데이터베이스 워크로드 식별을 위한 수정된 퍼지 k-NN 알고리즘

오정석° 이상호
숭실대학교 컴퓨터학과
dbstar@nate.com°, shlee@comp.ssu.ac.kr

A Modified Fuzzy k-NN Algorithm for Identifying Database Workloads

Jeong Seok Oh° Sang Ho Lee
Department of Computing, Soongsil University

요 약

데이터베이스 관리자는 효과적인 데이터베이스 관리를 위해 워크로드 특성을 잘 알아야 한다. 워크로드 특성은 데이터베이스 응용분야에 따라 다르며, 데이터베이스 환경에서 하나 이상의 응용 분야가 수행될 수 있다. 복합적인 데이터베이스 응용 분야 때문에, 관리자가 데이터베이스 시스템에서 발생하는 워크로드를 식별하기가 더욱 어려워졌다. 복합적인 데이터베이스 응용 분야의 효과적인 데이터베이스 관리를 수행하기 위해 워크로드를 식별할 수 있는 방법이 요구된다. 이를 위해, 본 연구는 TPC-C와 TPC-W 성능평가의 워크로드와 두 성능평가의 혼합된 워크로드들을 생성하여 워크로드 식별을 수행하였다. 워크로드 식별은 퍼지 k-NN 알고리즘을 수정하여 진행하였다. 수정된 k-NN 알고리즘은 혼합 비율에 따라 시험 워크로드 데이터와 훈련 워크로드 데이터간의 워크로드 식별 실험에 사용되었고, 분류를 위한 k-NN, 퍼지 k-NN, 분산 가중치 퍼지 k-NN 알고리즘의 결과와 비교되었다. 수정된 k-NN 알고리즘은 다른 알고리즘보다 k 인자에 따른 변동과 오차율이 감소하여 워크로드 식별에 더 적합함을 보였다. 본 논문의 결과는 복합된 데이터베이스 응용 분야의 특성을 보이는 데이터베이스 환경에서 워크로드 식별 정보를 참조하여 융통성 있는 튜닝 기법을 고려하는데 기여한다.

1. 서 론

데이터베이스 관리자는 효과적인 데이터베이스 관리를 위해 워크로드 특성을 잘 알아야 한다. 워크로드 특성은 데이터베이스 응용 분야에 따라 다르다. 데이터베이스 시스템의 응용 분야가 다양해짐에 따라, 데이터베이스 시스템 환경에서 발생되는 워크로드 집합은 복잡해지고 특성이 복합되어 관리자가 분석하고 식별하기 더욱 어려워졌다. 또한 데이터베이스 시스템은 데이터베이스 응용분야가 혼합되어 수행될 수 있기 때문에 혼합된 응용분야의 워크로드를 식별할 수 있어야 한다.

데이터베이스 워크로드에 대한 연구는 워크로드 특성 분석부터 워크로드 분류(classification)에 이르기까지 많은 연구가 수행되었다. 그러나 대부분의 연구가 단일 데이터베이스 응용분야(OLTP, DSS, 웹 전자상거래)에서 진행되었기 때문에 실제 데이터베이스 시스템에서 발생되는 워크로드와는 차이가 존재할 수 있다[8]. 복합된 데이터베이스 응용 분야가 수행되는 데이터베이스 환경에 단일 데이터베이스 응용분야에서 고려된 데이터베이스 튜닝 방식을 수행한다면 예상치 못한 성능의 하락을 발생될 수도 있다. [3,4]는 이러한 문제를 데이터마이닝의 의사 결정 트리(decision tree)에 의해 분류로서 해결하려고 시도하였다. 이 연구는 두 개의 응용분야에서 수집된 워크로드를 이용해 분류를 시도하지만 복합된 워크로드에 대한 식별이 아니어서 응용분야가 복합된 데이터베이스 환경에는 부적합 할 수 있다.

