의 변경에 따른 워크로드 데이터를 사용하여 분류한 결과이다. 각 커널의 파라미터를 최적으로 설정하였을 때, C값에 따른 SVM의 정확도를 나타낸다. 실험 결과, RBF 커널 파라미터가 σ=5, C=1로 설정되었을 때 86.63%의 최고 정확도를 보였다. 자원할당 파라미터의 변경에 따른 워크로드 데이터를 사용하여 분류하였을 때 8%이상 정확하게 워크로드를 식별함을 볼 수 있었다. 본 실험에서 커널 파라미터 σ=5, C=1로 설정한 RBF 커널을 사용하였을 때, 가장 높은 정확도를 보였으므로 이를 최적의 워크로드 분류기로 선택하였다.

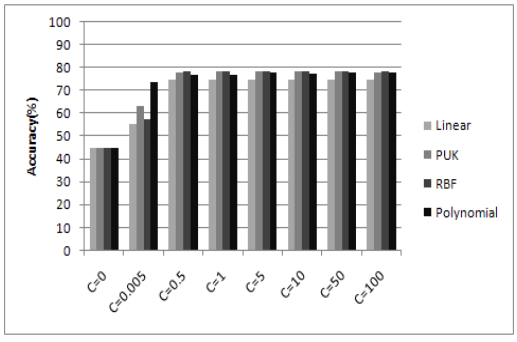

그림 3. 자원할당 파라미터를 고려하지 않았을 때 C 값에 따른 커널별 최고 SVM 정확도

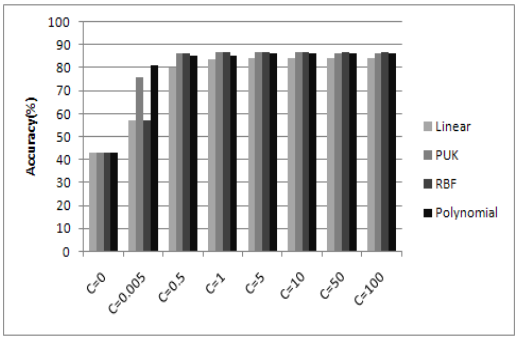

그림 4. 자원할당 파라미터를 고려했을 때 C 값에 따른 커널별 최고 SVM 정확도

2. 기계 학습 분류기별 분류 성능 비교

제안한 SVM 워크로드 분류기와 다른 기계 학습 분류기와 분류 성능을 비교하였다. SVM 워크로드 분류기와 마찬가지로 다른 기계 학습 분류기도 최적의 파라미터로 설정하여 실험하였다. [그림 5]의 결과를 살펴보

면 Decision Tree, Naïve Bayes, SVM, K-NN, MLP 워크로드 분류기는 각각 71.63%, 68.88%, 86.61%, 73.34%, 75.84%의 정확도를 보였다. 실험을 통하여 최적의 파라미터로 설정된 SVM 워크로드 분류기가 다른 기계 학습 분류기보다 9% 이상 정확하게 워크로드를 식별함을 볼 수 있었다.

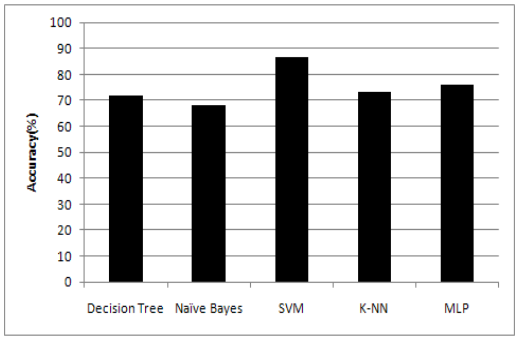

그림 5. 기계 학습 분류기별 분류 성능 비교

Ⅵ. 결 론

본 논문에서는 데이터베이스 워크로드 식별을 위한 SVM 워크로드 분류기를 제안하였다. TPC-C와 TPC-W 성능 평가를 이용하여 자원할당 파라미터 변경에 따른 15개의 성능지표 값에 대한 워크로드 데이터를 수집하여 워크로드를 식별하였다. 최적의 파라미터로 설정된 SVM 워크로드 분류기를 실험을 통해 선택하였다. 이 분류기는 자원할당 파라미터를 고려하지 않은 분류기보다 8% 이상 정확하게 워크로드를 식별하였으며, Decision Tree, Naïve Bayes, K-NN, MLP 분류기보다 9% 이상 정확하게 워크로드를 식별하였다. 제안한 SVM 워크로드 분류기를 통해 얻을 수 있는 기존 연구와의 차별성은 다음과 같다.

