Bitmap Index을 이용한 Incognito 성능개선

강현호, 이상원
성균관대학교 컴퓨터공학과
e-mail : hanpaldduk@skku.edu

Improvement of Incognito by using Bitmap Index

Hyun-Ho Kang , Sang-Won Lee
Dept. of Computer Engineering, SungKyunKwan University

요 약

현대사회에서는 자신도 알지못하는 많은 정보들이 유포된다. 이때 정보들은 개인의 익명성을 보장하기위해 성명, 성별, 주민등록번호와 같은 개인식별 애트리뷰트를 생략한채로 유포된다. 그러나 널리퍼져있는 이러한 정보들은 다른 외부 정보와 조인되므로써 유일하게 개인을 식별하게끔 하는 조인공격을 받을 수 있다. 하지만 이러한 조인공격시 여러데이터가나오게하므로써 개인식별을 어렵게 또는 불가능하게하는 방법을 k-anonymization이라고하고 이러한 k-anonymization을 지원하는 방법으로 이전부터 여러가지가 있다. 이전의 방법들로는 각 subset마다 k-anonymization을 검사해야했으나 Lefevre와 DeWitt가 제안한 Incognito 방법을 사용하면 한번의 검사로 모든k-anonymization을 보장할 수 있다. 이 논문에서는 이러한 Incognito를 bitmap index를 사용하므로써 성능을 개선시키는 기법을 제시한다.

1. 서론

여러 기관들에서 광고의 목적이나 또는 다른 어떤 특정목적을 위하여 많은 정보들을 유포한다. 이때 이 정보들은 개인식별 애트리뷰트를 제거한 형태로 유포되어진다. 그런데 이러한 의미 없을 수 정보들이 외부의 다른 정보들과 조인되므로서 개인을 유일하게 식별할 수 있는 조인공격을 받을 수 있다. 아래 그림-1에서 투표등록자 테이블과 병원 환자 데이터의 'Birth, Sex, Zipcode' 코드를 조인키로 사용할 경우 Andre라는 개인신상정보가 나타남을 알 수 있다. 이러한 조인공격에 대해 유일한 값이 아닌 여러개의 결과를 나타냄으로서 개인식별을 막는 방법을 k-anonymity라고 하고 k-anonymity을 지원하는 기법들로는 binary search와 Incognito가 있다. Incognito 기법은 이전까지 나온 기법들과는 달리 조인공격에 대한 모든 경우에 대하여 k-anonymity의 가능성을 검사할 수 있다는 장점을 가지고 있다. 우리는 bitmap index를 이용하여 Incognito가 k-anonymity을 검사함에 있어 성능을 향상 시킬수 있는 방법을 제시하도록 한다.

Name	Birth	Sex	Zipcode
Andre	1/21/76	Male	53715
Beth	1/10/81	Female	55410
Carol	10/1/44	Female	90210
Dan	2/21/84	Male	02174
Ellen	4/19/72	Female	02237

<표 1> 투표 등록자

Birth	Sex	Zipcode	Disease
1/21/76	Male	53715	Flu
4/13/86	Male	53703	Broken Arm
2/28/76	Male	53703	Bronchitis
1/21/76	Female	53715	Hepatitis
4/13/86	Female	53706	Sprained Ankle
2/28/76	Female	53706	Hang Nail

<표 2> 병원 환자

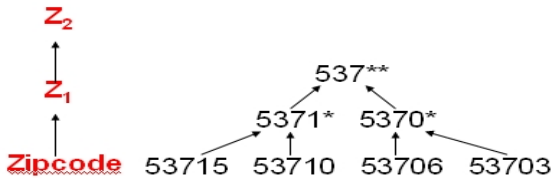
2. 관련연구

2.1 기본개념

Quasi-Identifier attribute set : 외부 정보와 조인되어 개인정보를 식별할 수 있게 하는 애트리뷰트의 최소집합
Frequency set : Quasi-Identifier의 부분집합에 대하여 group by를 하였을 때 나타나지는 총 행의 수
K-Anonymity 성질 : 조인공격시 최소한 k개 이상의

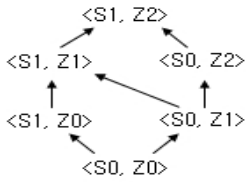
결과를 나타낼 경우 K-anonymity성질을 만족한다고 한다. 또는 Frequency set의 모든 count가 size k를 넘을 경우 k-anonymity를 만족한다고 한다.

