동형암호를 이용한 대용량 데이터의 통계 분석 방법

강동우*, 원동호*
*성균관대학교 전자전기컴퓨터공학과
dwkang@security.re.kr dhwon@security.re.kr

Statistical analysis method of large data using homomorphic encryption

Dongwoo Kang*, Dongho Won*
*Sungkyunkwan University
Department of Electrical and Computer Engineering

요 약

동형암호를 이용한 통계 분석은 기존의 개인정보보호 문제로 수행할 수 없었던 데이터에 대해서 통계 분석이 가능하게 만든다. 본 논문에서는 대용량 데이터에 사용되는 대표적 통계 수치인 평균, 분산, 왜도, 첨도를 병렬처리를 사용하여 구하는 방법을 제안한다. 또한, 연산이 비교적 제한적인 동형암호에서도 통계적 수치를 구하기 위하여 동형암호문끼리의 뺄셈, 나눗셈, 제곱근 연산을 제안한다. 이를 통해, 분산된 대용량 데이터에 대해서도 동형암호를 통해 다양한 통계 연산이 가능할 것으로 기대된다.

1. 서론

최근 인터넷 통신량의 증가와 더불어 소셜 네트워킹의 광범위한 사용 및 인간의 사회적 행위가 증가하고 있다. 이 과정에서 부산물로 생성된 대량의 데이터는 기계학습이나 통계 등 여러 분야에서 사용되고 있다. 그러나, 데이터의 종류에 따라 개인정보를 포함하고 있는 민감한 데이터에 대한 취급과, 데이터가 분산되어 저장된 경우를 고려한 분산 데이터의 병렬처리 방안이 필요하다.

동형암호를 이용하면 개인정보를 포함하는 데이터에 대해 데이터 원문을 공개하지 않고 암호화를 한 상태로 직접 연산을 수행할 수 있다[1]. 그러나, 현재 동형암호의 경우 현재 덧셈, 곱셈 같은 간단한 연산밖에 지원하지 못하고, 대용량 데이터를 처리하는 경우 소요시간이 길어지는 문제점을 가지고 있다[2]. 본 논문에서는 동형암호를 이용하여 대용량 데이터에 대해 통계 분석을 시행할 때 사용할 수 있는 병렬처리 방식을 제안한다. 또한, 개인정보보호 및 연계 분석과 분산처리가 가능한 동형암호 연산방식을 제안한다.

2. 병렬처리를 이용한 대용량 데이터 통계 분석

대용량 데이터에 대한 통계 분석을 진행할 때, 데이터가 분산되어 저장되어 있거나 개인정보보호를 위해 동형암호화되어 저장된 경우 중앙서버로 전송하여 통계 분석을 시행한다. 이때, 데이터의 그룹을 나누어 병렬처리를 이용한다. 본 논문에서는 각각 n개의 데이터를 가지고 있는 m개의 그룹, 총 $m \times n$개의 데이터에 대해 평균, 분산, 첨도, 왜도를 병렬처리를 이용하여 구하는 방법을 제시한다. 또한, 통계 분석에서 사용할 수 있는 동형암호 기반의 산술연산을 제안한다. 다음에서 표현하는 x_{ij}는 i번째 그룹에 포함된 j번째 데이터를 의미한다.

2.1 평균

전체 데이터 $m \times n$개에 대한 평균을 구하는 공식은 $\bar{x} = \frac{1}{m \times n} \cdot \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}$이다. 평균을 병렬처리를 이용하여 구하는 방법은 다음과 같다.

Input : m개의 그룹마다 각각 분산 저장된 n개의 정수형 데이터

Algorithm :

1. 그룹별로 평균 $\overline{x_m} = \frac{1}{n} \cdot \sum_{j=1}^{n} x_{mj}$을 병렬처리를 이용하여 구한다.
2. 병렬처리로 구한 그룹별 평균값을 토대로 전체 데이터 평균인 $\bar{x} = \frac{1}{m} \cdot \sum_{i=1}^{m} \overline{x_i}$을 구한다.

Output : 전체 데이터 $m \times n$개에 대한 평균 $\bar{x}$

2.2 분산

전체 데이터 $m \times n$개에 대해 분산을 구하기 위해서는 평균값이 먼저 산출이 되어야 한다. 2.1절을 통해 평균값이 $\overline{x}$로 산출되고 평균이 각 그룹으로 전송되었다고 가정하였을 때, 전체 데이터 $m \times n$개에 대한 분산을 구하는 공식은 $\overline{v} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}-\overline{x})^2}{m \times n}$ 이다.

전체 데이터에 대한 분산을 병렬처리를 이용하여 구하는 방법은 다음과 같다. 또한, 도출한 분산의 양의 제곱근을 구하여 표준편차를 구할 수 있다.

