Ⅱ 데이터 노출 제어 기법 현황

이 장에서는 현재 많이 사용되는 데이터 가명·익명처리 방법인 마스킹(Masking)과 차등적 정보보호(Differential Privacy)에 대해 살펴보고 각각의 장단점에 대해 논의하겠다.

1. 마스킹(Masking) 기법

마스킹은 원자료의 적절한 변환을 통해 민감한 정보를 가리는 기법을 말하며, 마스킹된 자료란 변환된 자료와 변환에 관한 모든 정보를 의미한다. 이용자가 올바른 분석을 진행하기 위해서는 변환에 관한 정보를 보유하여야 하므로 함께 제공되어야 한다. 이해가 용이하다는 장점이 있지만, 마스킹 기법을 과도하게 사용하면 정보손실이 커지기 때문에 유의하여야 한다. 데이터마다 다른 특성을 가지고 있기 때문에 상황에 따라 적용되는 마스킹 방법 역시 달라져야 하는데, 보통 마스킹은 크게 변조(Perturbation), 비 변조(Non Perturbation) 방법론으로 나뉜다.

변조(Perturbation)는 데이터값을 억제하지 않고 실제 값 주변에 불확실성을 만들어 실제 원데이터값을 변경하는 방식인 반면, 비 변조(Non Perturbation)는 데이터 구조 자체를 왜곡하지 않고 특정 값을 대체하거나 억제해서 데이터의 세부사항을 가리는 방식이다. 변조와 비 변조 방법 모두 명목형 변수와 연속형 변수에 사용할 수 있다.

마스킹 기법은 확률론적 방법(Probabilistic method)과 결정론적 방법(Deterministic method)으로 나누어 생각할 수도 있다. 확률론적 방법은 확률 메커니즘 또는 무작위 난수 생성 메커니즘에 따라 달라지는데, 사용될 때마다 다른 결과가 생성된다는 특징이 있으며 이러한 방법의 경우 계속 같은 결과를 도출하고 싶으면, 난수 생성기에 시드(set.seed)를 설정해야 한다. 이에 반해 결정론적 방법은 지정된 특정 알고리즘을 정확히 따르기 때문에 동일한 데이터에 반복적으로 적용하더라도 동일한 결과를 도출하게 된다.

가. 변조(Perturbation)

변조는 데이터값을 억제하는 것이 아니라 특정 알고리즘이나 모형을 이용하여 실제 값을 변경하여 노출위험을 낮추는 방법이다. 즉, 원데이터의 값들이 변경되었기 때문에 준식별자 값의 빈도 수에 기초한 위험 측정은 변조를 적용한 후에는 더 이상 유효하지 않다.

〈표 Ⅱ-1〉 데이터 변조 예시

ID	원데이터			변조 이후 데이터		
	성별	지역	교육 수준	성별	지역	교육 수준
1	female	rural	high	female	rural	high
2	female	rural	high	female	rural	*low**
3	male	rural	low	*male**	rural	low
4	male	rural	low	female	rural	low
5	female	urban	low	*male**	urban	*high**
6	female	urban	low	female	urban	low

주: *는 원데이터에서 변조 이후 값이 변경된 부분을 뜻함
자료: SDC practical guide(https://sdcpractice.readthedocs.io/en/latest/index.html)

〈표 Ⅱ-1〉을 예로 들면, 데이터 변조 전과 후 모두, 모든 관측치는 $k=3$ 수준에서 k-익명성을 위반하지만[5] 그럼에도 불구하고 각 관측치의 정확한 재식별 위험은 감소함을 알 수 있다. 침입자가 샘플에 유일하게 있는 조합인 {'male', 'urban', 'high'}를 원데이터와 맞추려 할 때, 원데이터에는 이러한 특성을 가진 개인이 포함되어 있지 않으므로 일치 관계를 찾아내기가 힘들다. 따라서 침입자는 해당 관측치의 다른 변수값이 올바른지 확신할 수 없게 된다.