Generalization : 값(value) 또는 도메인(domain)에 대해서 그 상위값을 의미한다. 기호로는 $<_D$ 를 사용하고 $A<_D B$로 표기한다. 이것은 B는 A의 generalization(일반화)라는 의미가 된다. 예를들어 아이스크림$<_D$빙과류는 빙과류는 아이스크림의 일반화(상위집단)라는 의미가 된다. 일반화는 값에 의한 value generalization hierarchy와 어떤 특정 범위에 의한 domain generalization hierarchy으로 나뉜다. (그림 1)은 지역에 대한 value generalization과 domain generalization을 나타내고 있다. 일반화는 연결성에 따라 직접적 일반화(direct generalization)와 내포적 일반화(implied generalization) 두가지로 나뉘게 되는데 직적접 일반화는 $53715<_D 5371*$와 같이 직접적인 관계를, 내포적 일반화는 $53715<_D 5371*$, $5371*<_D 537**$일 경우 $53715<_D 537**$와 같은 간접적인 관계를 나타낸다.



(그림 1) Value/Domain generalization hierarchy

(그림-1)은 단일 일반화 계층도(single generalization hierarchy)를 나타낸 것이다. 그러나 실제로는 이와 같은 단일형태가 아닌 지역-성별과 같이 다중 일반화 계층도(multi generalization hierarchy)이 주로 사용된다. (그림 2)를 다중 일반화 격자(multi generalization lattice) 또는 전체-도메인 일반화(full-domain generalization)라고 한다.



(그림 2) 성별/지역 Generalization lattice

2.2 binary search 알고리즘

높이 h를 가지는 격자(lattice)에서 높이 h'가 k-anonymity을 만족하지 못할 경우 h"(<h') 높이의 모든 노드들은 k-anonymity성질을 만족시키지 못한다는 특성을 이용한다. 이진탐색기법 알고리즘은 높이가 h일 때 ㄴh/2ㄱ를 검사한 뒤 해당 높이에서 k-anonymity을 만족할 경우 이 보다 높은 지점에서는 모두 k-anonymity를 만족하기 때문에 그 보다 아래지점인 ㄴh/4ㄱ를 검사한다. 그렇지 않고 ㄴh/2ㄱ에서 k-anonymity을 만족하지 않을 경우 이 위치보다 위에서 k-anonymizatin을 만족하지 않는다는 말이므로 ㄴ3h/4ㄱ 지점을 검사한다. 이 과정을 반복한다.

2.3 Incognito

이진탐색기법이 단일 일반화 격자에 대하여 k-anonymity 검사를 수행한 것에 반해 Incognito는 quasi-identifier attribute set으로 구성될 수 있는 모든 일반화 격자에 대하여 검사를 수행한다는 장점이 있다. Incognito를 이해하기 위해서는 몇 가지 속성을 이해하고 있어야한다.

Generalization property : $A <_D B$ 관계일 때 만약 A가 k-anonymity을 만족한다면 A의 일반화관계에 있는 B도 k-anonymity을 만족한다.

Rollup property : $A <_D B$ 관계일 때 A에 관련된 frequency set f1이 존재할 경우 B의 frequency set f2는 f1을 합으로써(summing) 구해질 수 있다.

Subset property : quasi-identifier attribute set의 부분집합인 Q가 k-anonymity을 만족시킬 경우 Q의 부분집합들도 k-anonymity을 만족시킨다.

Incognito 알고리즘은 기본적으로 BFS(Bread-First Search)방법으로 검사를 진행한다. 진행순서는 처음에 quasi-identifier의 개별 애트리뷰트에 대하여 frequency set을 구한 뒤 애트리뷰트들이 k-anonymity을 만족하는지를 확인한다. 만족하는 노드들에 대하여 노드들의 조합을 이용하여 2-subset 격자(lattice)를 생성하고 그 격자에 대하여 위에서 설명한 3가지 속성을 이용하여 k-anonymity 적합성 여부를 판단한다. 2단계에서 검사를 통과한 애트리뷰트 집합을 이용하여 3-subset 격자를 구성하고 k-anonymity 검사를 한다. 이와 같은 과정이 quasi-identifier의 모든 애트리뷰트 조합이 될 때까지 반복되게 된다.

