KIPS Trans. Softw. and Data Eng.
Vol.11, No.4 pp.157~162
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2022.11.4.157

A Method of Reducing the Processing Cost of Similarity Queries in Databases

Sunkyung Kim† · Ji Su Park†† · Jin Gon Shon†††

ABSTRACT

Today, most data is stored in a database (DB). In the DB environment, the users requests the DB to find the data they wants. Similarity Query has predicate that explained by a similarity. However, in the process of processing the similarity query, it is difficult to use an index that can reduce the range of processed records, so the cost of calculating the similarity for all records in the table is high each time. To solve this problem, this paper defines a lightweight similarity function. The lightweight similarity function has lower data filtering accuracy than the similarity function, but consumes less cost than the similarity function. We present a method for reducing similarity query processing cost by using the lightweight similarity function features. Then, Chebyshev distance is presented as a lightweight similarity function to the Euclidean distance function, and the processing cost of a query using the existing similarity function and a query using the lightweight similarity function is compared. And through experiments, it is confirmed that the similarity query processing cost is reduced when Chebyshev distance is applied as a lightweight similarity function for Euclidean similarity.

Keywords : Similarity, Lightweight Similarity, Similarity Query, Database

데이터베이스에서 유사도 질의 처리 비용 감소 방법

김 선 경† · 박 지 수†† · 손 진 곤†††

요 약

오늘날 대부분의 데이터는 데이터베이스(database: DB)에 저장된다. 이러한 DB 환경에서 사용자는 자신이 원하는 데이터를 찾아줄 것을 DB에게 요청하게 된다. DB 질의 중 유사도 질의는 DB 사용자가 원하는 조건으로 유사도가 포함되어 있는 것을 말한다. 그러나 유사도 질의를 처리하기 위한 과정은 처리 레코드의 범위를 줄일 수 있는 색인을 이용하기 힘들어 테이블의 전체 레코드에 대해서 매번 유사도를 계산하는 비용이 높다. 본 논문은 이러한 문제점을 해결하기 위하여 경량 유사도 함수를 정의한다. 경량 유사도 함수는 유사도 함수에 비해 데이터를 여과하는 정확도는 떨어지지만 비용이 유사도 함수에 비하여 적게 소모되는 특징이 있다. 이러한 경량 유사도 함수의 특징을 이용하여 유사도 질의 처리 비용 감소 방법을 제시한다. 그리고 유클리드 거리 함수에 경량 유사도 함수로 체비쇼프 거리를 제시하고 기존의 유사도 함수를 이용하는 질의와 경량 유사도 함수를 이용하는 질의의 처리 비용을 비교한다. 그리고 실험을 통하여 유클리드 유사도에 대한 경량 유사도 함수로 체비쇼프 거리를 적용하였을 때 유사도 질의 처리 비용이 감소하는 것을 확인한다.

키워드 : 유사도, 경량 유사도, 유사도 질의, 데이터베이스

1. 서 론

지난 20년간 데이터의 양은 매우 빠른 속도로 증가하고 있다. 정보기술 분야 전문 시장조사 기관인 인터내셔널 데이터 코퍼레이션(International Data Corporation, IDC)에 따르면 전 세계 데이터는 2018년 약 33 제타바이트(Zettabyte, ZB)이며 2025년에는 약 5배인 175 제타바이트로 증가할 것으로 예상한다[1]. 데이터양의 엄청난 증가로 인하여 데이터를 저장하고 처리하는 데이터베이스의 중요성은 점점 커지고 있다[2]. 그리고 여러 분야에서 다양한 형태의 데이터가 저장되고 있고 데이터를 추출하는 방식도 점점 복잡해지고 있다. 데이터를 추출하는 방식이 복잡할수록 연산 비용이 증가한다. 이러한 복잡한 추출 방식 중 하나가 유사도 질의 처리이다[3].

