PCA-SVM 분류기를 이용한 데이터베이스 워크로드의 다중 클래스 분류

(Multi-class Classification of Database Workloads using PCA-SVM Classifier)

김 소 연 † 박 상 현 ††
(Soyeon Kim) (Sanghyun Park)

요 약 정보산업 사회가 되면서 생겨난 대용량의 데이터로 인해 기업들은 데이터베이스 시스템을 필수적으로 활용하고 있다. 데이터베이스 시스템 관리자는 효과적인 데이터베이스 시스템의 활용을 위해서 워크로드의 정보를 필요로 한다. 그러나 다양화되고 복잡해지는 데이터베이스 응용 분야로 인해 관리자가 데이터베이스 시스템에서 발생되는 워크로드를 식별하기 힘들어졌다. 따라서 복합적인 데이터베이스 응용 분야에서 워크로드를 자동적으로 식별하는 방법이 요구된다. 본 논문에서는 데이터베이스 워크로드를 자동적으로 식별하는 PCA-SVM 워크로드 분류기를 제안한다. TPC-C와 TPC-H 성능평가의 수행 비율별로 자원할당 파라미터 변경에 따른 워크로드 데이터를 수집한다. PCA(Principal Components Analysis)을 적용하여 워크로드 데이터의 특징 벡터의 차원을 축소시키고 다중 클래스 SVM(Support Vector Machine)의 일대다(one-against-all) 기법을 이용하여 워크로드를 분류한다. SVM의 커널별 커널 파라미터와 오류 허용 임계치 값인 C의 조정을 통하여 최적의 PCA-SVM 워크로드 분류기를 선택한다. 실험 결과, PCA-SVM 워크로드 분류기는 특징 벡터의 차원을 2/5로 축소시키면서도 다른 분류기보다 7%이상 정확하게 워크로드를 식별하였다. 또한, 분류 시간은 특징 벡터의 차원을 축소시키기 이전과 비교하여 약 1/18로 단축되어 향상된 분류 성능을 보였다.

키워드 : 워크로드 분류, 주성분 분석, 다중 클래스 서포트 벡터 머신, 데이터베이스 관리 시스템, 데이터베이스 튜닝

Abstract A lot of companies have essentially exploited Database Management System (DBMS) to process huge amounts of data due to emerging of the information industry. Database administrators need the information of workload in order to maintain high performance DBMS. However, it has been hard to identify workload due to being diversified and complicated of database application. Therefore, the method which can automatically identify workload is required in these environments. In this paper, we propose PCA-SVM workload classifier for identifying DBMS workloads automatically. For achieving this, we collect workload data according to performance ratio while changing the resource parameters. We reduce the dimension of the feature vectors existing in the workload data by Principal Components Analysis (PCA) and classify the workload by one-against-all approach of multi-class Support Vector Machine (SVM). We experimentally select an optimal PCA-SVM workload classifier by adjusting kernel parameters for each kernel and error-tolerance threshold, C. Experimental results show that the proposed PCA-SVM workload classifier reduces dimension of the feature vector by a factor of 2/5, and its accuracy is about 7% higher than other classifiers. Moreover, the computation

· 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2010-0010689)

† 비 회 원 : 연세대학교 컴퓨터과학과
sykim@cs.yonsei.ac.kr
†† 종신회원 : 연세대학교 컴퓨터과학과 교수
sanghyun@cs.yonsei.ac.kr
(Corresponding author임)

논문접수 : 2010년 3월 8일
심사완료 : 2010년 10월 11일

정보과학회논문지: 데이타베이스 제38권 제1호(2011.2)

time for classification is also improved as much as 18 times compared with the one without dimensionality reduction.

Key words : Workload classification, PCA(Principal Components Analysis), Multi-class Support Vector Machine, Database management system, Database tuning

1. 서 론

정보산업 사회에서 정확하고 신속한 정보의 제공은 기업의 경쟁력 강화를 위해 더욱더 중요하게 되었다.

