결정트리 기반의 기계학습을 이용한 익명화기법 연구

김영기O, 홍충선*
경희대학교 컴퓨터공학과
{qoo0144, cshong}@khu.ac.kr

A Study on Anonymization Technique Using Decision Tree Based Machine Learning

Youngki KimO, ChoongSeon Hong*
Department of Computer Science and Engineering, Kyung Hee University

요 약

사물인터넷, 클라우드 컴퓨팅, 빅데이터 등 새로운 기술의 도입으로 처리하는 데이터의 종류와 양이 증가하면서, 개인의 민감한 정보가 유출되는 것에 대한 보안이슈가 더욱 중요시되고 있다. 민감정보를 보호하기 위한 방법으로 데이터에 포함된 개인정보를 공개 또는 배포하기 전에 일부를 삭제하거나 알아볼 수 없는 형태로 변환하는 익명화기법을 사용한다. 그러나 준식별자의 일반화 수준을 계층화하여 익명화를 수행하는 기존의 방법은 데이터 테이블의 레코드가 추가 또는 삭제되어 K-익명성을 만족하지 못하는 경우에 더 높은 일반화 수준으로 데이터를 익명화해야 한다. 이와 같은 과정으로 인한 정보의 손실이 불가피하며 이는 데이터의 유용성을 저해하는 요소이다. 따라서 본 논문에서는 결정트리 기반의 기계학습을 적용하여 기존의 익명화방법의 정보손실을 최소화하여 데이터의 유용성을 향상시키는 익명화기법을 제안한다.

1. 서 론

최근에는 사물인터넷, 빅데이터 등 새로운 기술의 도입으로 처리하는 데이터의 종류와 양이 점차 증가하면서, 개인의 민감한 정보가 유출되는 것에 대한 보안이슈가 더욱 중요시되고 있다. 따라서 개인의 민감한 정보를 보호하기 위한 방법으로 데이터에 포함된 개인정보를 공개하거나 배포하기 전에 일부를 삭제하거나 알아볼 수 없는 형태로 변환하는 익명화기법을 사용한다. 익명화 기법의 종류에는 가명, 일반화, 치환, 섭동 등이 있으며, 이를 통해 민감한 정보가 포함된 개인정보를 보호할 수 있다.

한편, 기존의 익명화 방법에서는 데이터의 준식별자에 해당되는 값을 알아볼 수 없는 형태로 변환하는 일반화의 수준(level of generalization)을 계층화(hierarchy)하는 방법으로 익명화를 수행한다. 그러나 기존의 익명화 방법에서 사용자가 요구하는 K-익명성을 만족하지 않도록 레코드가 추가/삭제된 경우 더 높은 준식별자의 수준으로 익명화를 수행해야 한다. 이 과정에서 불필요한 정보의 손실을 피할 수 없으며 이는 데이터의 유용성을 저해하는 요소이다. 따라서 본 논문에서는 익명화 기법에 결정트리기반의 기계학습을 적용하여 정보의 손실을 최소화하여 익명화를 수행할 수 있는 방안을 제안한다.

2장에서는 데이터 테이블의 구조와 익명화, K-익명성, 기계학습의 관련 연구를 분석하고, 3장에서는 기존 익명화 방법의 문제점 및 제안하는 익명화 기법에 대해 기술한다. 4장에서는 실험 결과 및 성능평가를 분석하고, 마지막으로 5장에서는 본 논문의 결론에 대하여 언급한다.