첫째, 워크로드 분류기에 SVM을 적용하여 분류 정확도를 향상시켰다. 둘째, 자원할당 파라미터를 고려한 워크로드 데이터를 사용하여 SVM 워크로드 분류기를 선택하였다. 이로 인해 다양한 데이터베이스 시스템 환경에서도 보다 정확하게 워크로드 식별하였다.

향후 연구로는 워크로드를 식별한 결과를 바탕으로 워크로드 종류에 따른 데이터베이스 시스템의 튜닝에 관해 연구할 예정이다.

참 고 문 헌

[1] James Martin, "*Information Engineering Book II*", Prentice Hall Pub, 1990.

[2] Oracle 8 : Database Administration, 성능튜닝 워크숍, SQL 튜닝, ORACLE, 1998.

[3] S. Chaudhuri and V. Narasayya, "AutoAdmin "What-if" index analysis utility", Proceedings of the 1998 ACM SIGMOD international conference on Management of data, pp.367-378, 1998.

[4] http://www.tpc.org/tpcc/default.asp

[5] http://www.tpc.org/tpcw/default.asp

[6] A. Ben-Hur, D. Horn, H. T. Siegelmann and V. Vapnik, "Support Vector Clustering," The Journal of Machine Learning Research, Vol.2, pp.125-137, 2002.

[7] J. Lee and D. Lee, "An Improved Cluster Labeling Method for Support Vector Clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.27, No.3, pp.461-464, 2005.

[8] B. Y. Sun and D. S. Huang, "Support Vector Clustering for Multiclass Classification Problems," IEEE Evolutionary Computation Congress, Vol.2, pp.1480-1485, 2003.

[9] A. Aboylnaga and S. Chaudhuri, "Self-tuning histograms : building histograms without looking at data," Proceedings of the 1999 ACM SIGMOD international conference on Management of data, pp.181-192, 1999.

[10] S. Chaudhuri and G. Weikum, "Rethinking Database System Architecture : Towards a

Self-Tuning RISC-Style Database System", Proceedings of the 26th International Conference on Very Large Databases, pp.1-10, 2000.

[11] P. S. Yu, M. S. Chen, H. U. Heriss and S. Lee, "One Workload Characterization of Relational Database Environments," IEEE Transacion on Software, Vol.18, No.4, pp.347-355, 1992.

[12] P. Martin, H. Y. Li, M. Zheng, K. Romanufa, and W. Powley, "Dynamic Reconfiguration Algorithm : Dynamically Tuning Multiple Buffer Pools," Proceedings of the 11th International Conference on Database and Expert Systems Applications, pp.92-101, 2000.

[13] S. Elnaffar, "A Methodology for Auto-Recognizing DBMS Workloads," Proceedings of the 2002 conference of the Centre for Advanced Studies on Collaborative research, 2002.

[14] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," Proceedings of International Joint Conference on Artificial Intelligence, pp.1137-1143, 1995.

저 자 소 개

김 소 연(So-Yeon Kim) 준회원

- 2008년 2월 : 숭실대학교 컴퓨터학부(공학사)
- 2008년 9월 ~ 현재 : 연세대학교 컴퓨터과학과(공학석사 과정)

<관심분야> : 고성능 데이터베이스, 데이터 마이닝, 데이터베이스 튜닝

노 홍 찬(Hong-Chan Roh) 정회원

- 2006년 2월 : 연세대학교 컴퓨터과학부(공학사)
- 2008년 2월 : 연세대학교 컴퓨터과학과(공학석사)
- 2008년 3월 ~ 현재 : 연세대학교 컴퓨터과학과(공학박사 과정)

<관심분야> : 플래쉬메모리 인덱스, SSD, 데이터마이닝

박 상 현(Sang-Hyun Park) 정회원

- 1989년 2월 : 서울대학교 컴퓨터공학과(공학사)
- 1991년 2월 : 서울대학교 컴퓨터공학과(공학석사)
- 2001년 2월 : UCLA대학교 전산학과(공학박사)
- 1991년 3월 ~ 1996년 8월 : 대우통신 연구원
- 2001년 2월 ~ 2002년 6월 : IBM T. J. Watson Research Center Post-Doctoral Fellow
- 2002년 8월 ~ 2003년 8월 : 포항공과대학교 컴퓨터공학과 조교수
- 2003년 9월 ~ 2006년 8월 : 연세대학교 컴퓨터과학과 조교수
- 2006년 9월 ~ 현재 : 연세대학교 컴퓨터과학과 부교수

<관심분야> : 데이터베이스, 데이터 마이닝, 바이오인포매틱스, 적응적 저장장치 시스템