이에 따라 대부분의 기업은 데이터베이스 시스템의 활용을 필수 요소로 인식하고 있다. 데이터베이스 시스템을 효율적으로 활용하기 위해서 데이터베이스 시스템 관리자는 데이터베이스 시스템의 자원 사용과 응용 프로그램의 요구 사항, 워크로드 특성, 데이터베이스 시스템 정보를 필요로 한다[1]. 워크로드 특성은 데이터베이스 응용 분야에 따라 다르며, 데이터베이스 환경에서 하나 이상의 응용 분야가 수행될 수 있다. 다양화되고 복잡해지는 데이터베이스 응용 분야에 따라 데이터베이스 시스템 관리자가 워크로드 특성을 식별하기 힘들어졌다. 따라서 OLTP, DSS, 웹 전자 상거래 형태의 워크로드가 교차되어 수행되는 복합된 데이터베이스 응용 분야에서 데이터베이스 시스템의 효율적인 활용을 위해 워크로드를 자동적으로 분류하는 연구가 필요하다. 또한, 다양한 워크로드 종류로 인해 생기는 다중 분류 문제의 해결이 필요하다.

이러한 필요성으로 인해 선행 연구[2]에서 우리는 TPC-C와 TPC-W 성능 평가를 이용하여 자원할당 파라미터 변경에 따른 15개의 성능지표 값에 대한 워크로드 데이터를 수집한 후 SVM(Support Vector Machine) 워크로드 분류기를 이용해 워크로드를 식별한 바 있다. 그러나 이 방법은 복합된 데이터베이스 응용 분야에서의 워크로드에 대한 식별이 아닐 뿐만 아니라 워크로드 분류 시간이 많이 소요되는 문제점을 가지고 있기 때문에 현실적으로 실무에 적용할 수 있는 워크로드 분류기와는 여전히 거리가 멀다.

본 논문에서는 복합된 데이터베이스 응용 분야에서의 다중 클래스 워크로드 분류문제를 해결하고, 향상된 분류 성능을 통해 실무에 적용할 수 있는 PCA-SVM 워크로드 분류기를 제안한다. 워크로드 데이터 구축을 위해서 국제 표준 데이터베이스 성능 평가인 TPC-C[3]와 TPC-H[4]를 사용한다. TPC-C는 도매 업체의 재고 관리 시스템을 시뮬레이션하는 OLTP(OnLine Transaction Processing) 환경의 워크로드를 제공하고, TPC-H는 복잡한 질의를 실행하며 실제 비즈니스 상황을 시뮬레이션하는 DSS(Decision Support System) 환경의 워크로드를 제공한다. TPC-C와 TPC-H 성능평가의 수행 비율별로 자원할당 파라미터 변경에 따른 15개의 성능 지표값에 대한 워크로드 데이터를 수집한다.

효율적으로 고차원의 워크로드 데이터를 분류하기 위해서 특징 벡터의 차원을 저차원으로 축소시키는 PCA (Principal Components Analysis)을 적용한다. PCA에 의해 축소된 워크로드 데이터를 다중 클래스 SVM의 one-against-all 기법을 이용하여 분류한다.

본 논문의 구성은 다음과 같다. 2장에서는 데이터베이스 워크로드에 대한 관련 연구와 본 논문의 차별성에 대해 기술한다. 3장에서는 제안하는 PCA-SVM 워크로드 분류기에 대해 기술한다. 4장에서는 제안한 SVM 워크로드 분류기와 다른 기계 학습 분류기와의 분류 성능을 비교한다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해 논한다.

2. 관련 연구

2.1 데이터베이스 워크로드 분석에 대한 관련 연구

데이터베이스 워크로드에 대한 연구는 워크로드 분석에서 분류에 이르기까지 많은 연구가 진행되었다. [5]는 DB2 관계형 데이터베이스 시스템에서 SQL 구문의 구조와 복잡도 분석을 통해 트랜잭션 및 쿼리의 run-time 워크로드 특징을 분석하는 REDWAR(Relational Database Workload Anaylzer)를 개발하였다. [6]은 전자 상거래 시스템에서 세 개의 응용 분야를 분류하고, 각각의 응용 분야의 워크로드 특징들에서 QoS 요구 사항들을 정립하여 Quarermaster 시스템을 개발하였다. [7]은 기업에서 분석된 데이터베이스 워크로드의 특성을 TPC-C와 TPC-D 성능평가들과 비교하였다. 데이터베이스 워크로드 특성은 트랜잭션들의 특징, 동시성의 정도(degree of concurrency), 객체들의 특징, I/O 강도(intensity)와 폭발성(burstiness)을 중심으로 비교되었다. [8]은 인터넷 기반의 전자상거래에 환경에서 데이터베이스 서버에서 발생되는 데이터베이스 워크로드를 분석하기 위해 TPC-W 성능평가에서 워크로드를 수집하여 워크로드 특성을 분석하였다. 질의 결과 캐시, 테이블 캐시, 하이브리드 캐시로 분류될 수 있는 동적 캐시가 워크로드 특성에 의해 받는 영향을 측정하였다. [9]는 동시성 제어 모델을 데이터베이스 모델, 트랜잭션 모델, 사용자 모델, 시스템 모델로 분류하여 생성한 후, 실제 데이터베이스의 워크로드 분석을 수행하였다.

