다차원 데이터의 효과적인 유사도 검색을 위한 색인구조

복경수° 허정필 유재수
충북대학교 정보통신공학과
{ksbok°, hjungpili}@netdb.chungbuk.ac.kr
yjs@cbucc.chungbuk.ac.kr

Index Structure for Efficient Similarity Search of Multi-Dimensional Data

Kyoung Soo Bok° Jung Pil Heo Jea Soo Yoo
Dept. of Computer and Communication Engineering, Chungbuk National University

요 약

본 논문에서는 다차원 데이터의 유사도 검색을 효과적으로 수행하기 위한 색인 구조를 제안한다. 제안하는 색인 구조는 차원의 저주 현상을 극복하기 위한 벡터 근사 기반의 색인 구조이다. 제안하는 색인 구조는 부모 노드를 기준으로 KDB-트리와 유사한 영역 분할 방식으로 분할하고 분할된 각 영역은 데이터의 분포 특성에 따라 동적 비트를 할당하여 벡터 근사화된 영역을 표현한다. 따라서, 하나의 노드 안에 많은 영역 정보를 저장하여 트리의 깊이를 줄일 수 있다. 또한 다차원의 특징 벡터 공간에 상대적인 비트를 할당하기 때문에 군집화되어 있는 데이터에 대해서 효과적이다. 제안하는 색인 구조의 우수성을 보이기 위해 다양한 실험을 통하여 성능의 우수성을 입증한다

1. 서론

최근 몇 년 동안 다차원 데이터들을 효과적으로 처리하기 위한 많은 색인 기법들이 제안되었다[1]. 기존에 제안된 다차원의 색인 구조는 차원이 증가할수록 노드의 팬아웃이 감소되고 색인 구조의 높이를 증가시킬 뿐만 아니라 분할된 영역들 사이에 겹침이 증가되어 검색 성능이 저하되는 문제점이 있다. 특히, k-최근접 검색에 대해서는 차원이 증가함에 따라 검색 성능이 급속히 저하되는 문제점이 있다. 이러한 문제를 해결하기 위해 데이터 공간을 작은 단위로 분할하고 분할된 영역을 비트 형태로 표현하는 벡터 근사 기법에 대한 연구들이 진행되고 있다[2].

벡터 근사 기법의 대표적인 VA-파일은 각 차원에 특정 비트를 할당하여 전체 데이터 영역을 2^b의 셀로 분할한다[2]. 분할된 각 셀은 근사화되어 배열에 저장되고 검색 과정에서 전체 배열을 검사한다. CS-트리는 특징 벡터 공간을 VA-파일과 유사한 방법으로 분할하고 근사화된 값을 R-트리 기반의 색인 구조로 표현한다[3]. A-트리는 R-트리 기반의 계층 구조를 기준으로 상대적인 영역 정보 또는 데이터 값을 비트 형태로 표현한다[4]. 이러한 색인 구조들은 팬아웃을 증가시켜 색인 구조의 높이를 감소시키는 장점이 있다. 그러나 전체 영역 또는 상대적인 영역을 표현하기 위해 고정된 비트 값을 할당하기 때문에 근사화된 영역들에 사이에 정확도가 감소되는 문제점이 있다. 또한 데이터들이 특정 영역에 집중되어 있는 경우에는 계속적인 분할로 영역들 사이에 겹침 영역이 증가된다.

따라서 본 논문에서는 기존에 제안된 벡터 근사 기법의 문제점을 해결하기 위한 새로운 다차원 색인 구조를 제안한다. 제안하는 색인 구조에서는 KDB-트리와 유사한 영역 분할을 수행하고 각각의 분할된 영역에 대해 실제 데이터가 존재하는 영역에 대해서 근사화를 수행한다. 또한 하위 노드는 부모 노드를 기준으로 상대적으로 표현하여 데이터의 분포 특성에 따라 동적으로 비트를 할당한다.

"이 논문은 2003년도 한국학술진흥재단의 선도연구자 지원사업(KRF-2003-041-D00489)에 의하여 연구되었음"

본 논문의 구성은 다음과 같다. 먼저 2장에서는 기존에 제안된 벡터 근사 기반의 다차원 색인 구조를 기술하고 3장에서는 제안하는 색인 구조에 대해 기술한다. 4장에서는 제안하는 색인 구조에 대한 성능 평가를 수행하고 5장에서는 결론에 대해 기술한다.