변조의 장점은 수치를 없애버리는 것이 아니라 말 그대로 변화를 주는 것이기 때문에 비변조 방식에 비해 정보 손실이 덜하다는 것이다. 단점은 변조에 사용된 알고리즘이나 모형의 성능에 따라 변조된 데이터가 원데이터의 구조를 보존하는 정도가 달라지고, 외형상 원데이터와 흡사해 익명화 처리가 미흡하다는 인상을 줄 수 있다는 점이다.

5) 즉, 각 키가 데이터 세트에 두 번 이상 나타나지 않음. k-익명성에 대해서는 Ⅳ장에서 자세히 설명할 것임

1) PRAM(Post RAndoMization)[6]

PRAM은 범주형 변수에 주로 사용하며 제어하고자 하는 값에 의도적으로 노이즈를 첨가하는 방법이다. 재코딩과 국소적 감추기로는 노출 제어가 충분하지 않고 정보 손실이 큰 경우 이 방법을 사용할 수 있으며 일부 데이터 개체들에 미리 지정한 전이확률 행렬에 따라 다른 값으로 변환하는 기법이다.[7]

구체적인 행렬을 예시로 들어 PRAM을 설명하려 한다. 어떤 데이터셋에 여아를 출산한 산모가 30명, 남아를 출산한 산모가 50명, 사산아를 출산한 산모 2명에 대한 정보가 있다고 하자. 사산아를 낳은 산모의 수가 상대적으로 적기 때문에 추가정보와 결합 시 신원 노출이 일어날 수 있다. 이때 전이행렬 P를 목적에 맞는 수치로 지정해 PRAM을 적용할 수 있다. 전이행렬 P는 범주가 n개일 때 $n \times n$ 크기로 만들어지는 행렬 형태이며 원래 범주를 어느 정도 유지할 것인지 비율을 먼저 정한다. 이 비율은 행렬에서 대각 행렬 부분에 나타나는데 원데이터를 최대한 보존하고 싶다면 그 수치를 1에 가깝게 하면 된다. 데이터 수를 맞춰주기 위해서는 열별 합이 1이 되도록 지정해야 하며 대각행렬 비율 지정 후 나머지 범주에서 다른 범주로 얼마나 변경할지에 대한 비율 수치를 정한다. 이 행렬의 성분으로 들어가는 수치는 분석 목적과 정보보호 수준에 따라 자의적으로 정하게 된다. 즉, 프로그래밍 상에서 PRAM을 적용할 때, 확률생성 메커니즘에 따라 전이행렬이 다르게 생성되어 결과가 달라질 수 있다. 예를 들어 P를 다음과 같이 두자.

$$P = \begin{bmatrix} 1 & 0.05 & 0.17 \\ 0 & 0.8 & 0.03 \\ 0 & 0.15 & 0.8 \end{bmatrix}$$

이 행렬의 원소들은 자의적으로 정한 것이다. 먼저 첫째 열에서는 (1,1)번째 원소의 값을 1로 두고 나머지 (2,1)과 (3,1)의 원소 값을 0으로 지정해 열의 합을 1로 맞췄다. 나머지 대각원소인 (2,2)와 (3,3)은 0.8로 두어 원데이터에서 관측된 비율의 80%가 유지되도록 했다. 이 전이행렬의 전치(Transpose)를 원데이터 응답값 R_x와 내적해서 나온 다음 결과가 마스킹을 적용한 데이터가 된다.