본 논문은 복합된 데이터베이스 응용 분야에서 발생되는 워크로드를 식별하기 위해 TPC-C와 TPC-W 성능 평가의 워크로드와 두 성능평가의 혼합된 워크로드를 생성하여 워크로드 식별을 수행하였다. 워크로드 식별에 필요한 훈련 데이터는 성능평가별로 14개의 성능지표(performance indicator)를 통해 수집하였고, 시험 데이터는 TPC-C와 TPC-W 성능평가를 동시에 수행시켜 소비된 CPU 시간을 기준으로 혼합하여 생성하였다[1]. 워크로드 식별은 워크로드 데이터 속성의 특징을 모두 반영할 수 있는 k-NN(k-nearest neighbor) 알고리즘을 이용하였다. 본 논문은 분류를 위한 기존 k-NN 알고리즘들이 워크로드 식별에 대해 한계점을 보이므로 퍼지 k-NN 알고리즘을 수정하였다. 수정된 퍼지 k-NN 알고리즘은 혼합 비율에 따라 시험 데이터와 훈련 데이터간의 워크로드 식별 시험에 사용되었고, 기존 k-NN 알고리즘들의 결과와 비교되었다. 본 논문의 결과는 복합적인 데이터베이스 응용분야의 환경에서 워크로드 식별 정보를 참조하여 융통성 있는 튜닝 방식을 고려하는데 기여한다.

본 논문의 구성은 다음과 같다. 2장은 분류를 위한 기존 k-NN 알고리즘들을 설명하고 워크로드 식별의 한계를 기술한다. 3장은 워크로드 식별을 위해 수정된 퍼지 k-NN 알고리즘에 대해서 설명한다. 4장은 워크로드 식별 시험을 수행하고 기존 k-NN 알고리즘과 수정된 k-NN 알고리즘의 결과를 비교한다. 5장은 결론을 맺고 향후 연구 계획을 제시한다.

2. 분류를 위한 k-NN 알고리즘과 한계

k-NN 분류기(classifier)는 유사성(analogy)에 기반을 둔 학습을 수행하며 시험 데이터가 분류될 때까지 분류 모델을 생성하지 않아 인스턴스 기반의 학습기라고도 한다. k-NN 분류기는 k-NN 알고리즘을 사용함으로써 필요한 모든 데이터를 n-차원 공간의 점으로 간주하여 분류 및 예측을 수행한다[6].

k-NN 알고리즘은 시험 데이터와 훈련 데이터 집합 사이에서 거리를 계산하여 가장 가까운 k 개의 이웃 집합을 선별한다. 시험 데이터의 클래스는 선별된 k 개의 이웃 집합에서 다수 분포(majority distribution)를 가지는 클래스로 결정된다[2].

k-NN 알고리즘은 가중치를 부여하거나 퍼지 집합의 개념을

적용하여 정확성을 개선시키는 노력을 수행해왔다. [7]은 퍼지 개념을 적용한 퍼지 k-NN 알고리즘을 제시하였다. 이 알고리즘은 시험 데이터의 클래스 부여를 위해 해당 클래스의 소속 정도(class membership grade)를 사용하였다. 클래스 소속 정도는 초기 클래스 소속 정도와 최종 클래스 소속 정도에 의해 계산된다. 초기 클래스 소속 정도는 k 개의 이웃 집합에서 클래스 분포와 훈련 데이터의 클래스 정보를 이용하여 결정된다. 최종 클래스 소속 정도는 시험 데이터와 k 개의 이웃 집합에 속하는 훈련 데이터간의 거리와 초기 클래스 소속 정도에 의해 결정되며, 시험 데이터가 존재하는 클래스에 얼마나 소속될 수 있는가를 나타낸다.

[5]는 퍼지 k-NN 알고리즘으로서 정확성을 향상시키는 목적으로 제안되었다. 이 알고리즘은 해당 클래스 소속 정도의 구동 원리를 변경시킴으로써 정확성 향상을 시도하였다. k 개의 이웃 집합의 결정은 훈련 데이터와 시험 데이터간의 유사도(similarity)를 이용하였다. 유사도 수식을 이용하기 위해 모든 차원들은 정규화 되고 차원의 합을 차원 개수로 나누었다. 초기 클래스의 소속 정도는 시험 데이터의 조건적 밀도(conditional density)를 반영하여 결정된다. 최종 클래스 소속 정도는 이웃 집합에서 해당 클래스 소속에 대한 분산 가중치와 시험 데이터와 k 개 이웃 집합의 훈련 데이터간의 유사도와 초기 클래스 소속 정도에 의해 결정된다.