질의란 임의의 조건을 만족하는 데이터 집합을 사용자가 데이터베이스에 요청하는 것이다. 이 중 조건에 유사도가 포함된 질의를 유사도 질의라 한다. 그리고 유사도 질의에 대해

※ 이 논문은 한국방송통신대학교 학술연구비 지원을 받아 작성된 것임.
† 준 회 원 : 메가존클라우드㈜ DB Architect 팀장
†† 종신회원 : 전주대학교 컴퓨터공학과 교수
††† 종신회원 : 한국방송통신대학교 컴퓨터과학과 교수
Manuscript Received : November 29, 2021
Accepted : January 14, 2022
* Corresponding Author : Jin Gon Shon(jgshon@knou.ac.kr)

Input: Compare Field f_c Compare Value c, Threshold τ, Table T
Output: Table E
1: $E \leftarrow \varnothing$
2: **foreach** $r \in T$ **do**
3: **if** $s(r.f_c, c) \geq \tau$ **then**
4: $E \leftarrow E \cup r$.
5: **end if**
6: **end do**

Fig. 1. Algorithm of Range Query

서 데이터베이스에서 응답하는 일련의 과정을 유사도 질의 처리라 한다. 유사도 질의 처리에서는 유사도를 이용하며 유사도는 두 대상 간 유사성을 계산한 값이다[4]. 대표적인 유사도로 유클리드 공간에서 두 점간 거리인 유클리드 거리가 있다.

유사도 질의를 처리하기 위해서는 유사도를 반복 계산하여야 한다. 계산된 유사도와 특정 임계값을 비교하게 되는데 이 경우 색인 사용이 어려워 전체 레코드 수만큼 유사도 계산이 필요하다. 특히, 유사도 계산은 제곱근과 같은 비용이 많이 드는 연산이 포함되어 있다[5]. 따라서 반복적인 유사도 계산이 유사도 질의 처리 비용을 높이는 원인이다[6].

본 연구는 데이터베이스에서 유사도 질의 처리 비용을 감소시키기 위하여 경량 유사도 함수를 제시하였다. 경량 유사도 함수는 여과와 저비용 조건을 만족해야 한다. 여과 조건은 경량 유사도를 사용하여 선택된 레코드가 기존 유사도를 사용하여 선택되는 모든 레코드를 포함해야하는 조건이다. 저비용 조건은 경량 유사도 연산 비용이 기존 유사도보다 작아야 하는 조건이다. 제3장에서 이 두 가지 조건에 대해서 수학적으로 정의하였다. 그리고 두 가지 조건에 의해 경량 유사도 함수가 어떻게 유사도 질의 비용을 감소시키는지를 설명하였다. 제4장에서 유클리드 거리에 대한 경량 유사도를 제시하였다. 제시한 경량 유사도가 두 가지 조건을 만족시킴을 이론적으로 분석하였다. 그리고 실험을 통하여 실제로 비용이 감소함을 확인하였다.

본 연구에서 경량 유사도를 제시하여 데이터베이스에서 유사도 질의 처리 비용을 감소시키는 근본 원리를 제시하는 기여를 하였다. 그리고 대표적인 유사도인 유클리드 유사도의 경량 유사도를 제시하여 유클리드 거리를 사용하는 유사도 질의 비용을 줄이는데 기여하였을 뿐만 아니라 앞에 제시된 경량 유사도가 효과적임을 확인하는데 기여하였다.

2. 관련 연구

2.1 유사도 질의

데이터베이스에서 질의 중 조건에 유사도가 포함된 질의를 유사도 질의라 한다. 유사도 질의는 범위 질의와 유사도 조인이 있다[7]. 이 중 범위 질의는 대표적인 유사도 질의이다. 범위 질의란 임의의 테이블에서 레코드의 특정 컬럼 값과 비교하고자 하는 값 사이의 유사도가 임계값 τ 이상인 레코드 집합을 추출하는 것이다. 범위 질의를 수학적인 방법으로 설명하면 다음과 같다. 레코드 r은 n개의 필드$(f_1, f_2, f_3, \ldots, f_n)$로 구성된다. 테이블 T는 m개의 레코드의 집합$\{r_1, r_2, r_3, \ldots, r_m\}$이다. i번째 레코드의 j번째 필드를 $r_i.f_j$로 표현한다. 정의 1은 범위 질의를 수학적 정의로 기술하였다.

정의 1. (범위 질의) 유사도 함수 s, 거리 함수 d, 테이블 T의 레코드 q, 비교 필드 f_c, 비교값 c, 임계값 τ가 주어졌을 때 집합 $E = \{q | q \in T$이고$s(q.f_c, c) \geq \tau\}$ 추출을 요청하는 것을 범위 질의고 한다.
또는 $E = \{q | q \in T$이고$d(q.f_c, c) \leq \tau\}$ 추출을 요청하는 것을 말한다.