2.2 데이터베이스 워크로드 분류에 대한 관련 연구

선행연구[2]에서 우리는 TPC-C와 TPC-W 성능 평가를 이용하여 자원할당 파라미터 변경에 따른 15개의 성능지표 값에 대한 워크로드 데이터를 수집한 후 SVM 워크로드 분류기를 이용해 워크로드를 식별하였다. [10]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. [11]은 데이터베이스 시스템을 지속적으로 관찰하면서 워크로드 종류의 변화를 탐지하는 Psychic-Skeptic Prediction(PSP) Framework을 사용한 분류기를 개발하였다. 특정시간에 PSP Framework을 사용하여 DSSness을 기준으로 3가지 구역으로 나누어 워크로드를 선택적으로 관찰함으로써 워크로드의 주요 변화를 탐지하였다. [12]는 SQL 구문의 다양성으로 인해 생기는 지속적인 워크로드 분석의 오버헤드를 감소하기 위한 클러스터링 기법을 개발하였다. 즉, 비슷한 워크로드 이벤트들을 클래스로 그룹화하는 거리 함수를 사용한 워크로드 분류를 통해 워크로드 분석의 오버헤드를 감소시켰다.

2.3 기존 연구와의 차별성

기존 연구에서는 주로 단일 데이터베이스 응용 분야(OLTP, DSS, 웹 전자 상거래)의 데이터베이스 시스템 환경에서의 워크로드에 대해 분석해왔다. 단일 응용 분야의 워크로드 구성 분포나 데이터 버퍼 같은 국부적인 자원에 대해 소수의 성능 지표를 이용해서 연구를 진행하였다. 그리고 동일한 응용 분야에서 자원할당 파라미터 크기가 변경됨에 따라 성능 지표값이 달라지는 것을 고려하지 않았다.

그러나 데이터베이스 시스템은 많은 프로세스가 상호적으로 연관되어 수행되기 때문에 소수의 자원과 성능 지표의 분석만으로는 워크로드를 분류하기 어려울 뿐만 아니라, 자원할당 파라미터가 고정된 상태의 워크로드와 실제 데이터베이스 시스템에서 발생되는 워크로드는 차이가 존재한다는 한계점을 가진다.

[2]와 [10]에서는 두 개의 응용 분야에서 수집된 워크로드를 이용해 분류를 하지만 복합된 워크로드에 대한 식별이 아니어서 복합된 응용 분야가 실행되는 환경에서 사용하기는 부적합하다. [2]에서 워크로드 분류에 사용한 SVM은 다른 기계 학습 알고리즘과 비교했을 때 우수한 분류 성능을 보였다. 그러나 이진 분류기라는 기능적 한계점으로 인하여 현재 해결하고자 하는 복합된 데이터베이스 응용 분야에서의 워크로드 분류와 같은 다중 분류 문제에는 SVM을 직접적으로 적용할 수 없다. 그리고 [10]에서 워크로드 분류에 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 복합된 데이터베이스 응용분야에서 발생되는 워크로드를 식별하고자 한다. TPC-C와 TPC-H 성능 평가의 수행 비율별로 다수의 자원할당 파라미터 크기 변경에 따른 워크로드를 수집하여 실제 데이터베이스 시스템에서 발생되는 워크로드와의 차이를 줄이도록 한다. 수집된 모든 워크로드 데이터를 분류에 이용할 경우 특징 벡터의 차원이 커지게 되어 분류 성능 및 속도의 저하를 가져오게 된다. 따라서 PCA를 사용해 워크로드 데이터의 주성분을 분석하여 축소된 특징 벡터를 추출한다. PCA로 축소된 데이터를 다수의 SVM을 유기적으로 결합시킨 다중 클래스 SVM을 사용하여 분류한다. PCA-SVM 워크로드 분류기를 이용해 분류 효율성을 높이고 다중 분류 문제를 해결하고자 한다.