표-1을 사용하여 Quasi-identifier가 (Birth, Sex, Zipcode) 2-anonymity를 만족는지 확인해보자. 첫번째 단계로 1-subset 즉 생년월일, 성별, 지역별(B0, S0, Z0)로 frequency set을 구한다

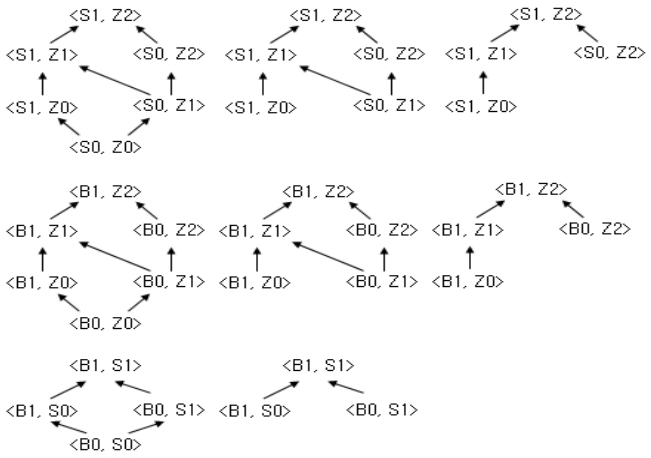
생년월일	count
1/21/76	2
4/13/86	2
2/28/76	2

성별	count
Male	3
Female	3

지역	count
53703	2
53706	2
53715	2

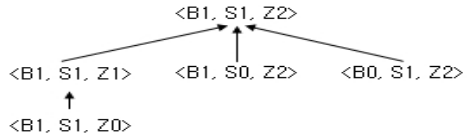
(그림 3) 1-subset freqeuncy set

각 attribue별로 frequency-set이 k(=2)를 만족한다 (F>=2). 따라서 1-subset 검사에서 통과한 노드들(B0, S0, Z0)을 조합하여 2-subset에 대한 검사를 수행한다. 이때 검사는 1-subset검사에서 통과한 노드들을 격자형태로 조합하므로써 수행되어진다.



(그림 4) 2-subset generalization lattice

<S0, Z0>를 루트로하는 격자를 예로 들어보자. <S0, Z0>의 경우 count가 1인(<Male, 53715>, <Female, 53715>) frequent set이 있기 때문에 k-anonymity을 만족하지 못한다. 그 다음 노드인 <S1, Z0>는 모두 count가 2보다 크기 때문에 k-anonymity을 만족한다. 이때 앞에서 정의한 generalization property 성질에 의하여 <S1, Z0>의 일반화된 노드(<S1, Z1>, <S1, Z2>) 역시 k-anonymity을 만족하므로 이 2개의 노드에 대해서는 검사를 생략하게 된다. <S0, Z1>을 검사한 결과 count가 1인(<Male, 5371*>, <Female, 5371*>) frequent set이 있기 때문에 <S0, Z1> 역시 제거되고 그것의 일반화된 노드인 <S0, Z2>를 검사하게 된다. 이때 이 노드는 k-anonymity을 만족하므로 남겨둔다. 이런 방식으로 <B0, Z0>, <B0, S0>를 검사하면 2-subset에서 k-anonymity을 만족하는 격자만이 남게 된다. 결과로 나온 격자의 루트를 이용하여 3-subset에 대한 격자를 구성하면 아래그림과 같이 된다.



(그림 5) 3-subset generalization lattice

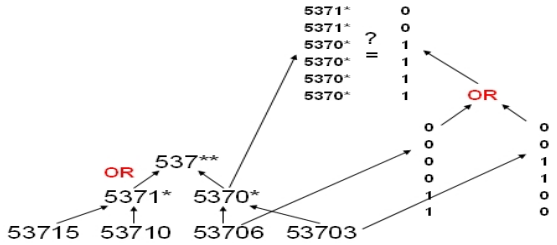
구성된 격자의 각 루트들에 대하여 k-anonymity 검사를 해주면 우리가 요구했던 quasi-identifier 애트리뷰트 set(생년월일, 성별, 지역)에 대한 모든 검사가 완료된다. 이처럼 Incognito는 검사->확장->검사->확장이라는 반복적인 과정을 통하여 모든 quasi-identifier attribute set에 대하여 k-anonymity 검사를 수행하는 장점이 있다.