유사도 함수 및 거리 함수에 대한 엄밀한 정의는 2.2절에서 하겠다.

범위 질의를 처리하기 위한 알고리즘은 Fig. 1과 같다. 1행에서 테이블 E를 공집합으로 초기화하고 2행에서 6행까지 테이블 T의 각 레코드 r의 비교 필드 f_c와 비교값 c 사이의 유사도를 유사도 함수 s를 이용하여 계산한 후 임계값 τ보다 유사도가 크거나 같으면 레코드 r을 테이블 E에 포함시킨다. 알고리즘을 보면 추가적인 도구가 없을 경우 범위 질의 처리를 위하여 테이블 T가 가지는 레코드 수인 m번만큼의 유사도 연산을 하여야 범위 질의 처리를 할 수 있음을 알 수 있다. 최근 데이터베이스에서 테이블이 가지는 레코드 수는 수십억 건을 넘는 경우가 매우 흔하다[8]. 이런 경우 유사도 연산을 수십억 번 해야 하며 매우 높은 비용이 소모된다. 이로 인하여 한정된 자원에서 데이터베이스의 원래의 목적인 사용자가 원하는 시간 내에 정보를 얻는 것이 불가능하다. 따라서 반복되는 유사도 연산의 비용을 줄여서 유사도 질의의 전체 비용을 줄일 필요성이 있다.

2.2 유사도

유사도는 수학적으로 거리의 반대 개념으로 유도된다. 따라서 유사도의 개념을 논하기 전에 거리에 대해서 먼저 설명한 후 유사도를 설명하겠다. 거리는 거리 함수에 의해서 계산된다. 임의의 집합 S에 대해서 거리함수 $d: S \times S \rightarrow \mathbb{R}$는 다음 조건을 만족하는 함수이다[7].

임의의 $x, y \in S$ 대하여
$d(x,y) \geq 0$ (nonnegativity);
$d(x,y) = d(y,x)$ (symmetry);
$d(x,x) = 0$ (reflexivity).

거리는 집합 S의 두 원소가 얼마만큼 다른지를 실수로 나타내는 것이다. 거리가 클수록 두 원소는 더 다르다는 것을

의미한다. 그러므로 동일한 두 원소의 거리는 0이 되어야 한다. 왜냐하면 동일한 두 원소는 전혀 다르지 않고 이것은 수치적으로 다름을 표현하면 0이기 때문이다. 이것을 reflexivity 조건이라 한다. 모든 원소의 쌍의 다른 정도는 동일한 두 원소의 다른 정도보다 크거나 같음이 자명하다. 따라서 임의의 두 원소의 거리는 항상 0보다 크거나 같아야 한다. 이 조건을 수학적으로 기술한 것이 nonnegativity 조건이다. 직관적으로 거리는 두 원소의 순서를 바꿔서 계산해도 같아야 한다. 이러한 거리의 특성을 symmetry 조건이 기술하고 있다.

거리의 반대 개념이 유사도이다. 즉, 두 원소가 얼마만큼 비슷한지를 실수로 나타내는 것이다. 유사도는 유사도 함수에 의해서 계산된다. 임의의 집합 S에 대해 유사도 함수 $s: S\times S\rightarrow \mathbb{R}$는 nonnegativity, symmetric 조건과 다음 조건을 만족하는 함수이다[9].

임의의 $x, y\in S$ 대하여
$s(x,y) \leq s(x,x)$ (inequality).

비슷한 정도를 판단할 때 동일한 두 원소를 비교한 것보다 더 비슷한 것은 있을 수 없다. 이러한 특성을 inequality 조건이 기술하고 있다. 많은 경우 거리와 유사도를 구분하지 않고 유사도라 말한다[10]. 본 논문에서는 거리와 유사도 구분하지 않고 유사도로 말하며 구분이 필요한 경우에만 거리라고 하겠다.