3. 제안하는 PCA-SVM 워크로드 분류기

3.1 제안하는 PCA-SVM 워크로드 분류기의 구조

제안하는 PCA-SVM 워크로드 분류기의 구조도는 그림 1과 같다. 실제 데이터베이스 시스템과 제안된 PCA-SVM 워크로드 분류기가 연계되어 작동되고 워크로드 분류 결과가 데이터베이스 관리자에게 전달된다. 데이터베이스 관리자는 분류 결과를 통해 워크로드의 종류가 변경되었으면 데이터 접근 방식, 물리적 구조, 자원 할당량 등을 효과적으로 조율하여 데이터베이스 시스템의 성능을 향상시키도록 한다.

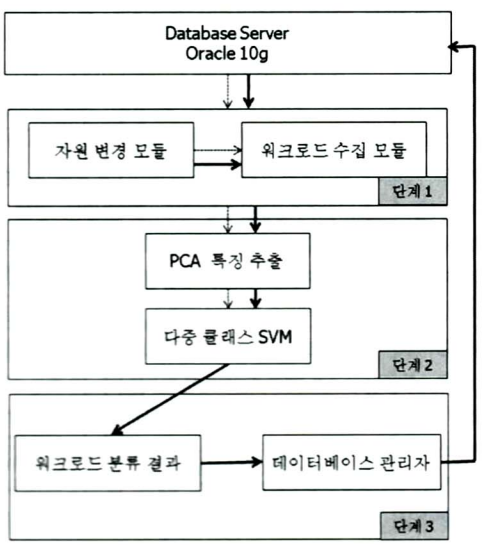

그림 1 PCA-SVM 워크로드 분류기의 구조

3.2 제안하는 PCA-SVM 워크로드 분류기의 작업 흐름도

제안하는 PCA-SVM 워크로드 분류기는 그림 2처럼 워크로드 처리 단계와 학습 단계를 거쳐 생성되고 분류 단계에서 실제 워크로드를 식별한다.

단계 1. 워크로드 처리 단계

복합된 응용 분야의 워크로드 데이터를 수집하기 위해 TPC-C와 TPC-H 성능평가를 수행시킨다. 성능평가의 수행 비율별로 4개의 자원할당 파라미터를 변경해가면서 15개의 성능 지표값에 대한 워크로드 데이터를 수집한다. 혼합된 워크로드 클래스는 6개로 TPCC_100, MIX 80_20, MIX 60_40, MIX 40_20, MIX 20_80, TPCH_100으로 정한다.

단계 2. 학습을 통한 PCA-SVM 워크로드 분류기 생성 단계

(1) PCA을 이용한 특징 추출

PCA는 다양한 분야의 패턴 인식에서 대표적으로 차원 축소를 위해 이용되는 다변량 통계 분석 방법이다. 주어진 데이터를 분산이 최대가 되는 축으로 변환하는 것으로, 이 새로운 차원에서의 데이터의 벡터들을 주성분(Principal Components)이라고 한다. 이 때 분산이 작은 성분을 제거함으로써 데이터의 차원을 줄이는 동시에 데이터에 포함되어 있던 잡음(noise)를 제거할 수 있다[13].

따라서 PCA을 이용해 워크로드 데이터의 특징 벡터가 표현 하고 있던 분산을 최대화 하는 방향으로 특징 공간을 선형 사영하여 차원을 줄인다. 분산을 이용하여 고유값과 특징 벡터를 구한다. 고유값들의 크기순으로 나열하여 이에 해당하는 원하는 차원(k차원)의 특징 벡터를 추출한다. 누적기여율(누적 분산)이 전체 고유값 합의 99%를 차지하는 k개를 선택한다.

(2) 다중 클래스 SVM 설계

PCA에 의해 저차원으로 축소된 데이터를 다중 클래스 SVM의 학습 데이터로 사용한다. SVM은 구조적 최적 분류를 보장하여 뛰어난 일반화 성능을 보여주는 하나의 학습이론으로써 유연하고 강력한 알고리즘이다. SVM은 비선형 매핑함수를 이용하여 학습 데이터의 공간을 선형 초평면(Hyperplane)이 만들어지는 고차원 특징 공간으로 매핑하고, 인식 오류를 최소화하는 최적 초평면을 찾는다. 이 초평면과 가장 가까운 입력샘플벡터를 지지벡터(Support vector)라고 하며 서로간의 거리가 최대가 되도록 최적화된다. 이 때 커널함수는 비선형 매핑함수의 내적함수의 내적계산을 함수 형식으로 치환하여 간단하게 해준다.