Input : m개의 그룹마다 각각 분산 저장된 n개의 정수형 데이터, 전체 데이터의 평균 $\overline{x}$
Algorithm :
1. 그룹별로 분산 $\overline{v_m} = \frac{1}{n} \cdot \sum_{j=1}^{n}(x_{mj}-\overline{x})^2$을 병렬처리를 이용하여 구한다.
2. 병렬처리로 구한 그룹별 분산 값을 토대로 전체 데이터 분산인 $\overline{v} = \frac{1}{m} \cdot \sum_{i=1}^{m}\overline{v_i}$을 구한다.

Output : 전체 데이터 $m \times n$개에 대한 분산 $\overline{v}$

2.3 첨도

첨도는 데이터 분포의 뾰족한 정도를 나타내는 통계 척도로써, 데이터들이 어느 정도 집중적으로 중심에 몰려 있는지를 측정할 때 사용된다[3]. 첨도가 3에 가까울수록 데이터 분포가 정규분포에 가깝다는 것을 의미한다. 첨도 역시 평균값이 먼저 산출이 되어야 하며 전체 데이터 $m \times n$개에 대한 첨도를 구하는 공식은 다음과 같다.

$$K = m \times n \cdot \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}-\overline{x})^4}{(\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}-\overline{x})^2)^2}$$

전체 데이터에 대한 첨도를 병렬처리를 이용하여 구하는 방법은 다음과 같다.

Input : m개의 그룹마다 각각 분산 저장된 n개의 정수형 데이터, 전체 데이터의 평균 $\overline{x}$
Algorithm :
1. 그룹별로 $\overline{A_m} = \sum_{j=1}^{n}(x_{mj}-\overline{x})^2$, $\overline{B_m} = \sum_{j=1}^{n}(x_{mj}-\overline{x})^4$ 을 병렬처리를 통해 구한다.
2. 병렬처리로 구한 그룹별 $\overline{A_m}, \overline{B_m}$ 값을 토대로, 전체 데이터 첨도를 다음과 같이 계산한다.

$$K = m \times n \cdot \frac{\sum_{i=1}^{m}\overline{B_m}}{(\sum_{i=1}^{m}\overline{A_m})^2}$$

Output : 전체 데이터 $m \times n$개에 대한 첨도 K

2.4 왜도

왜도는 데이터 분포의 비대칭성을 나타내는 통계 척도이다. 왜도의 값은 양수나 음수가 될 수 있으며 정의되지 않을 수도 있다[3]. 왜도가 음수일 경우에는 데이터 분포에서 왼쪽 부분에 긴 꼬리를 가지며 중앙값을 포함한 자료가 오른쪽에 더 많이 분포해 있음을 의미하고 양수일 경우에는 중앙값을 포함한 자료가 왼쪽에 더 많이 분포해 있음을 의미한다. 왜도 역시 평균값이 먼저 산출이 되어야 하며 전체 데이터 $m \times n$개에 대한 왜도를 구하는 공식은 다음과 같다.

$$S = \sqrt{m \times n} \cdot \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}-\overline{x})^3}{(\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}-\overline{x})^2)^{\frac{3}{2}}}$$

이를 병렬처리를 이용하여 구하는 방법은 다음과 같다.

Input : m개의 그룹마다 각각 분산 저장된 n개의 정수형 데이터, 전체 데이터의 평균 $\overline{x}$
Algorithm :
1. 그룹별로 $\overline{A_m} = \sum_{j=1}^{n}(x_{mj}-\overline{x})^2$, $\overline{B_m} = \sum_{j=1}^{n}(x_{mj}-\overline{x})^3$ 을 병렬처리를 통해 구한다.
2. 병렬처리로 구한 그룹별 $\overline{A_m}, \overline{B_m}$ 값을 토대로, 전체 데이터 왜도를 다음과 같이 계산한다.

$$S = \sqrt{m \times n} \cdot \frac{\sum_{i=1}^{m}\overline{B_m}}{(\sum_{i=1}^{m}\overline{A_m})^{\frac{3}{2}}}$$

Output : 전체 데이터 $m \times n$개에 대한 왜도 S

3. 통계 분석을 위한 동형암호 산술 연산

3.1 동형암호의 개요

동형암호란 평문을 연산한 결괏값이 암호문을 연산한 결괏값을 복호화한 값과 같은 암호이다. 동형암호를 이용하면 사용자와 서버 간의 통신에서 서버에게 평문을 공개하지 않고서도 연산을 담당하게 할 수 있다.

$$Dec_k(f(c_1, c_2, \dots, c_n)) = f(m_1, m_2, \dots, m_n)$$

실제 컴퓨터가 하는 모든 연산은 and, or, not 등의 논리연산 합성으로 이루어져 있으므로, 세 가지 연산에 대해 동형암호가 제안될 수 있으면 모든 연산에 대해 동형암호가 성립된다[4].