3. 경량 유사도 함수

3.1 경량 유사도 함수 개념 및 정의

유사도 질의를 처리할 때 비용이 단순 질의에 비하여 많이 소모됨을 2.1절에서 확인했다. 따라서 유사도 질의 처리할 때 비용을 감소시킬 수 있는 방법을 고안할 필요가 있다. 본 절에서는 유사도 질의 처리 비용을 감소시킬 수 있는 방법으로 경량 유사도 함수를 제시한다. 정의 2는 경량 유사도 함수에 대해 수학적으로 기술하였다. s와 s_l은 유사도 함수이다. $C(s)$는 유사도 함수 s의 연산 비용을 나타낸다.

정의 2. (경량 유사도 함수) 아래의 조건을 만족하면 s_l을 s의 경량 유사도 함수라 한다.
$\forall x\in X, s_l(x,c) \geq s(x,c)$ (여과);
$C(s_l) < C(s)$ (저비용).

s_l이 s의 경량 유사도 함수일 때 반대 개념으로 s는 중량 유사도 함수라고 부르겠다. 중량 유사도 함수의 계산 결과는 경량 유사도 함수의 계산 결과보다 여과 조건에 의해 항상 작거나 같음이 보장된다. 따라서 어떤 비교 쌍의 중량 유사도 함수의 결과 값보다 작다면 그 비교 쌍의 경량 유사도 함수 계산 결과 값보다 작음이 보장된다. 따라서 유사도가 임계값보다 크거나 같은 객체 쌍을 찾기 위해 중량 유사도 대신 경량 유사도 함수를 사용하면 임계값보다 유사도가 작은 것으로 판단하여 걸러낸 쌍 중에는 중량 유사도 함수를 사용하여 조건을 만족하는 쌍이 없음이 보장된다. 하지만 원래의 의도된 유사도가 중량 유사도 함수에 의해 계산된 값이라면 경량 유사도를 이용하면 유사도가 임계값을 넘지 못하는 쌍에 대해서도 경량 유사도 함수로 계산하면 유사도가 임계값을 넘게 된다. 이 부분은 경량 유사도 함수를 사용함으로써 발생하는 오류이다. 즉, 경량 유사도 함수를 사용하면 연산 비용을 줄일 수 있는 장점을 가진 반면 정확도가 떨어지는 결과를 얻는 단점도 가진다.

다음 절에서 경량 유사도 함수를 이용하여 유사도 질의 처리 비용을 감소하는 원리에 대해서 설명하겠다.

3.2 유사도 질의 비용 감소 원리

범위 질의를 할 때 경량 유사도 함수 사용에 대해서 설명하겠다. 비교값 c와 중량 유사도를 계산한 값이 τ보다 크거나 같은 어떤 x에 대해서 경량 유사도 함수로 계산한 값이 중량 유사도 함수로 계산한 값보다 항상 크거나 같으므로 경량 유사도 함수로 계산한 값은 τ보다 크거나 같다. 그리고 비교값 c와 중량 유사도를 계산한 값이 τ보다 작은 어떤 x에 대해서 비교값 c와 경량 유사도 함수로 계산한 값은 τ와 대소 관계를 알 수 없다. 이를 표로 정리하면 Table 1과 같다. 원래의 범위 질의에서 추출을 원하는 Case는 (a)이다. 하지만, 경량 유사도 함수를 이용하여 레코드를 추출하면 Case (a), Case (b) 조건을 만족하는 레코드를 추출하게 된다. 그리고 경량 유사도 함수를 이용하여 여과되는 경우는 Case (c)이며 이 경우는 중량 유사도에서도 동일하게 여과되는 조건이다. 따라서 유사도 범위 질의에서 중량 유사도 함수 대신 경량 유사도 함수를 이용하여 질의를 처리하면 추출한 레코드 집합은 중량 유사도 함수를 이용하여 추출한 집합을 포함한다. 원래의 범위 질의는 중량 유사도 함수를 이용하여 추출한 레코드를 원한다. 하지만 경량 유사도 함수를 이용하여 Case (b)와 같이 원하지 않은 레코드가 더 추출될 가능성이 있다. 하지만 경량 유사도 함수는 유사도 연산 비용이 적으므로 비록 추출할 레코드의 정확성은 떨어지지만 적은 비용으로 결과를 얻은 장점이 있다. 그리고 경량 유사도 함수를 이용하여 추출한 레코드 집합에 대해서 중량 유사도 함수를 이용하여 원하는 데이터를 추출할 수 있다. 추가적인 비용이 소모되지만 경량 유사도 함수를 이용하여 감소시킨 비용 보다 추가된 비용이 적다면 전체적인 비용은 줄어든다. 따라서 유사도에 대한 적