데이터베이스 워크로드 분류가 다중 클래스 분류이므로 one-against-all 기법을 적용하여 SVM을 설계한다. 그림 3처럼 각 클래스마다 이진 분류가 가능한 SVM을 생성하여, 총 6개의 SVM을 얻는다. 예를 들어, TPCC_100 클래스인 특징 벡터를 분류하기 위한 SVM은 학습 데이터에서 TPCC_100 클래스인 특징 벡터에 대해 클래스 값을 +1, 나머지 5개의 클래스에 속한 특징 벡터에 대해 클래스 값을 -1로 첨부한 뒤, 이들을 입력으로 하여 SVM을 생성한다. 이와 같은 방식으로 TPCC_100, MIX80_20, MIX60_40, MIX40_60, MIX20_80, TPCH_100 클래스 각각에 대한 SVM을 모두 생성한다. 이후 실험 데이터에 속한 특징 벡터를 6개의 SVM에 모두 입력하여, +1 또는 -1 값이 아닌 실수 값을 출력한다. 이 중에서 출력 값이 가장 큰 SVM에 해당하는 클래스로 특징 벡터를 분류한다.

SVM의 분류 정확성(accuracy)과 복잡성(complexity)

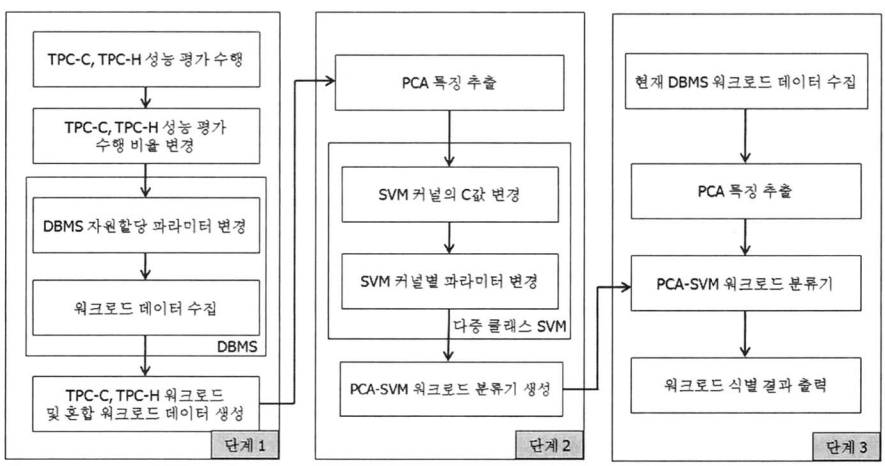

그림 2 제안하는 PCA-SVM 워크로드 분류기의 작업 흐름도

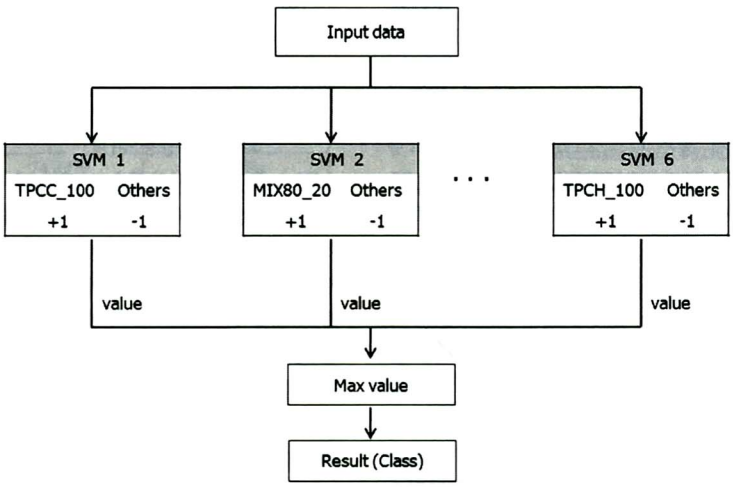

그림 3 one-against-all 기법을 적용한 다중 클래스 SVM 구조

표 1 커널 파라미터 변경값

커널 종류	커널 파라미터	변경값
선형 커널	없음	C 값만 변경함.
다항 커널	d	2, 3, 4, 5, 6, 7, 8, 9, 10.
RBF 커널	σ	1, 5, 10, 15, 25, 50, 100.
PUK 커널	ω	1, 5, 10, 15, 25, 50, 100.

