|  |
| --- |
| **익스포넨셜 메커니즘과 트랜짓 노드 기반의 차분 프라이버시가 적용된 최단 경로 탐색 문제[[1]](#footnote-1)** |
| 정재헌 함다인 박석  서강대학교 컴퓨터공학과  kinkin21c@u.sogang.ac.kr , denn@u.sogang.ac.kr, spark@sogang.ac.kr |
| Differential Private Shortest Path Problem using Exponential Mechanism and Transit Node |
| Jae Heon Jung, Dain Ham, Park Seog  Computer science and Engineering, Sogang University |
| **요 약**  최근 수집된 데이터들을 보호하기 위한 다양한 데이터 보호 기법이 제안되고 있다. 그 중에서도 차분 프라이버시는 데이터에 확률적으로 노이즈를 섞어 데이터를 숨김으로써 기존에 사용하던 보호 기법들과 다르게 공격자의 사전 지식 여부와 관계 없이 데이터를 보호할 수 있다. 그 중, 주로 사용되어 왔던 라플라스 메커니즘은 간편하지만, 이를 최단경로 문제에 적용하게 될 경우 기존의 효율적인 최단 경로 탐색을 그대로 사용할 수 없어 시간 복잡도가 크게 증가하게 된다. 이러한 이유로 본 논문에서는 도로의 혼잡도와 관련된 데이터 값을 이용해 간선의 가중치를 이산적으로 분할하고 이에 차분 프라이버시를 만족하는 익스포넨셜 메커니즘을 적용하는 방법을 제안한다. | | |

**1. 서 론**

최근 개인정보는 다양한 경로에서 수집되어 축적되고 있다. 이러한 데이터들을 보호하기 위해 K-익명화, L-다양성, T-근접성 같은 익명화[1,2]를 통한 보호 기법들이 제안되었으나 기존 기법들은 공격자의 사전 지식이 있을 경우에 취약한 모습을 보인다. 따라서 이를 보완하기 위해 데이터에 확률적인 노이즈를 섞어 데이터를 숨기는 차분 프라이버시 기법이 제시되었다. 차분 프라이버시는 확률적인 노이즈로 인해 공격자의 사전 지식과 관계없이 데이터를 안전하게 보호할 수 있다.

차분 프라이버시를 만족시키기 위해 일반적으로 라플라스 메커니즘이 사용되고 있다. 라플라스 메커니즘은 라플라스 분포를 사용해 데이터에 Lap(1/ε)의 노이즈를 삽입[7]한다. 이 기법을 최단경로 문제에 적용하게 될 경우 기존의 효율적인 최단 경로 탐색을 그대로 사용할 수 없어 시간 복잡도가 크게 증가하게 된다. 이를 보완하기 위해, 본 논문에서는 연속적인 값을 적절한 이산적인 값으로 분할하고, 차분 프라이버시를 만족하는 익스포넨셜 메커니즘을 적용하는 기법을 제안한다.

그래프를 실제 도로교통망에 적용할 경우, 간선은 도로를 나타내며, 도로 위에는 움직이는 차들이 존재하기 때문에 그 거리는 시간에 따라 연속적으로 변화한다. 따라서 이에 차분 프라이버시를 적용하는데 라플라스 메커니즘을 사용할 경우, 시간복잡도가 크게 증가하게 된다. 따라서, 본 논문에서는 간선의 가중치를 이산적으로 분할하여 이에 익스포넨셜 메커니즘을 적용하는 방법을 제안한다. 또한, 이러한 이산적인 분할에 따라 다양한 결과가 도출되기 때문에 이를 동적으로 계산하기 위하여 최단 거리 알고리즘에 주로 사용되어 왔던 다익스트라 알고리즘이 아닌 트랜짓 노드를 사용하여 최단 거리를 구하고자 한다.

**2. 관련 연구**

**2.1 트랜짓 노드(Transit node)**

트랜짓 노드는 일대일 최단거리를 구하는 데 불필요한 계산을 필요로 하는 다익스트라 알고리즘의 단점을 보완하여, 빠른 계산속도를 보장하는 알고리즘이다[5]. 트랜짓 노드는 자주 방문하는 노드를 트랜짓 노드로 지정하고, 트랜짓 노드 간 거리를 미리 계산하여 계산시간을 크게 단축시킬 수 있다.

**2.2 차분 프라이버시**

정의 1. . 모든 사건 에 대해 하나의 튜플만 다른 두 데이터베이스 를 입력 값으로 하는 랜덤성을 갖는 메커니즘 이 다음과 같은 식을 만족할 때 은 차분 프라이버시를 만족한다고 한다.[4]

이 때 일 시에는 -차분 프라이버시라고 한다. 그림 1은 각 데이터베이스의 출력 값의 분포에 대해 차분 프라이버시 관점에서의 차이 값을 나타낸다. 이처럼 차분 프라이버시는 이전의 비식별화 모델들과는 다르게 특정 속성에 대한 삭제 및 일반화를 수행하는 것이 아닌 해당 질의에 대해 특정 개인의 정보가 드러나지 않을 만큼의 랜덤성을 갖는 응답 값을 반환하는 메커니즘을 제공하는 것을 의미한다. 이때 추가되는 노이즈의 크기는 해당 질의 함수의 민감도에 의존하게 되며, 질의 함수에 대한 민감도의 정의는 다음과 같다.[9]

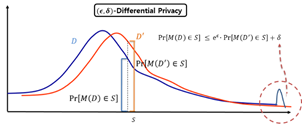


그림 1 차분 프라이버시 [6]

정의2. 민감도(Sensitivity). 질의 함수 및 이웃 데이터베이스 에 대하여 민감도 는 다음과 같이 계산된다.