---

6) Gouweleeuw, J. M., Kooiman, P., and De Wolf, P. P.(1998)

7) 김승현(2020)

$$P^T R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0.05 & 0.8 & 0.15 \\ 0.17 & 0.03 & 0.8 \end{bmatrix} \begin{bmatrix} 30 \\ 50 \\ 2 \end{bmatrix} = \begin{bmatrix} 30+0+0 \\ 1.5+40+0.3 \\ 5.1+1.5+1.6 \end{bmatrix} = \begin{bmatrix} 30 \\ 41.8 \\ 8.2 \end{bmatrix} \approx \begin{bmatrix} 30 \\ 42 \\ 8 \end{bmatrix}$$

따라서 이 전이행렬 P를 이용해 데이터를 처리하게 되면 여아를 출산한 산모는 30명, 남아를 출산한 산모는 42명, 사산아를 출산한 산모는 8명으로 바뀌어 기록된다. 사산아 출산 산모가 2명에서 8명으로 늘어 정보 노출의 가능성이 줄어드는 것이다.

PRAM의 단점은 전이행렬의 개별 원소 값에 대한 결정이 자의적이고, {중학교 재학, 63세}와 같은 이상치 조합이 생성될 수 있다는 점이다. 따라서 PRAM을 적용할 때 특정 전이가 불가능하도록 설계가 되어야 한다. 예를 들어, 학교 재학 관측치의 경우 나이는 6세에서 18세 사이여야 하는 조건을 주고 PRAM을 수행해야 한다. 또는 나이별로 Strata를 구성하고 그 Strata 내에서만 PRAM이 이루어질 수 있도록 해야 한다.

2) 순위 스와핑(Rank Swapping)

순위 스와핑은 관측치 간 특정 변수 값을 교환하는 것에 기반을 두고 있는 변조 방법으로써 순서형 및 연속형 변수에 다 적용될 수 있다. 순위 스와핑에서는 대상이 되는 변수의 값들이 먼저 내림차순 또는 오름차순으로 정렬된다. 관측치 간 변수 값을 교환하려 할 때 순서대로 정렬된 데이터셋에서 교환 대상이 되는 행은 그 주변 몇몇 값들로만 바꾸어 변경에 제약 조건을 주는 것이다. 즉, 대상이 되는 변수 값은 원래 값 근처에 있는 동일하거나 유사한 값으로 교환되게 된다. 일반적으로 순위 스와핑은 정보 손실과 데이터 보호 사이 균형을 잘 지키면서 노출위험을 낮추는 좋은 방법이지만 다음과 같은 한계가 존재한다.

첫째, 대상 변수의 관측치 주변에 바꿀 수 있는 이웃 값이 많지 않거나 적은 범주만 있는 경우 또는 결측값이 많은 경우 순위 스와핑은 실행해도 값이 변경되지 않기 때문에 적절하지 않다. 예를 들어 '교육' 변수에 {초등, 중등, 고등, 대학}과 같이 소수의 범주만 있는 경우, 순위 스와핑보다는 다른 노출 제어 방법을 사용하는 것이 낫다.

둘째, 순위 스와핑은 침입자가 특정 변수 최고 또는 최저값이 누구에게 속하는지 아는 경우 적절하지 않다. 이는 값 자체가 변경되는 것이 아니라 원래 값이 모두 공개되기 때문에 순위 스와핑 이후 침입자는 이 변수 수준을 알 수 있게 되기 때문이다.

마지막으로 여러 변수에 동시에 순위 스와핑을 적용하게 되면 이후 변수 간 상관 구조

(Correlation structure)가 적절하게 유지되는지 확인해야 하는 작업이 필요하다.

3) 셔플링(Shuffling)

셔플링은 자료를 섞는 방식으로 스와핑과 유사하지만, 차이점은 변수에 대해 적절한 지도학습모형 혹은 분포추정기법을 사용하여 모형에서 추정된 값들의 순위에 따라 원변수를 다른 순서로 치환하는 방식을 취한다는 것이다.