은 커널 파라미터와 오류 허용 임계치 값인 C의 조정을 통하여 균형을 맞출 수 있다[14-16]. 이로 인해 SVM의 분류 성능은 커널 파라미터와 C값을 어떻게 정의하느냐에 달려 있어 최적의 파라미터를 찾아야 한다.

SVM의 커널은 일반적으로 사용되는 선형 커널, 다항 커널, RBF 커널, PUK 커널을 사용한다. 각 커널의 C 값은 1, 5, 10, 15, 25, 50, 100으로 동일하게 변경하고 각 C값에 따른 커널별 커널 파라미터의 변경 내용은 표 1과 같다.

(3) PCA-SVM 워크로드 분류기 생성

PCA를 통해 워크로드 데이터의 특징 벡터를 추출하여 one-against-all 기법을 적용한 다중 클래스 SVM을 설계하여 실험한다. SVM의 대표적인 4개의 커널의 커널 파라미터와 오류 허용 임계치 값인 C를 조절해가며 분류 정확도와 분류 시간을 10-fold 교차 검증을 사용하여 측정한다. 실험 결과를 통해 최적의 파라미터로 설정된 PCA-SVM 워크로드 분류기를 생성한다.

단계 3. 실제 워크로드 분류 단계

실제 워크로드 분류시, 학습 단계에서 생성된 PCA-SVM 워크로드 분류기에 워크로드 데이터를 넣어 자동적으로 분류한다. 식별한 결과를 데이터베이스 관리자에게 전달한다.

4. 실험 및 결과 분석

4.1 실험 데이터

복합된 데이터베이스 시스템 환경의 워크로드 구축을 위해 표준 국제 성능평가 도구인 TPC-C와 TPC-H를 사용하였다.

TPC-C는 도매업체의 재고 관리 시스템을 시뮬레이션하고 웨어하우스(warehouse) 1개당 10개의 터미널(사용자)이 생성되는 다중 사용자용이다. 웨어하우스 수는 100개로 설정하였다. TPC-H는 대용량 데이터베이스에 대한 복잡한 질의를 실행하며 실제 비즈니스 상황을 시뮬레이션한다. 데이터베이스 크기는 1GB으로 설정하였다.

데이터베이스 시스템은 오라클 10g를 사용하고 snapshot 기능을 이용한다. TPC-C와 TPC-H을 동시에 수행시키며 소비된 CPU시간을 기준으로 워크로드를 혼합시켰다. TPC-C와 TPC-H의 워크로드 혼합 비율이 100%와 0%, 80%와 20%, 60%와 40%, 40%와 20%, 20%와 80%, 0%와 100%로 구성되도록 수행하였다. 각 수행 비율별로 자원할당 파라미터를 변경해가면서 15개의 성능 지표값에 대한 워크로드 데이터를 수집하였다.

자원할당 파라미터의 종류와 변경크기는 표 2와 같다. DB_CACHE_SIZE는 버퍼 캐시의 크기를 설정하고, SHARED_POOL_SIZE는 공유 커서, 내장 프로시저, SQL 문장의 파싱 등에 사용되는 메모리의 크기를 설정한다. PGA_AGGREGATE_TARGET는 정렬, 해시 조인 등 메모리를 집중적으로 사용하는 질의에 대해 메모리 크기를 설정하고, LOG_BUFFER는 데이터베이스 변경에 관한 정보를 보관 유지하는 버퍼의 크기를 설정한다.

15개의 성능지표의 종류는 데이터베이스 시스템의 작동 시간, 데이터 변경률, 데이터 버퍼 적중률, 공유 메모

리 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. 워크로드 클래스는 6개로 TPCC_100, MIX 80_20, MIX 60_40, MIX 40_20, MIX 20_80, TPCH_100으로 정하였다.

워크로드 데이터는 네 개의 자원할당 파라미터(데이터 버퍼, 공유 메모리, 개인 메모리, 로그 버퍼)을 증가시키면서 총 4320회(성능평가 수(6)×변경된 자원할당 파라미터 수(15)×파라미터 종류 수(4)×시간당 횟수(12 : 5분단위로 한시간동안 측정))를 수행시켜 수집하였다. 수집된 워크로드 데이터는 20차원으로 4개의 자원할당 파라미터, 15개의 성능지표, 클래스를 특징으로 가지며 학습 데이터와 실험 데이터는 10-fold 교차 검증을 이용하여 9:1의 비율로 나누어 사용한다.