**3. 본론**

**3.1 라플라스 메커니즘 (Laplace mechanism)**

어떠한 데이터베이스 에 대해 민감도가 인 질의 를 수행할 시의 라플라스 메커니즘 은 다음과 같은 식을 만족한다.

이 때 각각의 는 라플라스 분포 에서 독립적으로 균등하게(Independent and identically distributed) 선택되며 이를 수식으로 나타내면 이다. 이때 는 라플라스 분포의 확률밀도함수 를 따른다.[4]

**3.2 익스포넨셜 메커니즘(Exponential mechanism)**

데이터 전체집합 에서의 스코어 함수및 이웃 데이터베이스 와 에 대한 스코어 함수의 민감도를 라고 할 때 익스포넨셜 메커니즘 은 출력 집합 에 대해 출력 값 가 선택될 확률이 에 비례한다. 따라서 익스포넨셜 메커니즘 는 -차분 프라이버시를 만족한다. [4]

**3.3 차분 프라이버시가 적용된 트랜짓 노드**

본 논문에서는 최단 거리에 대한 알고리즘을 다음과 같이 구현하였다. 트랜짓 노드의 경우, 그래프를 비슷한 개수의 노드를 포함하고 있는 하위 그래프로 나누어, 해당 하위 그래프에서 다른 하위 그래프로 가기 위해 반드시 지나야 하는 노드를 트랜짓 노드로 지정하였다. 따라서 최단 거리는 다음의 식을 기준으로 계산하였다[5].

그래프에서 간선의 가중치는 도로의 혼잡도에 해당하며, 이를 이동시간으로 표현한다. 도로의 혼잡도는 도로 위의 차량 수에 비례하고, 시간에 따라 차량은 연속적으로 움직이기 때문에 혼잡도도 연속적으로 변화한다. 본 논문에서는 각 도로의 최대 혼잡도와 최소 혼잡도를 이용하여 도로의 혼잡 상태를 3 단계로 나누고, 다음과 같이 평균 값을 적용하여 가중치로 지정한다.

프로그램은 그래프와 시작점, 도착점에 해당하는 노드를 입력받는다. 익스포넨셜 메커니즘을 이용하여 혼잡도에 확률적으로 노이즈를 섞어 현재 혼잡도를 숨기고, 노이즈가 섞인 혼잡도에 해당하는 혼잡 상태를 지정한다. 이 혼잡 상태의 평균 값을 간선의 가중치, 즉 도로의 혼잡도라 계산하여 최단 경로를 구한다. 결과적으로 프로그램은 구한 경로를 노이즈가 섞이지 않은 기존 그래프에서 적용했을 때의 거리를 출력한다.

실험에서는 제안하는 메커니즘과 비교하는 메커니즘을 통해 구한 최단 거리의 경로를 이용하여 해당 경로와 실제 최단 경로와의 오차를 확인하고, 그 수행 시간을 비교한다.

**4. 실험 및 결과**

**4.1 실험 환경**

프로그램은 Python 2.7로 Linux 환경에서 구현되었다. 실험에 사용한 데이터는 서울시 강남의 지도 데이터로, 노드가 1194개인 그래프를 사용하였다.

**4.2 실험 방법**

본 실험에서는 우선 원활한 최단 거리 알고리즘의 수행을 위해 그래프 분할 수를 구하기 위한 실험을 선행하였다. 따라서 랜덤한 10개의 시작점과 도착점을 지정하여 그 평균값을 실험에 사용하여 최적의 그래프 분할 수를 찾고, 이후 입실론에 따른 두 메커니즘의 결과값을 비교하였다.

**4.3 실험 결과**

그림 2은 그래프 분할 수에 따른 실행 시간을 비교한 값이다. 입실론을 0.01, 0.1, 1, 10으로 설정하고 실험하여 그 평균값을 확인한 결과, 노이즈가 섞인 그래프와 노이즈가 섞이지 않은 그래프의 알고리즘 수행 시간 모두 그래프를 200개의 그래프로 분할했을 때 가장 빠른 속도를 보였다. 즉, 약 6개의 노드를 포함하고 있도록 그래프를 분할했을 때 수행 시간이 가장 빠름을 확인했다. 따라서 하위 실험들의 그래프 분할 수는 200으로 설정하고 실험하였다.