3.2 동형암호에서 덧셈과 곱셈 외의 연산 처리

동형암호에서 덧셈과 곱셈에 대해서 암호문을 연산한 값을 복호화한 값은 평문을 연산한 값과 같다. 그러나, 이러한 동형암호의 가장 큰 한계는 정수나 실수 간의 덧셈과 곱셈 같은 매우 간단한 연산만을 지원한다는 것이다. 이런 한계점으로 인해 병렬처리를 이용한 통계 분석에서 사용되는 뺄셈, 나눗셈, 제곱근 등의 연산을 적용할 수 없다. 본 논문에서는 동형암호화된 실수형 자료의 덧셈, 곱셈 연산을 이용하여 2절에서 사용되는 산술 연산을 처리할 수 있도록 연구하였다.

3.2.1 동형암호에서의 뺄셈 연산

실수 단위의 동형암호에서의 뺄셈 연산은 뺄셈 연산자를 기준으로 우측 피연산자에 -1을 곱한 값을 더해서 구할 수 있다. 이를 알고리즘으로 표현하면 다음과 같다.

Input : 피연산자 a, b
Algorithm :
1. 사용자는 $E(a), E(b), E(-1)$을 서버로 전송한다.
2. 서버는 동형암호 곱셈 연산을 이용하여 $E(-b) = E(b) \times E(-1)$를 계산한다.
3. 서버는 동형암호 덧셈 연산을 이용하여 $E(a-b) = E(a) + E(-b)$를 계산하여 사용자에게 전송한다.
4. 사용자는 서버로부터 받은 $E(a-b)$을 복호화한다.
Output : $a-b$

3.2.2 동형암호에서의 나눗셈 연산

기본적으로, 나눗셈 연산은 나눗셈 연산자를 기준으로 우측 피연산자의 역수를 곱한 값으로 구할 수 있다. 즉, 동형암호에서의 나눗셈 연산은 임의의 암호화된 실수의 역수를 구할 수 있다면 연산 가능함을 알 수 있다.

임의 실수의 역수는 무한등비급수를 이용하여 근사치를 구할 수 있다. $0 < b < 2$를 만족하는 b에 대하여 $b^{-1} = \sum_{i=0}^{\infty}(1-b)^i$이 성립한다. 즉, 임의의 실수 b에 대해 $0 < b^{-1}k < 2$ 를 만족하는 전처리 상수 k를 설정하면 다음의 식이 도출된다.

$$b^{-1} = k^{-1}\sum_{i=0}^{n}(1-b\times k^{-1})^i = k^{-1-n}\sum_{i=0}^{n}k^{n-i}(k-b)^i$$

이를 알고리즘으로 표현하면 다음과 같다.

Input : a, b, 전처리 상수 k, 반복 횟수 n
Algorithm :
1. 사용자는 $\frac{b}{k}$의 값이 $(0,2)$이 되도록 전처리 상수 k를 결정한다.
2. 사용자는 $E(a), E(b), E(k), n$을 서버로 전송한다.
3. 서버는 다음을 계산하여 사용자에게 반환한다.

$$E(c) = E(a) \times \sum_{i=0}^{n}E(k)^{n-i}(E(k)-E(b))^i$$

4. 사용자는 서버로부터 받은 $E(c^{-1})$을 복호화하고 전처리 상수 k^{n+1}을 곱한다.
Output : $\frac{a}{b}$

3.2.3 동형암호에서의 제곱근 연산

실수 단위의 동형암호에서의 제곱근 연산을 구하기 위하여 뉴턴 방법과 3.2.2절에서 제시한 동형암호에서의 역수 연산을 이용해야 한다. 뉴턴 방법은 방정식의 근사해를 찾을 때 유용하게 사용되는 방법이다. b의 제곱근은 방정식 $f(x) = x^2 - b$의 해가 되며 초깃값 x_0에 대해 다음 수식에 대해 x값이 수렴할 때까지 반복한다.

$$x_{t+1} = x_t - \frac{f(x_t)}{f'(x_t)} = \frac{1}{2}(x_t + \frac{b}{x_t})$$

3.2.2절을 통해 임의의 실수에 대해 동형암호 연산을 통해 역수값을 찾을 수 있다. 이를 알고리즘으로 표현하면 다음과 같다.

Input : 피연산자 a, 반복 횟수 n, 초깃값 b_0
Algorithm :
1. 사용자는 $E(a), E(b_0)$과 반복 시행횟수 n을 채택하여 $E(0.5)$와 같이 서버로 전송한다.
2. 서버는 동형암호 덧셈, 곱셈, 나눗셈 연산을

이용하여 다음의 연산을 수행한다.

$$E(b_1) = E(0.5) \times (E(b_0) + E(a) \times E(b_0^{-1}))$$

3. 서버는 사용자가 같이 전송한 반복 시행횟수만큼 과정 2를 반복하여 최종적으로 $E(b_n)$을 계산한다.