2. 관련 연구

VA-파일은 전체 벡터 공간상에 존재하는 객체들의 특징 벡터 값을 간단한 비트 형태로 표현하기 위한 객체 근사화를 지원하는 색인 구조이다. 이러한 VA-파일은 사용자가 정의한 b개의 비트를 이용하여 데이터 공간을 2^b 개의 셀로 분할한다. 분할된 각 셀에 b개의 비트를 이용하여 유일한 값을 할당하여 셀에 포함된 데이터들을 근사화한다. VA-파일은 근사화된 데이터 값들을 배열 구조에 저장하고 k-최근접 검색 과정에서 근사화된 값들을 순차적으로 순회하기 때문에 데이터 수의 증가에 따라 검색 성능이 저하된다. 또한 데이터의 분포가 특정 영역에 군집화되어 있을 경우 근사화된 값들 사이에 변화가 없기 때문에 검색 성능이 급격히 저하되는 문제점이 있다

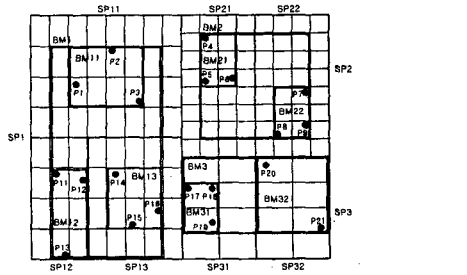
CS-트리는 기존 R-트리 기반의 다차원 색인 구조에서 특징 벡터의 차원이 증가할수록 팬아웃이 감소하여 트리의 높이가 증가한다는 문제점을 해결하기 위해 데이터 공간을 일정한 크기의 셀들로 분할하여 셀 기반의 MBR을 표현하는 색인 구조이다. 그러나 전체 영역을 고정된 셀로 분할하여 비트를 할당하기 때문에 객체의 동적인 삽입이나 삭제에 의해 정확도가 감소될 수 있으며 VA-파일과 같이 특정 영역에 군집화되어 있을 경우 검색 성능이 저하되는 문제가 발생할 수 있다.

3. 제안하는 색인구조

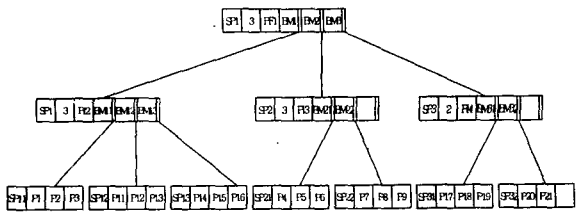
제안하는 ABA-트리는 다차원 색인 구조의 문제점을 해결하기 위해 다차원 데이터 공간을 KDB-트리와 유사한 영역 분할 기법을 이용하여 분할된 영역에 동적 비트를 할당하는 벡터 근사화 트리이다. 기존에 제안된 색인 구조는 벡터 근사화 기법을 수행하기 위해 전체 영역 또는 분할된 일부 영역에 고정된 비트를 할당하여 노드의

팬아웃을 증가시켰다. 그러나 이러한 기법들은 특정 영역에 데이터들이 군집화되어 있을 경우 분할된 영역들 사이에 선별력이 감소될 수 있다. 따라서 제안하는 색인 구조에서는 분할된 영역에 고정된 비트를 할당하는 것이 아니라 데이터의 분포 상태에 대한 동적으로 비트를 할당하는 *BMBR*(Bit MBR)를 나타낸다. 기존의 데이터 분할 기반의 색인 구조는 차원이 증가할수록 분할된 영역들 사이에 겹침 영역이 증가되어 검색 성능이 저하될 수 있다. 따라서 특정 노드에 오버플로우가 발생할 경우 KDB-트리와 유사한 형태의 영역 분할을 사용한다.

그림 1은 제안하는 색인 구조를 나타낸 것이다. 그림 1의 (a)와 같이 다차원의 데이터 공간이 분할되어 있다고 할 때 이를 통해 색인을 구성하면 그림 4의 (b)와 같다. 그림 1에서 P_i는 다차원의 포인트 데이터를 나타내고 BM_i는 분할된 영역에 대한 영역을 나타내는 *BMBR*을 나타낸다. 또한 SP_i는 영역 분할을 수행한 영역으로 SP_i을 통해 상대적인 비트를 할당하여 BM_i을 나타낸다. 제안하는 색인 구조는 영역 분할에 의해 분할된 영역에 동적인 비트를 할당하여 하위 노드에 대한 상대적인 비트를 표현한다. 따라서 그림 1의 (b)에서 보는 것과 같이 중간 노드의 처음에는 분할된 영역에 대한 *BMBR*을 나타내기 위한 헤더를 포함하고 있다. 또한 단말 노드에는 오버플로우가 발생할 경우 영역 분할을 수행하기 위해 기존에 분할된 영역을 헤더로 포함하고 있다.