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (R0126-16-1009, ICBMS 플랫폼 간 정보 모델 연동 및 서비스 매쉬업을 위한 스마트 중재 기술 개발) *Dr. CS Hong is the corresponding author

2. 관련 연구

2.1 데이터 익명화

Identifier Quasi-Identifier Sensitive Attribute

이름	나이	성별	우편번호	질병
Amy	27	F	482010	AIDS
Bobby	22	M	482750	cancer
Carol	37	F	420760	cold
David	39	F	420880	diabetes

그림 1. 데이터 테이블의 예

일반적으로 수집된 데이터 테이블은 그림 1[1]과 같이 식별자(identifier), 준식별자(quasi-identifier), 민감정보(sensitive attribute)로 구성된다. 이름이나 주민등록번호와 같이 특정 데이터를 다른 데이터와 구분할 수 있는 속성을 식별자라 하며 나이, 성별 등 직접적으로 대상을 알 수는 없지만 해당 데이터의 특징을 나타내는 속성을 준식별자라 한다. 또한 급여, 질병과 같이 개인의 민감한 속성을 포함한 데이터를 민감정보라 한다.

표 1. 데이터 익명화의 예

나이	성별	우편번호	질병
20-29	*	482***	AIDS
20-29	*	482***	cancer
30-39	*	420760	cold
30-39	*	420880	diabetes

표 1은 익명화된 데이터 테이블의 일례이다. 익명화는 일반적으로 데이터 테이블에서 식별자를 삭제하고 준식별자에 해당되는 데이터를 치환함으로써 프라이버시 보호를 수행한다.

2.2 K-익명성 (K-anonymity)

K-익명성은 수집된 데이터 테이블에 익명화를 수행하여 적어도 k개 이상의 준식별자 속성값들이 동일한 값을 갖도록 하는 것이다[2]. 표 3은 k=2를 만족하도록 데이터 테이블을 익명화한 결과이다.

표 2. K-익명성의 예 (K=2)

나이	성별	우편번호	질병
20-29	*	482***	AIDS
20-29	*	482***	cancer
30-39	F	420880	cold
30-39	F	420880	cold

2.3 기계학습 (Machine Learning)

기계학습은 사전에 정의된 입출력을 매핑(mapping)하는 함수를 학습하는 지도학습(supervised learning)과 목표값 없이 입력에 근거하여 학습을 진행하는 비지도학습(unsupervised learning)으로 분류된다. 본 논문에서는 개인정보를 보호하기 위한 익명화 기법으로 준식별자 값의 레코드가 정보의 손실을 최소화하는 단계로 일반화하는 것을 기준으로 결정트리를 구성하여 기계학습에 적용한다. 그림 2는 표 2의 데이터 테이블을 활용하여 기계학습에 필요한 결정트리를 구성하는 예이다.

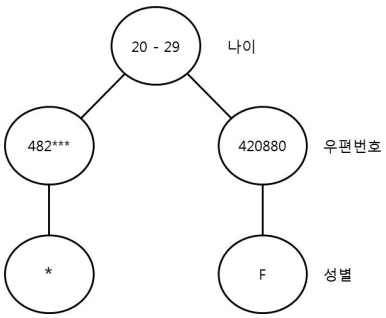

그림 2. 익명화된 테이블로 구성한 결정트리

각 노드는 익명화된 준식별자의 레코드값을 바탕으로 구성되며 같은 트리레벨에는 하나의 준식별자에 대한 데이터들이 나열된다. 이와 같이 구성된 결정트리의 리프노드(leaf node)의 수는 익명화된 데이터 테이블의 동질집합의 수와 같다.

3. 기존 연구의 문제점 및 제안사항

2 장에서 살펴본 바와 같이, 개인의 민감한 정보를 보호하기 위해 k-익명성과 같은 변수를 익명화과정에 추가하여 데이터 테이블이 해당 조건을 만족하도록 익명화를 수행한다. 그러나 준식별자의 일반화 수준을 계층화하여 익명화를 수행하는 기존의 방법은 데이터 테이블의 레코드가 추가 또는 삭제되어 K-익명성을 만족하지 못하는 경우에 더 높은 일반화 수준으로 데이터를 익명화해야 한다. 이와 같은 과정으로 인한 정보의 손실이 불가피하며 이는 데이터의 유용성을 저해하는 요소이다[3]. 그림 3과 그림 4는 데이터의 추가/삭제에 따라 발생되는 정보손실의 예이다.