분류를 위한 기존의 (퍼지) k-NN 알고리즘은 워크로드 식별에 대해 두 가지 한계를 보인다. 첫 번째 한계는 분류 과정의 강제 클래스 부여로 인해 발생된다. 분류를 위한 (퍼지) k-NN 알고리즘에서 시험데이터와 클래스간의 소속 정도가 확률로서 제공되더라도, 시험데이터가 k-NN 분류기에서 높은 확률을 가진 클래스로 무조건 귀속되는 한계가 존재한다. 다시 말해, 시험 데이터의 클래스는 클래스 소속 정도가 객관적으로 높지 않더라도 다른 클래스 집합의 소속 정도보다 높다면 강제로 부여된다. 두 번째 한계는 k 인자 변경에 따른 결과의 진동(oscillation)이다. 기존 (퍼지) k-NN 알고리즘은 이웃 집합에서 관계를 형성하기 때문에 k 인자의 수와 이웃 집합의 구성 요소에 따라 결과가 달라질 수 있다. 결과의 부동성(instability)은 알고리즘의 정확성을 하락시키고 사용자에게 혼란을 가중시킨다.

3. 워크로드 식별을 위한 수정된 퍼지 k-NN 알고리즘

본 연구는 강제 클래스 부여에 대한 한계 때문에 클래스 소속 정보를 유지하며 특정 클래스에 귀속시키지 않는다. 전체 시험 데이터에 대한 클래스 소속 정도는 각 시험 데이터의 클래스 소속 정도의 평균으로 계산된다.

$$U_{n\times c}=\begin{bmatrix}0.8 & 0.2\\ 0.3 & 0.7\\ 0.6 & 0.4\\ 0.9 & 0.1\\ 0.3 & 0.7\end{bmatrix} \Rightarrow U_{n\times c}=\begin{bmatrix}1 & 0\\ 0 & 1\\ 1 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix} \Rightarrow \begin{array}{l}\text{클래스 1: 60\%}\\ \text{클래스 2: 40\%}\end{array}$$

소속 정도 강제 클래스 부여

그림 1 기존 k-NN 방식

예를 들어, n이 데이터 개수이고 행을 이루며 c는 클래스의 개수이고 열을 이룬다고 가정할 때, $n\times c$인 매트릭스가 존재할 수 있다. 분류를 위한 k-NN 방식을 $n\times c$ 매트릭스에 표시해 보면 그림 1처럼 도식화 될 수 있다. 첫 번째 데이터가 클래스 1에 소속될 비율은 80%이며 클래스 2에 소속될 비율은 20%이므로, 클래스 1이 첫 번째 데이터의 클래스로 강제 부여된다. 나머지 데이터도 동일한 방식으로 강제 클래스 부여가 수행되어 전체 데이터의 60%가 클래스 1이 되고, 40%가 클래스 2가 된다.

$$U_{n\times c}=\begin{bmatrix}0.8 & 0.2\\ 0.3 & 0.7\\ 0.6 & 0.4\\ 0.9 & 0.1\\ 0.3 & 0.7\end{bmatrix} \Rightarrow \begin{array}{l}\text{클래스 1: 58\%}\\ \text{클래스 2: 42\%}\end{array}$$

소속 정도

그림 2 소속 정보를 유지하는 방식

같은 가정으로 소속정보를 유지하는 방식을 $n\times c$ 매트릭스에 표시해보면 그림 2처럼 도식화 될 수 있다. 시험 데이터의 소속 정도는 강제 클래스 부여 과정을 수행하지 않고 시험 데이터들의 평균 소속 정도를 구한다. 전체 시험 데이터는 클래스 1에 58% 정도 유사하고 클래스 2에 42% 정도 유사하다고 말할 수 있다.

본 연구는 k 인자의 변경에 따른 진동을 감소시키기 위해서 초기 클래스 소속 정도를 계산할 때 k 개의 이웃 집합 사이의 관계를 고려하지 않고 시험데이터와 각 클래스 집합의 중심점 사이가 얼마나 밀접한지를 고려한다. 그림 3은 기존 방식에서 변화된 점을 도식화하여 보여주고 있다. 이웃 집합의 개수가 3이라고 가정하자. 퍼지 3-NN 방식은 y1을 고려할 때 이웃 집합 y1, y2, y3만을 고려하여 분포를 내거나 가중치를 부여해 초기 소속 정도를 구한다. 제안된 방식은 y1을 고려할 때 x와 클래스 1의 중심점 및 클래스 2의 중심점이 얼마나 밀접한가를 계산하여 초기 소속 정도를 구한다. y2와 y3도 동일한 방식으로 계산된다.