셔플링은 대상이 되는 민감변수의 새로운 값을 나머지 변수들의 조건부 분포를 이용하여 생성한다. 예를 들어, 어떤 데이터에 '소득'과 같이 개인의 중요한 정보가 담겨있는 민감변수가 존재한다고 하고 이를 Y라 하자. 이때 나머지 변수인 '나이', '직업', '학력'을 독립변수로 하는 회귀모형을 적합하여 Y의 추정값을 산출할 수 있다. 이제 관측된 소득값들인 y_i를 순서대로 정렬해 $y_{(1)} < y_{(2)} < \cdots < y_{(n)}$로 두고 추정된 소득값들 $\hat{y}_i$에 대해서도 마찬가지로 순위를 모두 고려한다. 셔플링의 첫 단계로서, 가장 작은 추정값 $\hat{y}_{(1)}$이 자료의 i번째 레코드(혹은 i번째 관측값)에서 산출되었다고 하자. 그러면 i번째 원래의 Y값인 y_i를 $y_{(1)}$으로 치환한다. 두 번째 단계로서, $\hat{y}_{(2)}$가 자료의 j번째 경우에서 산출되었다고 하면 j번째 원래의 값인 y_j를 $y_{(2)}$로 치환한다. 이를 모든 순위에 대해 차례로 실행하면 원관측치들이 모두 치환이 되는데, 치환된 값들의 집합은 원래 값들의 집합과 동일하지만, 순서가 다르게 된다. 이런 의미에서 셔플링은 추정값의 순서에 기반해 원래 값을 재정렬하는 방식으로 이해할 수 있다. 따라서 주변분포에 대해서는 원데이터와 동일한 평균, 분산 등의 통계량을 얻을 수 있지만 공분산 구조는 유지되지 않는다는 단점이 있다. 위의 예를 확장하면 민감변수가 여러 개일 때 다변량(Multivariate) 회귀모형을 사용할 수 있으며 변수나 데이터의 종류에 따라 일반화선형모형(GLM) 혹은 다양한 코퓰라(Copula)를 적용한 모수·비모수적 분포추정 모형 역시 이용 가능하다.

4) 잡음추가(Noise Addition)

데이터의 원래 값에서 매우 작은 값을 더하거나 빼서 연속형 변수의 노출위험을 낮추는 방법이다. 평균이 0인 잡음을 추가하게 되면 연속형 변수에서 원데이터와의 정확한 일치를 방지하면서도 데이터 기본 형태는 대략적으로 유지할 수 있어 효과적인 노출 제어를

할 수 있다. 잡음을 추가할 때는 데이터 유형, 데이터의 사용 목적 및 잡음추가 전후 데이터의 분포적 특성(평균, 분산, 공분산, 상관관계 등)을 고려해야 한다. 다음은 잡음추가를 할 때 주의해야 할 사항들이다.

첫째, 제한 관계를 가지는 변수들에 잡음추가를 적용하면 그 구조에 오류가 생길 수 있다. 예를 들어 개인의 수입과 지출 데이터를 생각해보자. 수입과 지출의 세부 항목들은 총수입 또는 총지출에 합산되며, 원데이터에서 총수입 및 총지출은 그 세부 항목들의 합과 같아야 한다. 그러나 개별 세부항목과 총합에 잡음을 같이 추가하면 총계의 결과가 일치하지 않는다.

둘째, 극단값이 있는 데이터의 경우 잡음추가를 해도 극단값의 노출위험을 낮출 수가 없다. 이는 극단값의 경우 잡음추가 이후에도 여전히 극단값으로 감지되기 때문이다. 예를 들면 특정 지역에서 매우 높은 소득을 가지는 관측치는 이 소득값에 잡음을 첨가해도 여전히 해당 지역에서 가장 높은 소득으로 인식되어 재식별이 가능해질 수 있다. 만약 이러한 극단값의 노출 제어를 위해 분산이 큰 잡음을 추가하게 되면 데이터의 분포가 점점 원데이터와 다른 형태로 변해 정보 손실을 야기하게 된다.