Table. 1 Compare of Light and Weight Similarity

Case	Light Similarity s_l	Weight Similarity s
(a)	$s_l(x,c) \geq \tau$	$s(x,c) \geq \tau$
(b)	$s_l(x,c) \geq \tau$	$s(x,c) < \tau$
(c)	$s_l(x,c) < \tau$	$s(x,c) < \tau$

Input: Compare Field f_c Compare Value c, Threshold τ, Table T
Output: Table E
1: $E \leftarrow \varnothing$, $E_l \leftarrow \varnothing$
2: **foreach** $r \in T$ **do**
3: **if** $s_l(r.f_c, c) \geq \tau$ **then**
4: $E_l \leftarrow E_l \cup r$.
5: **end if**
6: **end do**
7: **foreach** $r \in E_l$ **do**
8: **if** $s(r.f_c, c) \geq \tau$ **then**
9: $E \leftarrow E \cup r$.
10: **end if**
11: **end do**

Fig. 2. Algorithm of Range Query using Lightweight Similarity

절한 경량 유사도를 고안한다면 유사도 질의 비용을 감소시킬 수 있다.

경량 유사도 함수를 이용하여 범위 질의 처리 알고리즘은 Fig. 2이다. 1행에서 테이블 E와 E_l을 공집합으로 초기화한다. 2-6행에서 경량 유사도 함수 s_l을 이용하여 범위 질의 처리를 하고 결과를 테이블 E_l에 저장한다. 7-11행에서 앞에서 경량 유사도 함수를 이용하여 추출한 테이블 E_l을 대상으로 중량 유사도 함수를 이용하여 범위 질의 처리를 하여 테이블 E를 얻는다. 2-6행은 비용이 낮은 경량 유사도 함수를 이용했으므로 범위 질의 처리 비용이 감소하는 것이 확실하다. 정확도를 높이기 위해서 수행한 7-11행은 추가적인 비용이 소모되지만 E_l의 레코드 수가 T에 비해서 충분히 작아졌다면 추가적인 비용이 적을 것이다. 데이터베이스에서 질의를 이용하여 추출하는 레코드 수의 비율은 매우 낮은 것이 일반적이다. 따라서 추가된 비용이 2-6행에서 감소한 비용보다 적을 가능성이 매우 높으므로 경량 유사도를 이용할 경우 정확도를 유지하며 비용을 감소시킨다.

4. 유클리드 거리 경량 유사도

4.1 유클리드 거리 경량 유사도 제시

유클리드 거리는 많은 분야에서 사용되며 유사도 질의에서도 많이 사용된다. 따라서 유클리드 거리 함수에 대한 경량 유사도 함수를 찾는 것은 매우 유용하다. 벡터의 각 성분끼리의 차의 절댓값 중 가장 큰 수를 체비쇼프 거리라고 한다. 본 논문에서 체비쇼프 거리를 유클리드 거리의 경량 유사도로 제시한다. 두 벡터 a, b의 체비쇼프 거리[11]는 Equation (1)과 같다.