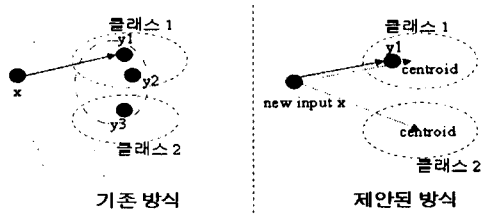

그림 3 기존 방식과 제안된 방식의 비교

변경된 방식의 초기 소속 정도 계산식은 수식 1에서 보인다. $\tilde{\mu}_i(y)$는 훈련 데이터가 클래스 i에 속할 때와 속하지 않을 때의 퍼지 값을 각 클래스의 중심점을 고려하여 계산한다. k는 클래스 집합의 개수이다. C_i는 현재 훈련 데이터의 클래스를 의미하고 C_j는 최종 클래스 소속 정도의 계산에서 사용되는 시험 데이터의 가정된 클래스를 의미한다.

$$\tilde{\mu}_i(y)=\begin{cases}0.51+\left(\dfrac{1}{\sum_{k=1}^{c}\left[\frac{\|x-c_i\|}{\|x-c_k\|}\right]^{2/(m-1)}}\right)*0.49, & \text{if } C_i=C_j\\ \left(\dfrac{1}{\sum_{k=1}^{c}\left[\frac{\|x-c_i\|}{\|x-c_k\|}\right]^{2/(m-1)}}\right)*0.49, & \text{if } C_i\neq C_j\end{cases} \quad \text{수식 1}$$

최종 클래스 소속 정도의 계산은 수식 2에서 보인다. i의 범위는 1부터 클래스 개수까지이며 k는 클래스 집합의 개수이고 $\mu_i(x)$는 클래스 i에 대한 x의 소속 정도를 의미한다.

$$\mu_i(x) = \frac{\sum_{j=1}^{k} \tilde{\mu}_j(y)(1/\| x - y_j \|^{2/(m-1)})}{\sum_{j=1}^{k} (1/\| x - y_j \|^{2/(m-1)})} \quad \text{수식 2}$$

4. 실험 및 결과

실험에 필요한 워크로드 데이터는 오라클 9i에서 TPC-C와 TPC-W 성능평가를 수행시켜 14개의 성능지표 값을 수집하였다. 훈련 데이터는 성능평가별로 수집되었으며, TPC-C 성능평가에서 수집된 훈련 데이터의 클래스는 TPC-C로 설정되고, TPC-W 성능평가에서 수집된 훈련 데이터의 클래스는 TPC-W로 설정되었다. 시험 데이터는 TPC-C와 TPC-W를 동시에 수행시켜 워크로드를 혼합 시켰다. 혼합 데이터는 오라클 9i에서 제공하는 데이터베이스 자원 관리기(resource manager)와 TPC-C의 웨어하우스와 TPC-W의 EB수를 조절하여 소비된 CPU 시간을 기준으로 혼합하였다. 혼합 비율은 TPC-C와 TPC-W를 80%와 20%(이하 비율 1), 50%와 50%(이하 비율 2), 20%와 80%(이하 비율 3)으로 구성되었고 각 비율별 시험 데이터는 다섯 번 수집되었다.

실험은 k 인자를 1부터 10까지 변경하고 다섯 개의 시험 데이터 집합을 이용하여 k-NN, 퍼지 k-NN, 분산 가중치 k-NN, 제안된 k-NN별로 결과를 산출하여 총 600회의 실험이 실시되었다. 실험의 결과는 k 인자에 따른 변동과 혼합 데이터 비율과 분류된 비율의 차이인 오차율을 이용하여 네 개의 알고리즘 결과를 비교하였다.

k 인자에 따른 변동은 혼합 비율별 식별 비율에 대한 평균 표준 편차 구하여 분석하였다. 변동은 평균 표준 편차가 낮을수록 적게 발생된 것이며, 변동의 분석 결과는 표 1에서 보인다. 제안된 퍼지 k-NN이 모든 비율에서 낮은 표준 편차 값을 보여 변동이 적었으며, 다른 k-NN 알고리즘에 비해 최소 2배에서 최대 104배까지 평균 표준 편차가 감소되었다.

표 1 평균 표준 편차

알고리즘	비율 1	비율 2	비율 3
k-NN	0.13996	0.61181	0.23052
퍼지 k-NN	0.036386	0.015891	0.010451
분산 가중치 퍼지 k-NN	0.036177	0.033973	0.031796
제안된 퍼지 k-NN	0.005484	0.00589	0.007204

알고리즘의 정확성을 알아보기 위한 평균 오차율은 표 2에서 보인다. 제안된 퍼지 k-NN이 모든 혼합 비율에서 낮은 오차율을 보였으며, 다른 k-NN 알고리즘에 비해 최소 1배에서 최대 4.3배까지 오차율이 감소되었다.