(a) 분할 영역



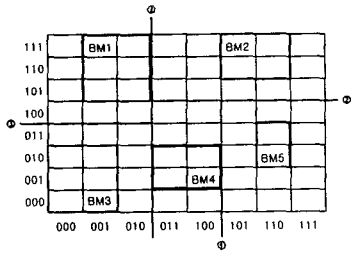
(b) 색인 구조

그림 1 제안하는 색인 구조

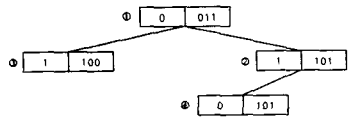
3.1 노드 구조

제안하는 색인 구조에서 중간 노드는 단말 노드에 존재하는 다차원 데이터를 포함하는 영역을 상대적 비트를 할당하여 표현한다. 또한 데이터 특징벡터의 분포에 따라서 할당되어진 동적 비트 정보를 저장한다. 이러한 중간 노드는 하위 노드의 영역을 표현하는 엔트리와 헤더로 구성된다. 그림 2는 중간 노드의 구조를 나타낸 것이다. 중간 노드에 존재하는 헤더는 동적인 비트 할당을 통해 자식 노드의 영역을 표현하기 위한 기준 정보로 $<SP,BitNum,PI>$와 같다. *SP*는 자식 노드를 포함하는 영역 정보로 자식 노드의 전체 영역 정보 *BMBR*을 표현하기 위한 기준 영역이다. *SP*는 영역 분할에 의해 생성된 영역으로 동일한 레벨에 존재하는 *SP*들 사이에는 겹침 영역이 발생하지 않는다. *BitNum*는 현재 노드에 존재하는 엔트리들의 *BMBR*을 표현하기 위한 비트 수로 분할 과정에서 데이터의 분포 특성에 따라 동적으로 결정된다. *PI*는 하위 노드에 대한 분할 정보로 중간 노드에 존재하는 엔트리들의 영역 기반 분할 정보를 나타낸다. 중간 노드에 존재하는 엔트리는 하위 노드를 포함하는 영역을 표현하기 위한 정보를 표현하는 것으로 $<BM_i,PTR_i>$와 같다. BM_i는 헤더에 포함된 *SP*을 기준으로 *BitNum*의 비트로 표현된 하위 노드에 대한 영역 정보 *BMBR*이고 PTR_i는 하위 노드에 대한 포인터를 나타낸다.

하위 노드에 대한 영역 정보를 나타내는 *BMBR*은 비트 형태로 분할된 영역을 나타내기 때문에 영역 기반을 삽입 및 분할을 수행하기 위해서는 하위 노드에 존재하는 영역들에 대한 분할 정보 즉, 영역 기반 분할을 수행한 정보를 포함해야 한다. *PI*는 하위 노드에 존재하는 영역이 분할된 정보로 중간 노드에 오버플로우가 발생할 경우 *BMBR*을 통해 영역 기반 분할을 수행할 수 없기 때문에 영역 기반의 분할 위치를 선택하거나 새로운 데이터가 삽입될 위치를 선택하기 위해 사용된다. 그림 2는 중간 노드에 존재하는 분할 정보 *PI*을 나타낸 것이다. 그림 2의 (a)와 같이 번호 순서에 의해 5개의 분할된 영역이 존재한다고 하자. 이때, BM_i는 영역 분할에 의해 생성된 하위 노드에 대한 *BMBR*을 나타낸 것이다. 분할된 5개의 영역에 대한 분할 정보는 그림 2의 (b)와 같이 분할 차원과 위치의 쌍 $<SplitAxis,SplitPosition>$에 의해 표현된다. 분할 차원과 분할 위치를 나타내는 *SplitAxis*과 *SplitPosition*는 비트 형태로 표현된다.



(a) 분할 영역



(b) 분할 정보

그림 2 중간 노드의 분할 정보

제안하는 색인 구조의 단말 노드는 실제적인 다차원의 데이터를 저장한다. 단말 노드는 그림 4에서 보는 것과 같이 중간 노드와 유사하게 헤더와 엔트리로 구성된다. 단말 노드의 엔트리 $<FV_i,OID_i>$는 실제적인 다차원의 데이터 FV_i와 객체 식별자 OID_i로 구성되어 있다. 다차원의 데이터 FV_i는 다차원의 데이터를 나타낸다. 단말

노드의 헤더는 *SP*로 구성되어 있으며 *SP*는 기존의 영역 분할에 의해 분할된 영역 정보를 나타낸다. 이러한 *SP*는 단말 노드에 영역 기반 분할을 수행하기 위해 단말 노드에 존재하는 엔트리들이 존재하는 영역을 나타낸다. 즉, 단말 노드에 다차원의 포인트 데이터를 포함하고 있다면 실제적인 데이터만 포함된 영역을 나타내기 때문에 영역 분할을 수행하기 위한 영역을 포함하고 있어야 한다.