표 2 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
log_buffer	20MB	20MB	300MB

4.2 실험 결과 및 분석

4.2.1 PCA을 이용한 특징 벡터의 차원 축소 결과

PCA을 통해 특징 벡터들 모임의 분산을 최대화하는 방향으로 특징 공간을 선형 사영하여 특징 공간의 차원을 줄였다. 클래스에 가장 중요한 정보가 분산이 가장 큰 방향을 따라 포함한다는 가정에 따라, 누적기여율(누적 분산)이 전체 고유값 합의 99%를 차지하는 개수 k를 선택하였다.

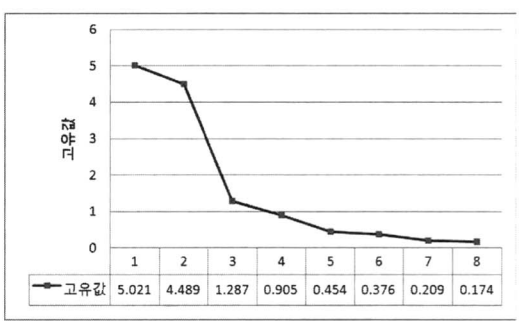

그림 4 상위크기순으로 정렬된 고유값 곡선

그림 4는 상위크기순으로 정렬된 고유값 곡선이다. 고유값이 큰 것부터 8개(k=8)를 선정, 이에 대응하는 특징 벡터로 구성된 8차원의 특징 공간으로 축소시켰다.

4.2.2 커널 함수에 따른 PCA-SVM 워크로드 분류기의 분류 성능 비교

PCA에 의해 저차원으로 축소된 데이터를 다중 클래스 SVM을 이용해 분류하였다. 학습 데이터와 실험 데이터는 10-fold 교차 검증을 이용하여 9:1의 비율로 나누어 사용하고, 분류 성능 평가를 위해 Accuracy(식 (1)), 분류 시간(seconds)를 측정하였다.

Accuracy는 전체 데이터를 하나로 놓고 제대로 분류가 된 양이 얼마나 되는지에 대해 알아보기 위한 평가기준이다[17]. 여기서 TP는 true positive, FN은 false negative, TN은 true negative, FP는 false positive를 의미한다.

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

표 3은 각 커널별로 C값 및 커널 파라미터를 최적으로 설정하였을 때의 분류 결과이다. 본 실험 결과, C=1, σ=5로 설정한 RBF 커널을 사용하였을 때, 91.22%의 가장 높은 정확도를 보였고 분류 시간은 5.42s로 다항 커널과 PUK 커널에 비해 빨랐다. 따라서 이를 최적의 워크로드 분류기로 선택하였다.

표 3 커널별 PCA-SVM 워크로드 분류기의 분류 성능 비교

커널 종류	C값 및 커널 파라미터	정확도 (%)	분류 시간 (seconds)
선형 커널	C=1	79.91%	4.59s
다항 커널	C=5, d=3	86.21%	7.48s
RBF 커널	C=1, σ=5	91.22%	5.42s
PUK 커널	C=5, ω=10	84.57%	10.97s

4.2.3 기계 학습 분류기별 분류 성능 비교

제안한 PCA-SVM 워크로드 분류기와 SVM, Decision Tree, K-NN(K-Nearest Neighbor), MLP(Multi-Layer Perceptron) 워크로드 분류기의 분류 성능을 비교하였다. PCA-SVM 워크로드 분류기와 마찬가지로 다른 기계 학습 분류기도 최적의 파라미터로 설정하여 실험하였다.

표 4의 결과를 살펴보면 PCA-SVM, SVM, Decision Tree, K-NN, MLP 워크로드 분류기는 각각 91.22%, 84.38%, 69.63%, 74.34%, 77.38%의 정확도를 보였다. 제안한 PCA-SVM 워크로드 분류기는 다른 기계 학습 분류기보다 7% 이상 정확하게 워크로드를 식별하였다. 그리고 제안한 PCA-SVM 워크로드 분류기와 가장 분류 시간이 짧았던 다른 기계 학습 분류기의 분류 시간을 비교했을 때 98.26s에서 5.42s로 감소하여 약 1/18로 단축되었다.