셋째, 둘 이상의 변수에 잡음추가를 적용할 경우에는 데이터의 분포적 구조를 보존하기 위해 원데이터의 공분산에 비례하는 오차를 가지는 다변량(Multivariate) 잡음을 이용하는 것이 도움이 된다. 만약 개별 변수에 독립적으로 단변량 잡음을 추가한다면 주변 분포들의 성질은 유지가 되겠지만 변수 간 상관 구조가 왜곡되기 때문이다.

마지막으로, 변수가 특정 범위에만 존재할 때 이렇게 잡음을 더하는 방식은 적절하지 않을 수도 있어 알맞은 변형이 필요하다. 예를 들어, 양의 변수에 잡음을 더하면 음수값이 생성될 수 있는데 이 경우 음수로 생성된 모든 관측치를 0으로 대체할 수 있다. 그러나 특정 임계값 미만의 값이 잘리는 형태가 되기 때문에 데이터의 분포가 왜곡된다. 이 경우 대안으로서 변수에 기댓값 1과 적절한 양의 분산을 갖는 잡음을 곱하는 방식을 이용할 수 있다.

5) 국소통합(Microaggregation)[8]

국소통합은 일반적으로 연속형 변수에 적용되는 변조 기법이다. 그러나 군집을 형성할 수 있고 군집값에 대한 집계값을 계산할 수 있는 경우에는 범주형 데이터에도 국소통합을 적용할 수 있다.

국소통합은 연속형 혹은 범주형 변수를 적절히 그룹화한 후 그룹별로 관측치의 총계 값을 구해 원데이터값을 대체하는 방법이다. 총계 값은 보통 각 그룹의 평균을 사용하고 회귀분석을 이용해 통계량을 구하는 등 다양한 방법으로 구할 수 있다. 범주형 변수에 국소통합을 사용하는 경우, 최빈값을 사용하는 것이 일반적이다. 또는 연속형 변수에 이상치가 있는 경우에도 중앙값 대체가 더 적절하다.

나. 비 변조(Non-Perturbation)

1) 재코딩(Recoding)

재코딩은 변수에 대한 범주 수 또는 연속형 변수 수를 줄이는 데 사용되는 결정론적 방법이다. 이 작업은 범주형 변수 범주들을 결합 또는 그룹화하거나 연속형 변수에 대한 구간을 생성하는 방식으로 수행된다. 재코딩은 특정 변수의 모든 관측치에 적용되며 노출위험이 있는 관측치에만 적용되는 것이 아니다. 재코딩은 글로벌 재코딩(Global Recoding)과 상·하위 코딩(Top and bottom Coding)으로 나눌 수 있다.

가) 글로벌 재코딩(Global Recoding)

글로벌 재코딩은 범주형 변수에서 범주들을 합하거나 연속형 변수에서 관측값들을 구간별로 나누는 방법이다. 이 방법은 희귀한 변수 값을 가진 관측치들의 노출위험을 줄일 수 있지만, 세부적인 정보들이 범주 안에 통합되어 버리기 때문에 정보의 손실을 피할 수 없다.

예를 들어, 〈표 Ⅱ-2〉와 같이 데이터셋에 5개의 지역 값(지역 1~5)이 있다고 가정해보자. 일부 지역에 속하는 관측치는 매우 적기 때문에 여기에 속하는 관측치들은 높은 재식별

8) 박민정(2016)

위험을 갖게 된다. 이 경우 일부 지역을 재코딩하여 노출위험을 낮출 수 있다.

〈표 Ⅱ-2〉 글로벌 재코딩 예시

ID	재코딩 이전 데이터				재코딩 이후 데이터			
	성별	지역	성적	빈도(f_k)	성별	지역	성적	빈도(f_k)
1	female	지역 1	A	1	female	북부	A	3
2	female	지역 2	A	2	female	북부	A	3
3	female	지역 2	A	2	female	북부	A	3
4	female	지역 3	B	2	female	동부	B	2
5	male	지역 3	B	1	male	동부	B	2
6	female	지역 3	B	2	female	동부	B	2
7	male	지역 3	C	2	male	동부	C	2
8	male	지역 4	C	2	male	남부	C	3
9	male	지역 4	C	2	male	남부	C	3
10	male	지역 5	C	1	male	남부	C	3