나이	성별	우편번호	질병
20-29	*	482***	AIDS
20-29	*	482***	cancer
20-29	F	420880	cold
20-29	F	420880	cold
23	F	418000	diabete

나이	성별	우편번호	질병
20-29	*	482***	AIDS
20-29	*	482***	cancer
20-29	F	*	cold
20-29	F	*	cold
20-29	F	*	diabete

그림 3. 데이터 추가에 따른 정보손실 (k=2)

나이	성별	우편번호	질병
20-29	*	482***	AIDS
20-29	*	482***	cancer
20-29	F	420880	cold
20-29	F	420880	cold

나이	성별	우편번호	질병
20-29	*	*	cancer
20-29	*	*	cold
20-29	*	*	cold

그림 4. 데이터 삭제에 따른 정보손실 (k=2)

그림 3과같이 데이터 테이블에 추가된 레코드를 k=2를 만족하도록 익명화하기 위해 두 번째 동질집합의 레코드들이 더 높은 수준의 일반화가 적용되었다. 이 과정에서 부수적으로 우편번호 데이터가 손실되었다. 마찬가지로 그림 4에서는 데이터 테이블의 레코드가 삭제되면서 수행되는 재익명화 과정에서 다른 레코드들의 정보가 손실되었다.

위와 같은 문제점들을 보완하기 위해 데이터를 익명화하는 과정에 결정트리기반의 기계학습을 적용한다. 익명화된 데이터를 학습 집단(training set)으로 결정트리를 구성하고 이를 통해 사용자가 요구하는 k-익명성을 만족시키지 못하는 경우 일반화의 수준을 기준으로 정보의 손실을 최소로 하도록 한다. 제안사항은 최적의 성능을 보장하기 위해 다음 두 가지의 제약사항을 만족해야 한다.

조건 1. 결정트리를 구성할 때는 일반화의 계층이 많게 존재하는 준식별자를 우선적으로 분류한다.

조건 2. 데이터의 추가, 삭제로 k-익명성을 만족할 수 없는 경우 넓이우선탐색을 수행하여 정보손실이 최소가 되는 노드를 찾는다.

정리 1과 정리 2를 이용하여 구성된 결정트리를 바탕으로 재익명화가 수행된다고 가정했을 때, 루트노드에 위치할수록 많은 정보손실이 발생한 준식별자 값을 고려할 수 있다. 또한 변경된 데이터가 결정트리의 어느 노드에 위치하는지 넓이우선탐색을 수행하여 정보손실을 최소화하는 노드를 찾는다.

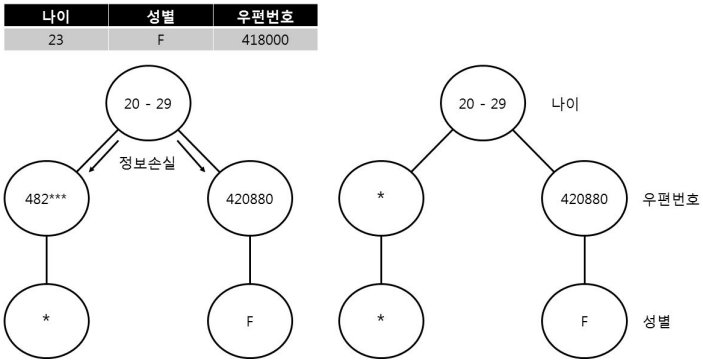

그림 5. 결정트리를 이용한 재익명화

그림 5는 결정트리를 이용하여 재익명화가 수행되는 과정을 데이터 익명화가 수행되는 과정을 나타낸다. 해당되는 준식별자 값이 조건을 만족하면 다음 자식노드들을 확인하여 정보손실이 최소화되도록 트리노드와 데이터 테이블을 변경하여 익명화를 수행한다.