3. bitmap을 이용한 Incognito의 성능향상

일반적으로 bitmap은 해당데이터의 값의 다양성(distinct value)이 작을수록 좋은 성능을 나타낸다.[6] 그러나 유일한(unique)한 경우에도 bitmap은 일반적인 B*tree보다 나쁘지 않은 성능을 나타낸다.[8] 그리고 bitmap의 경우 AND, OR과 같은 논리연산시 빠른 수행능력을 가졌다는 장점을 가진다.[1] 우리는 이에 착안하여 quasi-identifier attribute set에 포함된 각 애트리뷰트들을 대상으로 bitmap index를 생성하여 k-anonymity을 검사하는 과정에서(generalization과 격자생성) 사용한다. (그림-4)와 (그림-5)에서 k-anonymity 확인을 위해 각 격자의 노드마다 frequency set을 구해야하는데 이때 노드검사시 최소한 한번의 DB스캔 또는 인덱스 스캔이 일어나야 한다. 그러나 bitmap을 사용할 경우 직접적인 DB 스캔없이 quasi-identifier에 속하는 애트리뷰트의 해당 bitmap index만을 접근하여, AND 또는 OR 연산을 해주므로써 일반화, 격자조합생성이 가능하게 된다. 이러한 점을 이용하여 incognito의 성능을 개선시킬 수 있다. 그리고 추가적으로 이전단계에서(하위단계)에서 구성된 비트맵을 임시저장/이용하여 더욱빠른 일반화작업이 가능해진다. 추가적으로 bitmap을 이용할 경우 즉각적인 frequency set count를 알 수 있으므로 추후에 발생될 불필요한 연산과정을 생략할 수 있다는 또 다른 장점이 있다.

3.1 Bitmap OR 논리연산을 이용한generalization

일반화란 앞서 밝혔듯이 해당 값보다 더 일반적인 상위 값을 의미한다. 일반화는 일반화 범위에 포함되는 하위 값들의 bitmap OR 연산을 통하여 구할 수 있다. 투표 등록자 테이블의 Zipcode 애트리뷰트를 통한 예를 보도록 하자. 53703과 53706의 일반화 5370*의 bitmap(001111)은 53703의 bitmap(001100)과 53706(000011)의 OR연산을 통하여 구할 수 있음을 알 수 있다.

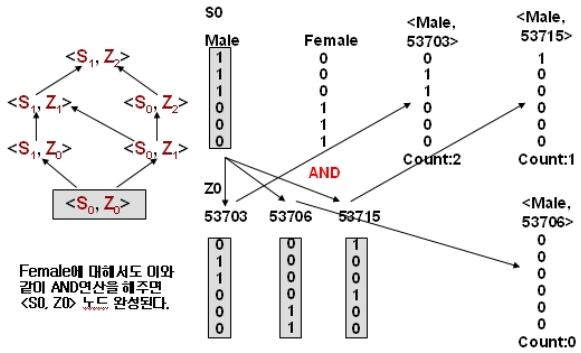


(그림 6) OR 논리연산을 이용한 generalization

3.2 Bitmap AND 논리연산을 사용한 격자구성

(그림-4)를 보면 2-subset일 경우 <S0, Z0>, <B0, Z0>, <B0, S0>와 같이 2개의 애트리뷰트가 조합되어져 검사에 사용된다. 1-subset에서 검사/통과 되어진 노드(B0, S0, Z0)를 사용하여 2-subset에서 사용될 조합을 결정하고 그것(<S0, Z0>, <B0, Z0>, <B0, S0>)들을 루트로 사용하는 격자를 구성한 뒤 k-anonymity을 만족하는지 여부를 검사하게 된다. 이때 bitmap을 사용하지 않는 incognito라면 격자의 루트를 구성할 때 DB스캔 또는 인덱스스캔을 해야하지만 bitmap을 이용할

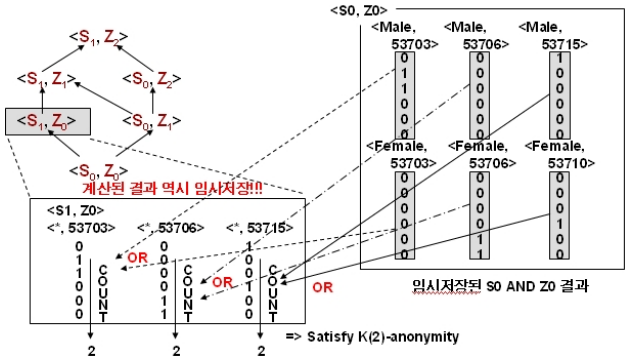
경우 두 컬럼에 생성된 bitmap에 AND 논리연산을 해주므로써 구할 수 있게 된다(그림-6).