자료: SDC practical guide(https://sdcpractice.readthedocs.io/en/latest/index.html)

이를테면 다섯 개의 지역을 {'북부', '동부', '남부'} 세 개의 그룹으로 나누어 재코딩하는 식이다. 이렇게 하면 그 변수 범주 수가 5개에서 3개로 줄어들면서 침입자는 각 준식별자에 대해 최소 2 이상의 빈도를 찾을 수 있게 되어 1~3번 관측치, 4번과 6번 관측치, 5번과 7번 관측치, 8~10번 관측치를 구분할 수 없게 된다. 즉, 원데이터에 재코딩을 적용함으로써 관측치 준식별자 조합에 대한 빈도 수(f_k)를 높이고 노출위험을 감소시킬 수 있다.

여기서 주의할 점은 재코딩 이후 나누어지는 범주 수는 임의적이라는 것이다. 몇 가지 범주로 나눌지는 통계적 노출 제어를 행하는 당사자가 적절히 선택해야 하며, 데이터 사용의 목적에 따라 정보 손실을 최소화하도록 주의해야 한다. 예를 들어, 나이 변수(Age variable)에 대해, 분석가가 6~11세와 12~17세의 학교에 다니는 아동·청소년에 대한 지표를 계산함과 동시에 노출위험도 줄이기 위해 연령을 그룹화해야 하는 경우, 이 분석에 적용할 수 있는 연령 간격을 신중하게 만들어야 한다. 적절한 그룹화는 0~5세, 6~11세, 12~17세 등이 될 수 있는 반면, 0~10세, 11~15세, 16~18세 그룹화는 데이터 유용성

(Data Utility)을 해칠 수 있다. 그룹화할 때는 너비가 동일한 구간을 만드는 것이 보통이지만, 필요에 따라 변수 일부만 그룹화하고 나머지는 그냥 두어도 된다. 예를 들어, 20세 이상의 모든 연령에 그룹화를 적용하고, 20세 미만은 원자료 값을 유지하는 식이다.

나) 상·하위 코딩(Top and bottom Coding)

상·하위 코딩은 분포 또는 범주의 위쪽 또는 아래쪽, 즉 극단값들만 재코딩하는 방법이다. 순서형, 범주형, 연속형 변수에 다 적용할 수 있으며 특히 나이와 소득 관련 변수에서 극단값으로 인해 신원 노출이 일어날 수 있을 때 자주 사용된다. 예를 들어, A동에 자산 500억 원 이상의 인물이 1명 있다면, 'A동 거주, 자산 500억 원 이상'의 정보를 입력하면 그 인물이 특정된다. 외부 데이터와 매칭 성공 시 질병과 같은 세부적인 다른 정보들도 노출되어 문제가 발생할 수 있다.

상·하위 코딩은 극단값을 극단적이지 않은 관측값들과 합쳐서 범주형으로 제공한다. 예를 들어 99세라는 극단값은 '90세 이상'의 범주에 넣으면 된다. 일반적으로 범주 내 관측치가 적을수록 노출위험이 높아지므로 분포의 끝에 있는 값을 하나의 범주로 그룹화하면 극단적 관측치의 노출위험이 감소하게 된다.

그룹화할 관측치의 임계치(Threshold)를 결정하려면 변수의 전체 분포를 확인하여 범주 내 관측치 개수가 특정 빈도 아래로 떨어지는 지점을 찾아내야 하는데, 데이터의 사용 목적과 노출위험의 허용 정도를 고려해서 결정한다.