3.2 BMBR 표현

ABA-트리에서는 영역 분할 기법을 사용하여 분할된 노드에 대해 부모 노드를 기준으로 상대적인 영역을 표현한다. 즉, 겹침 영역이 없이 분할된 영역은 정적인 비트로 할당하여 표현하지 않고 자식 영역의 분포 특성을 고려하여 효과적으로 표현할 수 있는 동적 비트를 할당하여 자식 영역 정보를 좀더 정확하고 효과적으로 표현한다. 분할된 영역 내에 *BMBR*을 표현하기 위한 비트 수를 결정하기 위해 실제 데이터들이 존재하는 영역과 비트 형태로 근사화한 *BMBR*의 차이를 이용하여 계산한다. 그림 3과 같이 중간 노드에서 분할된 두 개의 영역이 존재할 때 분할된 영역에 대한 *BMBR*을 표현하면 표 1과 같다.

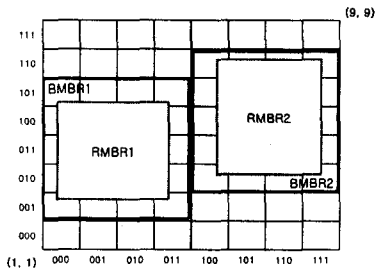


그림 3 두 차원 데이터의 분할 영역

표 1 BMBR 표현

	RMBR	BMBR
RMBR1	(1.4, 2.7, 4.5, 6.2)	(000 001 011 101)
RMBR2	(5.6, 3.5, 8.4, 7.8)	(100 010 111 110)

4. 성능 평가

실험 평가를 위해 제안하는 ABA-트리와 CS-트리를 Pentium-IV 1.8㎓ CPU와 256MB의 윈도우즈 2000 시스템 환경에서 C언어를 이용하여 구현한다. 성능 평가를 위해 각각 임의로 생성된 10만개의 랜덤 데이터 집합과 코렐 이미지에서 추출된 실제 데이터 집합을 사용한다. ABA-트리의 검색 성능을 평가하기 위해서 가장 대표적인 유사성 검색 질의인 k-최근접 질의와 범위 질의를 수행하여 수행 시간을 측정한다. 검색 시간은 질의를 50번 수행한 평균 값을 이용한다. 그림 4와 5는 범위 질의와 K-최근접 검색에 대한 시간을 나타낸 것이다.

제안하는 ABA-트리의 전체적인 검색 성능은 기존에 제안된 CS-트리보다 약 40%정도의 성능 향상을 보이고 있다. 제안하는 색인 구조는 단말 노드로의 검색에 있어서 KDB-트리와 같은 영역 분할 방식으로 표현된 노드로 인해서 접근해야 할 노드를 감소시켜 페이지의 접근 횟수가 감소시키고 엔트리 내에 저장되는 MBR 정보를 비트 MBR로 표현함으로서 팬아웃이 증가하게 된다. 이와 같이 접근해야 할 페이지의 감소와 팬아웃의 증가는 유사성 기반의 검색 시간을 단축시킨다.

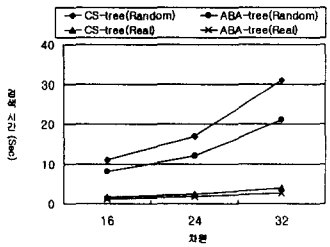


그림 4 범위 질의시 검색 시간

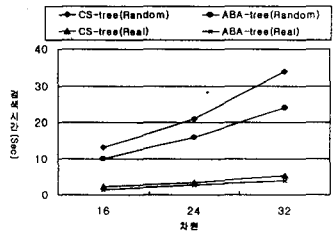


그림 5 k-최근접 질의시 검색 시간

5. 결론

본 논문에서는 차원의 저주 현상을 해결하기 위한 벡터 근사 기반의 다차원 색인 구조를 제안하였다. 제안하는 색인 구조는 하위 노드 영역을 부모 노드를 기준으로 상대적으로 표현하고 각각의 영역은 데이터 분포 특성에 따라 동적인 비트를 할당하여 근사화한다. 또한 차원의 증가에 따른 영역들 사이의 겹침 영역을 감소시키기 위해 영역 분할 방식을 사용한다. 성능 평가 결과 제안하는 색인 구조는 기존에 제안된 CS-트리보다 약 40%정도의 검색 성능이 향상됨을 보였다.

참고문헌

[1] V. Gaede and O. Gunther, "Multidimensional Access Methods," ACM Computer Survey, Vol.30, No.2, pp.170-231, 1998

[2] R. Weber, H. J. Schek and S. Blott, "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces," Proc. 24rd International Conference on Very Large Data Bases, pp.194-205, 1998.

[3] K. T. Song, H. J. Nam and J. W. Chang, "A cell-based index structure for similarity search in high-dimensional feature spaces," Proc. the 2001 ACM Symposium on Applied Computing, pp.264-268, 2001.

[4] Y. Sakurai, M. Yoshikawa, S. Uemura and H. Kojima, "Spatial indexing of high-dimensional data based on relative approximation," VLDB Journal, Vol.11, No.2, pp.93-108, 2002.