(그림 7) bitmap 논리연산을 이용한 격자구성

3.3 임시저장된 Bitmap을 이용한 일반화의 간략화

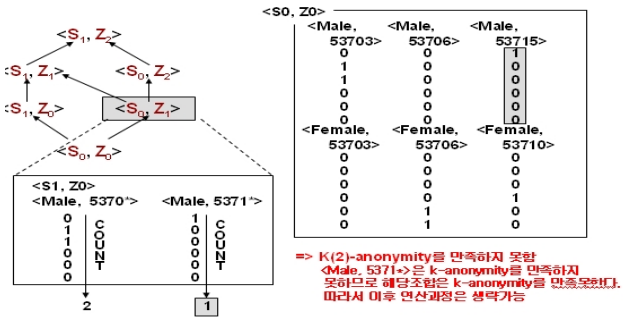
격자의 각 노드에 대하여 k-anonymity을 검사할 때 노드마다 새로 frequency set count를 구해야한다. 이때 실제 DB 또는 인덱스 스캔이 발생하게 된다. 상위레벨의 노드들은 하위레벨 노드들에 대하여 직접적인 일반화관계에 있다. 그렇기 때문에 이전레벨에서(하위레벨에서) 구한 bitmap을 임시저장하고 있다면 위에서 설명한 OR 논리연산을 이용하여 더 빠르게 해당 노드의 일반화를 구할 수 있게 된다. <S1, Z0>의 경우 <S0, Z0>에서 구한 값들을 Z1에 대하여 OR연산을 통하여 일반화 구할 수 있다(그림-9)



(그림 9) 임시bitmap을 이용한 generalization의 간략화

3.3 임시저장된 Bitmap을 이용한 일반화의 간략화

이 논문에서 제시한 bitmap을 이용한 상위레벨로의 일반화는 각 bitmap index에 대한 OR연산들의 집합이므로 집합 대 집합의 연산이 아니다. 따라서 각 연산들은 serial하게 또는 parallel하게 처리가 가능하다. 이때 각 연산작업 후 처리된 결과를 count하고 이 결과가 size k를 만족하는지 그렇지 않은지는 즉시 판단할 수 있다. 만약 count가 size k를 만족하지 않는다면 해당노드는 k-anonymity를 만족하지 않는다는 의미가 되므로 이후의 연산과정은 생략 가능하다. (그림-10)을 보면 <Male, 5371*>에 대하여 count가 1이므로 k-anonymity을 만족하지 못한다. 따라서 이후의 연산과정은 생략가능하다.



(그림 10) bitmap count를 이용한 불필요한 과정의 생략

4. 결론 및 향후연구과제

Incognito는 조인공격에 대하여 size k를 만족하는 모든 애트리뷰트의 집합을 구할 수 있게 해준다. 여기서 incognito는 기본적으로 group by, count, join과정의 반복으로 모든 애트리뷰트의 집합을 구한다. 이때 bit의 배열로 구성된 bitmap index를 incognito 알고리즘에 적용하면 실제 DB 또는 인덱스로의 접근없이 상위레벨 즉 일반화를 구할 수 있게된다(OR연산). 또한 격자에서 루트를 생성할 때 bitmap들을 AND연산하면 그 결과로서 격자의 루트가 생성된다. 격자에서 상위레벨로의 일반화가 필요할 때는 이전단계에서 구하고 임시저장한 bitmap들에 대하여 OR연산을 통하여 구할 수 있게된다.

향후 연구과제로는 실제로 incognito를 구현하고 이 논문에서 제시한 bitmap index를 이용한 기법들을 적용해보고 적절한 성능개선이 일어나는지를 관찰하는 것이다.

참고문헌

[1] ch5. scheme object in Oracle 10g concept book from http://otn.oracle.com

[2] Oracle 10g Data Warehousing guide from http://otn.oracle.com

[3] Kristen LeFevre, David J.DeWitt, Raghu Ramakrishnan "Incognito: Efficient Full-domain K-Anonymity" SIGMOD 2005 June

[4] P.samarati. Protecting respondanats' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering, 13(6), November/December 2001.

[5] P.Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory, 1998.

[6] Patrick O'Neil, Dallan Quass, Improved Query Performance with Variant Indexes SIGMOD 97

[7] Tadeusz Morzy, Maciej Zakrzewicz, Group Bitmap Index: A structure for association rules retrieval

[8] Vivek Sharma, Bitmap Index vs. B-tree Index: Which and When?, http://www.oracle.com/technology/pub/articles/sharma_indexes.html