2) 국소적 감추기(Local suppression)

국소적 감추기는 범주형 변수에 사용하는 방법으로 재코딩 이후에도 노출위험이 있을 시 사용한다. 보통 데이터에서 몇몇 변수 조합들은 빈도가 낮게 나타나게 되는데 이는 그러한 조합을 가진 관측치들의 노출위험이 높아짐을 뜻한다. 이때 민감변수 전체를 삭제하는 것이 아니라 일부 노출위험이 높은 변수의 값들(즉, 테이블 형태의 데이터에서는 특정한 행들)을 삭제하는 국소적 감추기 방식을 취한다.

하지만 연속형 변수에서는 고유한 수치 값이 과도하게 많아지므로 국소적 감추기가 적절하지 않을 수도 있다. 이는 범주형 변수에서 범주의 수가 너무 많은 경우도 마찬가지이다.

이럴 때는 앞서 설명한 재코딩으로 적절한 수의 범주로 먼저 나눈 후 국소적 감추기를 수행할 수 있다.

연구에 따라 사용되는 중요 변수의 노출 제어 혹은 억제를 최대한 줄여야 하는 경우가 있다. 예를 들어, 나이가 중요한 변수인 연구에서 나이 변수에 국소적 감추기가 적용되면 중요한 정보의 손실이 매우 커지는데, 이러한 경우 변수별로 중요도를 측정해 최적의 억제 패턴을 지정하는 것이 좋다. 예를 들면, 나이 변수를 배제한 나머지 변수들에 국소적 감추기를 적용하여 노출위험을 줄이는 것이다. 이렇게 되면 국소적 감추기의 적용 횟수 자체는 많아지지만 나이 변수를 가능한 원래의 값으로 보존할 수 있다.

〈표 Ⅱ-3〉 마스킹 종류 및 적용되는 변수 형태

마스킹 방법	분류	적용되는 변수 형태
글로벌 재코딩	비 변조, 결정론적 방법	연속형, 범주형
상·하위 코딩	비 변조, 결정론적 방법	연속형, 범주형
국소적 감추기	비 변조, 결정론적 방법	범주형
PRAM	변조, 확률론적 방법	범주형
국소통합	변조, 확률론적 방법	연속형
노이즈 첨가	변조, 확률론적 방법	연속형
셔플링	변조, 확률론적 방법	연속형
순위 스와핑	변조, 확률론적 방법	연속형

〈표 Ⅱ-3〉은 앞서 언급한 마스킹 방법론을 정리한 것이다. 목적과 변수 형태에 따라 적용되는 마스킹 방법이 달라진다는 것을 명심하고 적절한 방법을 선택해야 한다.

2. 차등적 정보보호(Differential Privacy)

차등적 정보보호(Differential Privacy, 이하, 'DP'라 함)는 알고리즘이라기보다 수학적 개념이다. 데이터베이스 안에 여러 요소들이 있을 때, 침입자가 쿼리(Query)[9]를 통해 특정 요소에 대한 정보를 알아내는 것을 막아낸다는 개념으로 개인정보보호와 연관이 있고, 이를 통해 다양한 방법론이 파생될 수 있어 가명처리 기법의 일환으로 이해할 수 있다.

DP는 공학 분야에서 2006년 처음 제시된 개념[10]으로 어떤 원데이터에 대해 특정 관측값 하나를 제외했을 때의 쿼리 결과가 원데이터에서 얻은 결과와 충분히 유사하다면 정보보호가 되었다고 여기는 것에서 출발한다.

DP를 수학적으로 정의하기 위해 두 개의 데이터 D_1과 D_2를 고려하자. 두 데이터는 동일하고 다만 특정 관측값 하나만 차이가 난다고 가정한다. 즉 둘 중 어느 한 데이터는 이 관측값이 빠져있다. 이제 특정 결과를 얻기 위한 쿼리를 K라고 하면, 이 쿼리를 각 데이터에 적용한 결과인 $K(D_1)$와 $K(D_2)$가 동일한 분포 S에 속할 확률의 비율을 일정 수준보다 작게 하는 것이 DP의 정의이다. 이를 수식으로 표현하면 어떤 양수 ε에 대해

$$\frac{\Pr\left[\mathrm{K}(\mathrm{D}_1) \in \mathrm{S}\right]}{\Pr\left[\mathrm{K}(\mathrm{D}_2) \in \mathrm{S}\right]} \leq \exp(\varepsilon),\ \text{for } all\ S \subseteq \mathrm{range}(\mathrm{K})$$