$$d_{Cb}(a,b) = \max(|a_1 - b_1|, |a_2 - b_2|, ..., |a_n - b_n|) \quad (1)$$

제시한 체비쇼프 거리가 유클리드 거리의 경량 유사도가 됨을 증명하겠다. 첫 번째로 체비쇼프 거리가 정의 2의 여과 조건을 만족하는지 보자. 우선 유사도의 역수인 거리를 다루기 때문에 여과 조건의 부호는 반대가 된다. 두 벡터의 k차원 성분의 차이의 절댓값이 가장 큰 값이 할 때($k = \arg\max(|a_x - b_x|)$) 체비쇼프 거리는 $|a_k - b_k|$이며 유클리드 거리는 $\sqrt{|a_k - b_k|^2 + \alpha}$, $(\alpha = \sum_{i=1}^{n} |a_i - b_i|^2 - |a_k - b_k|^2, \alpha \geq 0)$이므로 체비쇼프 거리는 항상 유클리드 거리보다 작거나 같다. 따라서 여과 조건을 만족한다. 두 번째로 정의 2의 저비용 조건을 만족하는지 보자. 유사도 함수의 비용은 측정할 때 기본 연산인 합(addition), 차(subtraction), 곱(multiplication), 제(division), 제곱근(square root), 최댓값(max), 절댓값(abs) 등의 연산을 몇 회 해야 하는지와 수행한 각 기본 연산의 비용에 따라 결정된다. 합 연산의 비용을 1이라고 했을 때 각 기본 연산의 비용을 상대적인 값으로 계산하겠다. 대중적으로 많이 사용되는 x86시스템에서 실험된 값으로 근거로 각 기본 연산의 시간 비용을 정하겠다. 실험 결과 합, 차, 곱, 제의 연산 시간 비용은 거의 비슷했다[12]. 따라서 사칙 연산의 시간 비용은 1로 한다. 제곱근의 경우 약 합 연산에 비해 1.5배 시간 소모가 더 되었다. 최댓값은 사칙연산과 비슷한 시간 비용이 소모되었다. 절댓값의 경우 차 연산 후 간단한 부호 변경으로 이루어지므로 사칙연산과 동일하게 본다. 유클리드 거리는 n차원에서 n번의 차(差) 연산(비용 $1n$), n번의 곱 연산(비용 $1n$), $n-1$번의 합(合) 연산(비용 $1n-1$), 1번의 제곱근 연산(비용 1.5)이 필요하다. 따라서 앞에서 언급한 비용을 측정하는 기준에 따라 n차원에서 $3n+0.5$의 비용이 필요하다. 반면 체비쇼프 거리는 n번의 절댓값 연산(비용 $1n$)과 $n-1$번의 비교 연산(비용 $1n-1$)이 필요하다. 따라서 체비쇼프 거리의 연산 비용은 n차원에서 $2n-1$의 비용이 필요하다. 따라서 체비쇼프 거리를 연산하는 비용이 유클리드 거리를 연산하는 비용에 비해서 작으므로 저비용 조건을 만족한다. 따라서 체비쇼프 거리는 정의 2의 경량 유사도 함수 두 가지 조건을 모두 만족하므로 유클리드 거리 함수의 경량 유사도 함수이다.

4.2 유클리드 거리 경량 유사도 실험

1) 실험 환경

하드웨어 실험 환경은 2.9 GHz의 Intel(R) Core i7-10700 CPU와 64GB RAM, Oracle Linux 7.9 운영체제가 설치되어 있는 데스크톱에서 실험하였다. 데이터베이스는 오라클 데이터베이스 19.1.0을 이용하였다.

2) Dataset

유클리드 유사도 질의를 실험하기 위해서 한국지역정보 개발원에서 제공하는 주소와 위치 정보 데이터를 이용했다[13]. 제공하는 전국 주소 데이터 6,302,195건을 주소정보 테이블로 적재했다. 한국지역정보 개발원에서 제공하는 데이터는 건물명 및 건물용도까지 포함한 주소와 UTM(Universal Transverse

Table. 2 Queries for Testing Lightweight Similarity

Query	Query Statements
Q1	SELECT Province, City, ... ,Buiding_No FROM Address WHERE SQRT(POWER(XC-:X,2)+POWER(YC-:Y,2)) <= :C
Q2	SELECT Province, City, ... ,Buiding_No FROM (SELECT * FROM Address WHERE ABS(XC-:X)<=:C AND ABS(YC-:Y) <=:C) WHERE SQRT(POWER(XC-:X,2)+POWER(YC-:Y,2)) <= :C

Mercator) 좌표계 기준 주출입구 좌표 값을 포함하고 있다. UTM 좌표 값은 숫자는 거리 1미터를 의미한다[14].

3) 실험 방법

경량 유클리드 거리의 효과를 실험하기 위하여 두 가지 방법으로 질의 처리를 한 후 각각의 질의 처리를 했을 때의 응답시간을 오라클 SQL TRACE 도구를 이용하여 측정하였다[15]. 질의문은 Address 테이블에서 레코드 10건을 추출하여 그 지점을 기준으로 100m 이내 주소지를 찾는 것이다. 첫 번째 방법은 경량 유사도를 사용한 것과 비교하기 위해 중량 유사도만으로 검색하였다. 두 번째 방법은 경량 유사도를 이용하여 처리하고 정확도를 높이기 위해 추출된 레코드에 대해서 중량 유사도를 이용하여 추출하였다. SQL 인라인 뷰에서 경량 유사도를 사용하고 결과를 외부 질의에서 중량 유사도를 처리하였다. 첫 번째 방법을 Q1, 두 번째 방법을 Q2라고 하겠다. 질의문은 Table 2와 같다.