4. 서버는 사용자에게 $E(b_n)$을 전송한다

5. 사용자는 전달받은 $E(b_n)$을 복호화한다.

Output : $b_n(\sqrt{a}$의 근사치)

하지만, 뉴턴 방법은 초깃값을 설정해야 한다는 단점이 있고 초깃값과 시행횟수에 따라 참값과의 오차가 매우 커질 수도 있다. Bourse와 Sanders에 따르면 정수에 한해서 동형암호화된 암호문끼리 비교 연산이 가능하다[5]. 본 논문에서는 뉴턴 방법을 사용하지 않고 동형암호문끼리의 비교 연산을 사용하여 제곱근을 다음과 같이 연산한다.

Input : 피연산자 x, 제곱근의 유효 숫자 개수 n, 전처리 상수 k, D'

Algorithm :

1. 사용자는 피연산자 x에 $10^{(2k+2n)}$꼴의 적절한 수를 곱하여 피연산자를 정수형으로 변환한다.

2. 사용자는 $E(X = x \times 10^{2k+2n})$, $E(1)$, 제곱근의 유효 숫자 개수 n, X의 자릿수 D를 통해 $D' = \left\lceil \frac{D-1}{2} \right\rceil$을 계산하여 서버로 전송한다.

3. 서버는 $a, b = E(0)$을 생성하고 다음의 과정을 유효 숫자 개수인 n번 반복한다. 즉, 다음의 과정을 1번 시행할 때 유효 숫자가 1개씩 증가하게 된다.

3-1. 서버는 동형암호의 비교, 곱셈 연산을 통해 $E(X)$를 넘지 않는 최댓값 $E(r) = (a + E(c)) \times E(c) \times E(10)^{2D'}$을 구한다. 단, 여기서 c는 한 자릿수이다.

3-2. $a = (a + E(2) \times E(c)) \times E(10)$

3-3. $b = b \times E(10) + E(c)$

3-4. $E(X) = (E(X) - E(r))$

3-5. $D' = D' - 1$

4. D'가 0이 될 때까지 1씩 감소시키면서 $b = b \times E(10)$을 계산하고 최종적으로 b를 사용자에게 전달한다.

5. 사용자는 서버로부터 전달받은 b의 값을 복호화하고 $10^{(k+n)}$으로 나눈다.

Output : 유효 숫자가 n개인 $\sqrt{x}$

4. 결론

최근 주목받고 있는 4세대 암호인 동형암호를 통해 데이터를 암호화된 상태에서 연산할 수 있게 되었다. 이를 이용하여 검색, 통계 처리 및 기계학습이 가능해졌고 데이터를 처리하는 중간 과정에서 복호화하지 않아도 되므로 데이터 유출 위험이 감소하는 장점이 있다. 하지만, 동형암호는 확장률, 다양한 연산 불가, 암복호화 속도가 느리다는 단점이 존재한다.

본 논문에서는 동형암호를 이용하여 대용량 데이터를 기반으로 통계 연산을 진행할 때, 통계 연산의 관점에서 병렬처리 방법을 제안하였다. 또한, 통계 연산에서 사용되는 뺄셈, 나눗셈, 제곱근의 연산을 동형암호에서 사용 가능한 덧셈, 곱셈만을 통하여 가능하게 하였다. 그러나, 현재 동형암호는 암호문끼리의 연산이 무한 번 가능하지 않다. 동형암호문끼리 덧셈이나 곱셈 연산을 하게 되면 노이즈가 점점 쌓이게 되고, 결국 어느 한계점에 도달하게 되면 암호문의 복호화가 불가능해진다. 이는 기존 동형암호 원천기술에 관한 내용으로써 동형암호 자체에 대한 성능 및 효율성에 관한 개선연구가 필요할 것이다.

ACKNOWLEDGEMENT

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임. (No.2021-0-00558, 동형암호 기술을 활용한 국가통계 분석 시스템 개발)

참고문헌

[1] Lu W, Kawasaki S, Sakuma J. “Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data”, IACTR Crypto, 2016

[2] 유준수, 윤지원. "LWE와 완전동형암호에 대한 분석 및 동향." 정보보호학회지 30.5 2020: 111-119.

[3] Brown S “Measures of shape: Skewness and kurtosis”, 2011

[4] Ducas L, Micciancio D. “FHEW: bootstrapping homomorphic encryption in less than a second.” In Annual International Conference on the Theory and Applications of Cryptographic Techniques, Berlin, Heidelberg, 2015: 617-640

[5] Bourse F, Sanders O, Traoré J. “Improved secure integer comparison via homomorphic encryption.” In Cryptographers’ Track at the RSA Conference, San Francisco, 2020: 391-416