표 2 평균 오차율

알고리즘	비율 1	비율 2	비율 3
k-NN	12.51%	21.62%	7.08%
퍼지 k-NN	5.13%	21.53%	6.1%
분산 가중치 퍼지 k-NN	11.08%	28.8%	5.12%
제안된 퍼지 k-NN	2.94%	21.45%	1.69%

5. 결론

데이터베이스 시스템 환경에서 발생되는 워크로드 집합은 복합적인 워크로드 특성을 보일 수 있어 관리자가 데이터베이스 시스템의 워크로드를 식별하기 더욱 어려워졌다. 효과적인 데이터베이스 시스템 관리를 위해 데이터베이스 용용분야의 특성이 복합적으로 나타나는 데이터베이스 환경의 워크로드를 식별할 수 있는 방법이 요구된다.

본 논문은 복합된 데이터베이스 응용 분야에서 발생되는 워크로드를 식별하기 위해 TPC-C와 TPC-W 성능 평가의 워크로드와 두 성능평가의 혼합된 워크로드를 생성하여 워크로드 식별을 수행하였다. 워크로드 식별에 필요한 워크로드 데이터는 14개의 성능지표를 통해 TPC-C와 TPC-W에서 워크로드를 수집하여 훈련 데이터로 사용하고, 두 성능평가에서 혼합된 워크로드를 생성하여 시험 데이터로 이용하였다. TPC-C와 TPC-W의 워크로드 혼합 비율은 20%와80%, 50%와50%, 80%와20%로 구축하였다. 워크로드 식별은 수정된 퍼지 k-NN 알고리즘에 의해 수행된다. 수정된 퍼지 k-NN 알고리즘은 혼합 비율에 따라 시험 데이터와 훈련 데이터간의 워크로드 식별 시험에 사용되었고, 분류를 위한 k-NN, 퍼지 k-NN, 분상 가중치 k-NN 알고리즘들의 결과와 비교하였다. 워크로드 식별 결과는 k 인자에 따른 변동과 오차율을 통해 분석되었으며 제안된 퍼지 k-NN 알고리즘의 방식이 다른 알고리즘보다 변동과 오차율이 적음을 확인하였다.

본 연구의 결과는 복합된 데이터베이스 응용 분야의 워크로드를 단일 응용 분야의 워크로드와 비교한 워크로드 식별 정보를 제공하여 단일 응용 분야에서 연구되었던 데이터베이스 튜닝방식에 워크로드 식별 정보를 적용하여 적합한 튜닝 방식을 고려할 수 있다. 본 연구의 향후 계획은 워크로드 식별 정보를 이용해 데이터베이스 튜닝에 적용할 수 있는 방법을 연구할 예정이다.

6. 참고문헌

[1] R. Baylis, Database Administrator's Guide: Release 2(9.2), Oracle Corporation, 2002.

[2] T. M. Cover and P. E. Hart, Nearest Neighbor Pattern Classification, IEEE Transactions on Information Theroy, Vol. 13, No. 1, pages 21-27, 1967.

[3] S. Elnaffar, A Methodology for Auto-recognizing DBMS Workloads", The proceedings of CASCON Conference, Toronto, USA, 2002.

[4] S. Elnaffar, P. Martin, and R. Horman, Automatically Classifying Database Workloads, The proceedings of 11th CKIM Conference, pages 622-624, McLean, USA, 2002.

[5] J. H. Han and Y. K. Kim, A Fuzzy K-NN Algorithm using Weights from the Variance of Membership Values, The proceedings of IEEE CVPR Conference, pages 394-399, Fort Collins, USA, 1999.

[6] J. Han and M. Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, 2001.

[7] J. M. Keller, M. R. Gray, and J. A. Givens, A Fuzzy k-Nearest Neighbor Algorithms, IEEE Transaction on System, Man and Cybernetics, Vol. 15, No. 4, pages 580-585, 1985.

[8] P. Martin, W. Powley, H. Y. Li, and K. Romanufa, Managing Database Server Performance to Meet QoS Requirements in Electronic Commerce Systems, International Journal on Digital Libraries, Vol. 3, No. 4, pages 316-324, 2002.