4) 실험 결과

실험 결과는 Table 3에 정리하였다. 실험 결과 모든 질의문을 무작위로 추출한 10개의 기준 지점에 대해서 실험 데이터에서 질의문별로 동일한 레코드를 추출하였다. 주소지가 밀집되어 있는 도시지역이나 주소지가 희소한 농촌 지역 모두 추출된 데이터의 건수가 질의문에 따라 차이가 없음을 확인하였다. 따라서 경량 유사도 함수를 사용하여도 유사도 질의 처리의 정확도는 변하지 않음을 확인하였다. 질의문을 처리하기 위해서 소요된 평균 시간은 Q1는 5.713초 Q2는 1.751초이다. Q1의 경우 6,302,195건의 레코드에 대해서 중량 유사도 연산을 하여 가장 소요된 시간이 가장 길었다. Q2는 경량 유사도 함수를 사용하여 중량 유사도 함수 사용 대비 약 69.4%를 시간 비용 감소 효과가 있었다.

5. 결 론

데이터베이스에서 점점 더 다양한 종류의 데이터를 관리함에 따라 질의도 단순 일치 조건이나 대소 비교 조건이 아닌 데이터의 유사성에 기반을 둔 질의가 필요해졌다. 그리고 데이터의 중요성이 점점 더 강조되면서 빅 데이터 처리에 대한 요구가 증가하고 있다. 빅 데이터 처리는 많은 양의 데이터를 처리하는 것도 의미하지만 다양한 형태의 데이터를 처리하는 것도 의미한다. 그리고 이러한 데이터 형태의 다양성 때문에 기존 단순 질의로는 원하는 것을 얻을 수 없게 되었다. 한 예로 위치 기반으로 검색하는 것은 유사도 질의 처리가 필요하다. 그러나 유사도 질의는 비용 효율 측면에서 많은 불리한 점을 가지고 있다. 본 논문은 유사도 질의 처리를 할 때 이용할 수 있는 경량 유사도 함수를 제시하고 가져야 할 조건을 정의하였다. 그리고 제시한 경량 유사도 함수가 유사도 질의 비용을 감소시키는 원리를 설명하였다. 추가적으로 경량 유사도 함수가 가지는 몇 가지 특성을 정의하였다. 제4장에서 유클리드 거리를 이용한 유사도 질의 처리에 이용할 수 있는 경량 유사도 함수를 제시하였다. 유클리드 거리는 체비쇼프 거리가 경량 유사도임을 증명하였다. 제시한 유사도가 경량 유사도의 두 가지 조건을 만족시키는 것을 확인하였다. 그리고 실험을 통하여 실제로 비용이 69.4% 감소하는 것을 확인하였다.

유사도 질의는 범의 질의 이외에 패턴 인식에서 널리 사용되는 최근접 이웃 검색 등도 있다. 본 논문에서 제시한 유사도 질의 처

Table. 3 The Experimental Results of the Lightweight Euclidean Similarity

X Coordinate	Y Coordinate	Number of Records	Elapsed Time(sec)	
			Q1	Q2
940,720	1,846,669	4	5.729	1.733
995,025	1,895,979	5	5.752	1.761
1,120,619	1,800,114	4	5.743	1.729
915,242	1,831,453	1	5.664	1.748
1,149,488	1,736,299	12	5.736	1.766
927,743	1,941,146	58	5.719	1.750
1,052,517	1,869,561	22	5.714	1.759
954,556	1,943,574	75	5.738	1.733
950,541	1,763,437	2	5.637	1.803
1,142,998	1,684,112	110	5.702	1.726
Average		29	5.713	1.751

리 감소 방법이 최근접 이웃 검색에서도 이용할 수 있는지에 대한 추가 연구가 필요하다. 그리고 코사인 유사도와 같이 많이 사용되는 유사도 함수에 대한 경량 유사도 함수 연구도 필요하다.

References

[1] D. Reinsel, J. Gantz, and J. Rydning, "Data age 2025: The evolution of data to life-critical," *International Data Corporation*, Retrieved 2, 2017.