그림 3은 입실론 값에 따라 기존 그래프에서 라플라스, 익스포넨셜 메커니즘을 적용한 그래프와의 최단 거리 오차 값을 비교한 값이다. 그림 4의 수행 시간과 함께 비교했을 때, 익스 포넨셜 메커니즘을 적용한 알고리즘의 경우 라플라스 보다 더 큰 오차를 보이지만, 그림 4의 실행 시간을 비교해 보았을 때 익스포넨셜 메커니즘을 적용한 경우 실행 시간에서 약 1600배의 시간 차이가 나는 것을 확인할 수 있었다. 즉, 라플라스보다 오차가 약 1정도 크게 나타나지만, 시간복잡도 측면에서 익스포넨셜이 현저히 낮기 때문에 실시간으로 질의 처리가 이루어져야 하는 실제 응용환경에서는 제안 기법이 더 적합하다.

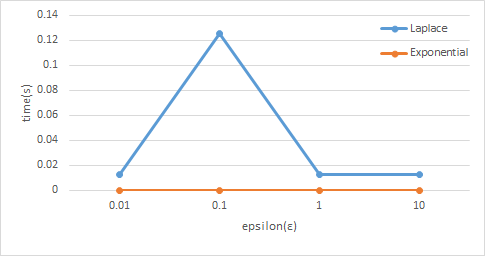


그림 4. 값에 따른 수행시간 변화

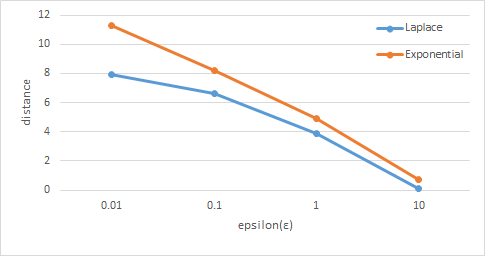


그림 3. 값에 따른 최단거리 오차 변화

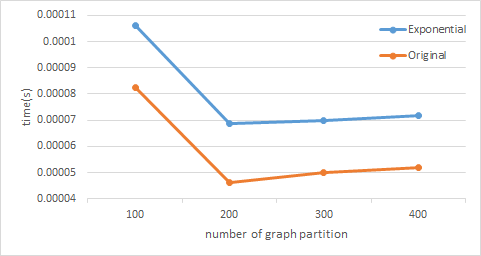


그림 2. 그래프 분할 수에 따른 수행시간

**4. 결 론**

본 논문에서는 최단 경로 탐색 문제에서 차분 프라이버시를 만족하기 위해 익스포넨셜 메커니즘을 트랜짓 노드 기법에 적용하였다. 기존에 사용하던 라플라스 메커니즘을 최단 경로 문제에 적용할 경우 연속적인 값을 내기 때문에 이를 이용해 부분적인 경로에 대한 거리를 미리 저장할 수 없다. 이를 보완하기 위해 제안 기법은 도로 구간의 이동시간을 3단계로 이산화하고 익스포넨셜 메커니즘을 적용한 트랜짓 노드 기법을 사용하였다. 실험을 통해 익스포넨셜 메커니즘을 적용하여 구한 최단 경로는 라플라스 메커니즘을 적용한 최단 경로와 적은 오차를 보이면서도 계산 시간 면에서 더 효율적인 결과를 보였다.

본 논문은 도로의 혼잡도를 원활, 보통, 혼잡으로 총 세 구간으로 나누었으나 좀 더 세세하게 혼잡도를 나누면 정확도 면에서 더 좋은 결과값을 낼 수 있을 것이다. 반면, 더 세세하게 나눌수록 시간 복잡도가 증가할 것으로 보인다.

**참고문헌**

[1] L. Sweeney, “k-anonymity: A model for protecting privacy,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 10, No. 5, pp. 557-570, 2002.

[2] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” TKDD, Vol. 1, No. 1, pp. 24-24, 2007.

[3] G. Hanjun, J. Woohwan, O. Seongwoong, K. Suyong, and Sh. Kyuseok, “Differentially Private k-Means Clustering based on Dynamic Space Partitioning using a Quad-Tree” in Journal of KIISE, Vol.45, No.3, pp.288-293, 2018.3

[4] K. Hyun-il, P. Cheolhee, H. Dowon, Ch. Daeseon, and Kongju National University, “A Study on Differentially Private Model for Financial Data\*” in Journal of The Korea Institute of Information Security & Crypto logy VOL.27, NO.6, Dec. 2017

[5] H. Bast, S. Funke, and D. Matijevic, “Transit—Ultrafast shortestpath queries with linear-time preprocessing,” in Proc. 9th DIMACS Implementation Challenge, 2006, pp. 175–192.

[6] K.Ligett, “Introduction to differential privacy, randomized response, basic properties,” The 7th BIU Winter School on Cryptography, BIU, 2017.

[7] C. Dwork, "Differential privacy," ICALP, pp. 1-12, 2006.

[8] C.Park, D.Hong, C.Seo “Differentially private data release method for general use of data,” Korea Computer Congress, pp.1036-1038, 2017.

[9] C.Dwork, A.Roth, “The algorithmic foundations of differential privacy,”Foundations and Trends® in Theoretical Computer Science, pp.211-407, 2014.

1. \* 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음(2015-0-00910) [↑](#footnote-ref-1)