표 4 기계 학습 분류기별 분류 성능 비교

분류기	정확도 (%)	분류 시간 (seconds)
PCA-SVM	91.22%	5.42s
SVM	84.38%	121.01s
Decision Tree	69.63%	98.26s
K-NN	74.34%	135.03s
MLP	77.38%	709.83s

5. 결론 및 향후 과제

본 논문에서는 데이터베이스 워크로드를 식별하는 PCA-SVM 워크로드 분류기를 제안하였다. TPC-C와 TPC-H 성능 평가의 수행 비율별로 자원할당 파라미터 변경을 고려한 워크로드 데이터를 수집하여 분류하였다. PCA을 이용하여 차원이 축소된 특징 벡터를 생성한 뒤 one-against-all 기법으로 다중 클래스 SVM을 적용하였다.

PCA-SVM 워크로드 분류기를 통해 특징벡터의 차원을 20차원에서 8차원으로 2/5로 축소시키면서도 다른 분류기보다 7%이상 정확하게 워크로드를 식별하였다. 분류 시간은 특징 벡터의 차원을 축소시키기 이전과 비교하여 98.26s에서 5.42s로 감소함으로써 약 1/18로 단축되었다.

제안된 PCA-SVM 워크로드 분류기를 통해 다양하고 복잡한 데이터베이스 시스템 환경에도 보다 정확하게 워크로드를 식별하도록 하였다. 또한, 데이터베이스 워크로드 분류 성능을 탁월하게 향상시킬 수 있었다.

향후 연구로는 워크로드를 식별한 결과를 바탕으로 각 워크로드 종류에 따른 자동화된 데이터베이스 튜닝 원칙을 제시하는 시스템을 연구할 예정이다.

참 고 문 헌

[1] 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.

[2] S. Y. Kim, H. C. Roh and S. H. Park, "Automatic Identification of database workloads by using SVM workload classifier," Journal of the Korea Contents Association, in press 2010(in Korean).

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

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

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

[6] P. Martin, W. Powley, H. Y. Li and K. Romanufa, "Managing Database Server Performance to Meet QoS Requirements in Electronic Commerce Systems," International Jornal on Digital Libraries, vol.3, no.4, pp.316-224, 2002.

[7] R. Hankins, T, Diep, M. Annavaram, B. Hirano and H. Eri, "Scaling and Characterizing Database Workloads : Bridging the Gap between Research and Practice," Proceedings of the 36th Annual ACM/IEEE International Symposium on Microarchitecture, pp.151-163, 2003.

[8] W. W. Hsu, A. J. Smith, and H. C. Young, "I/O Reference Behavior of Production Database Workloads and the TPC Benchmarks-An Analysis at the Logical Level," ACM Transactions on Database Systems, vol.26, no.1, pp.96-143, 2001.

[9] V. Singhal and A. J. Smith, "Analysis of Locking Behavior in Three Real Database Systems," The International Journal on Very Large Data Bases, vol.6, no.1, pp.40-52, 1997.

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

[11] S. Elnaffar and P. Martin, "The Psychic - Skeptic Prediction Framework for Effective Monitoring of DBMS Workloads," Data & Knowledge Engineering, 68(4) : 393-414, 2009.

[12] M. Holze, C. Gaidies and N. Ritter, "Consistent On-line Classification of dbs Workload Events," Proceeding of the 18th ACM Conference on Information and Knowledge Management Table of Contents, pp.1641-1644, 2009.

[13] M. A. Turk and A. P. Pentland, "Face Recognition using Eigenfaces," Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.586-591, 1991.

[14] 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.

[15] 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.

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

[17] 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.

김 소 연

2008년 숭실대학교 컴퓨터학부(공학사) 2010년 연세대학교 컴퓨터과학과(공학석사). 관심분야는 고성능 데이터베이스, 데이터 마이닝, 데이터베이스 튜닝

박 상 현

1989년 서울대학교 컴퓨터공학과(공학사). 1991년 서울대학교 컴퓨터공학과(공학석사). 2001년 UCLA대학교 전산학과(공학박사). 1991년~1996년 대우통신 연구원. 2001년~2002년 IBM T. J. Watson Research Center Post-Doctoral Fellow. 2002년~2003년 포항공과대학교 컴퓨터공학과 조교수. 2003년~2006년 연세대학교 컴퓨터과학과 조교수. 2006년~현재 연세대학교 컴퓨터과학과 부교수. 관심분야는 데이터베이스, 데이터 마이닝, 바이오인포매틱스, 적응적 저장장치 시스템