[2] C. Shang and F. You "Data analytics and machine learning for smart process manufacturing: Recent advances and perspectives in the big data era," *Engineering*, Vol.5, Iss.6, pp.1010-1016, 2019.

[3] T. Kim et al., "Similarity query support in big data management systems," Information System, Vol.88, pp.101455, 2020.

[4] S. K. Vangipuram and R. Appusamy, "A survey on similarity measures and machine learning algorithms for classification and prediction," *International Conference on Data Science, E-learning and Information Systems*, pp.198-204, 2021.

[5] D. Ryu and J. Baik, "A comparative study on similarity measure techniques for cross-project defect prediction," *KIPS Transactions on Software and Data Engineering*, Vol.7, No.6, pp.205-220, 2017.

[6] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," *arXiv:1408.2927v1* [cs.DS], 2014.

[7] J. S. Park, "A similarity join algorithm using a median as a filter", KIPS *Transactions on Software and Data Engineering*, Vol.4, No.2, pp.71-76, 2014.

[8] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications*, Vol.19, No.2, pp.171-209, 2014.

[9] P. M. B. Vitányi, "Information distance in multiples," *IEEE Transactions on Information Theory*, Vol.57, No.4, pp.2451-2456, 2011.

[10] S. H. Cha, "Compresive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, Vol.1, No.4, pp.300-307, 2007.

[11] J. A. O'Keefe, "The universal transverse mercator grid and projection," The Professional Geographer, Vol.4, Iss.5, pp.19-24, 1952.

[12] elgw, Tests the speed of common mathematical operations [Internet], https://github.com/elgw/math_ops_speed, (downloaded 2021, Oct. 19).

[13] Ministry of the Interior and Safety, Location Information Summary DB [Internet], https://www.juso.go.kr/addrlink/addressBuildDevNew.do?menu=geodata, (downloaded 2021, Nov. 14).

[14] K. Kawase, "Concise derivation of extensive coordinate conversion formulae in the Gauss-Krüger projection," *Bulletin of the Geospatial Information Authority of Japan*, Vol.60, pp.1-6, 2013.

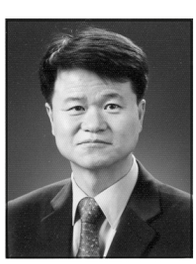
[15] H. K. Sharma, Mr. S.C. Nelson, "Explain plan and SQL trace the two approaches for RDBMS tuning," *Database Systems Journal*, Vol.8, No.1, pp.31-39, 2017.

김 선 경

https://orcid.org/0000-0003-3160-4993
e-mail : skkimic@gmail.com
1996년 고려대학교 전산학과(학사)
2020년 한국방송통신대학교 정보과학과 (석사)
2012년 ~ 2020년 ㈜위즈베이스 기술이사
2021년 ~ 현 재 메가존클라우드㈜ DB Architect 팀장
관심분야 : 유사도, 데이터베이스 포렌식, 데이터베이스 성능 최적화, 데이터 관리

박 지 수

https://orcid.org/0000-0001-9003-1131
e-mail : jisupark@jj.ac.kr
2013년 고려대학교 컴퓨터교육과(박사)
2015년 ~ 2018년 충남대학교 초빙교수
2020년 ~ 현 재 전주대학교 컴퓨터공학과 교수
2020년 ~ 2021년 한국정보처리학회 이사/JIPS 간사
2022년 ~ 현 재 JIPS Managing Editor
2022년 ~ 현 재 한국정보처리학회 상임부회장
2022년 ~ 현 재 KTSDE 편집위원장
관심분야 : 분산 시스템, 모바일 클라우드 컴퓨팅, e-Learning,



손 진 곤

https://orcid.org/0000-0002-0540-4640
e-mail : jgshon@knou.ac.kr
1991년 고려대학교 전산학전공(박사)
1991년 ~ 현 재 한국방송통신대학교 컴퓨터과학과 교수
1997년 ~ 1998년 State University of New York (Stony Brook) Visiting Professor
2000년 ~ 현 재 ISO/IEC JTC1/SC36 Korea Delegate
2009년 ~ 현 재 이러닝학회 부회장
2010년 한국정보처리학회 부회장
관심분야 : 컴퓨터통신망, 분산시스템, e-Learning, 정보기술 표준화