4. 성능 평가

표 3. 실험 환경

Table Size	100 × 6
Total Record	100
Num. of Added Records	10 - 50
Num. of Deleted Records	10 - 50
Total Attribute	6
Num. of Identifier	1
Num. of Q-Identifier	4
Num. of Sentitive Attribute	1
k-anonymity	k = 2

본 논문에서는 연구를 수행하기 위한 익명화 도구로 ARX[4] 라이브러리를 사용하였으며, Java-ML[5] 라이브러리를 활용하여 결정트리를 구성하고 익명화모듈에 기계학습을 적용한다.

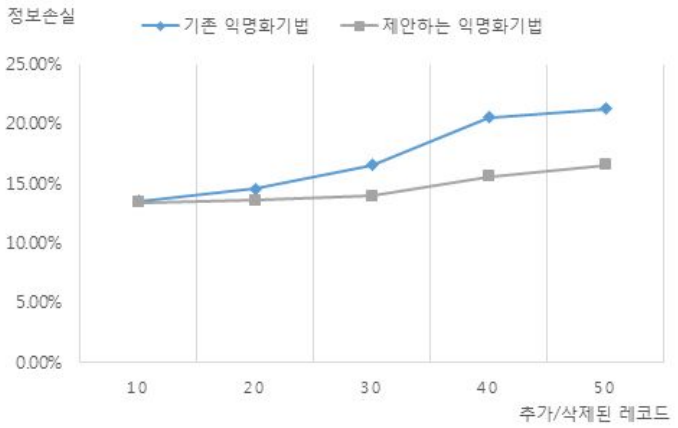

그림 6. 데이터 추가/삭제에 따른 정보손실

그림 6은 k=2로 익명화한 데이터 테이블에 표 3에서 언급한 시나리오를 바탕으로 결과를 분석한 그림이다. 평가기준은 준식별자가 최대로 일반화된 것을 최대치로 하며 일반화 단계의 수만큼 가중치를 달리하여 모든 준식별자의 정보손실을 계산하고 원본 데이터와 비교한다. 그림 3에서 확인할 수 있듯이 입력 데이터의 수가 많아질수록 정보손실의 차이가 늘어나는 것을 확인할 수 있다.

5. 결론

본 논문은 데이터 테이블의 레코드가 추가 또는 삭제되어 K-익명성을 만족하지 못하는 경우에 더 높은 일반화 수준으로 데이터를 익명화하는 기존 방법의 문제점을 해결하기 위한 방법으로 결정트리기반의 기계학습을 적용했다. 4 장에서 성능평가를 통해 추가/삭제되는 데이터의 양이 증가함에 따른 정보손실을 분석하여 제안하는 익명화 기법의 성능을 검증했다. 본 논문에서 제시하는 익명화기법은 정보의 손실측면에서는 기존 방법보다 우수하지만, 추가적인 연산을 필요로 하므로 실행시간이 더 오래 걸린다. 따라서 향후계획으로 연산과정을 최적화하여 익명화에 필요한 실행시간을 개선하는 연구가 진행될 것으로 예상한다.

참 고 문 헌

[1] 황치광, 홍충선, “프라이버시 보호와 데이터 유용성 향상을 위한 서비스 기반의 안전한 익명화 기법,“ 한국정보과학회 제 41회 동계학술발표회(KIISE 2014), 2014.12.18~20(18)

[2] L. Sweeny, “k-anonymity: a model for protecting privacy,”, International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, Vol. 10, No. 5, pp.557-570, 2012.

[3] Li, Tiancheng, and Ninghui Li. “On the tradeoff between privacy and utility in data publishing,”, The 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.517-526, 2009.

[4] ARX, [Online], Available: http://arx.deidentifier.org/

[5] Java-ML, [Online], Available: http://java-ml.sourceforge.net/