이며 이 조건을 만족하면 해당 관측값은 ε 수준에서 정보보호가 보장된다고 말한다. 다시 말해 쿼리의 입장에서 두 데이터는 구분이 불가능한 것이다. 매개변수 ε는 노출허용의 범위를 결정하는데 ε이 작으면 두 데이터 분포 차이가 작아 강한 정보보호를 실현할 수 있지만, 유용성이 떨어지는 반면, ε가 크면 두 데이터 분포 차이가 커 정보보호 정도가 작으나 원본과 바뀐 데이터가 비슷해 유용성이 높아지게 된다. 따라서 이 상충관계를 잘 고려해서 쿼리 K에 대한 적절한 ε의 크기를 설정하는 것이 중요하다.

위의 조건을 만족하는 대표적인 기법으로 라플라스 메커니즘인데, 쿼리의 결과에 라플라스 분포에서 추출한 잡음(Noise)를 더하는 방식이다. 라플라스 메커니즘에서 잡음의 크기

9) 쿼리는 데이터베이스에 특정한 정보를 보여달라는 사용자의 요청을 말함. 넓은 의미에서 데이터 분석도 쿼리로 이해할 수 있음

10) Cynthia Dwork et al.(2006)

는 민감도(Sensitivity)와 프라이버시 모수(Privacy) 값과 관련이 있다. 민감도는 쿼리가 해당 데이터에서 얼마나 민감한가를 나타내며 민감도가 클수록 더 큰 잡음을 추가해야 원데이터와 변형된 데이터에 쿼리를 적용했을 때 두 데이터의 구분이 불가하게 만들 수 있다. 프라이버시 모수는 위 식에서 ε으로 개인정보보호 취약 수준을 나타내는 모수로 커질수록 개인정보보호 정도가 약해진다. 최근에는 라플라스 메커니즘 이외에도 가우시안 메커니즘(Gaussian mechanism), 딥러닝을 이용한 메커니즘 등 다양한 방법론들이 연구되고 있다.[11)]

현재 Google, Microsoft, Meta 등 글로벌 기업들은 개발자들이 DP를 활용할 수 있도록 관련 오픈소스를 제공하고 있다. 대표적으로 Google은 자사의 머신러닝 학습 및 개발 프레임워크 TensorFlow의 내부 모듈로서 TensorFlow Privacy를 2019년에 공개해 DP 기법을 통해 사용자 데이터를 보호할 수 있도록 지원하고, 지속적으로 개인정보보호를 위한 실험용 모듈을 공개하고 있다. 특히 이 Privacy 모듈은 멤버십 추론 공격을 방어하기 위해 개인정보 침해 대상이 되는 특정 샘플이 인공지능 학습용 데이터에 포함되어 있는지를 판정하는 분류기를 구축했다는 점에 의의가 있다.

DP 방법론은 마스킹 기법의 한계를 보완하는 획기적인 개념이지만 한계점이 존재한다. 첫째, 주어진 데이터에 반복적으로 적용되는 쿼리에 대해서는 그 효율이 떨어져 실제 사용에 있어 만족할만한 수준의 보호를 보장하기 위해서는 상당한 크기의 잡음을 더해 데이터의 효용을 훼손해야 하는 것으로 알려져 있다. 둘째, DP 메커니즘의 구현이 데이터의 종류 및 특성, 쿼리와 분석의 종류, 그리고 데이터를 보유한 조직의 개인정보보호 요구 사항에 따라 다르게 되어 일관성이 떨어진다는 점이다. 따라서 목적에 따라 적절한 DP를 구현하기 위해서는 상당한 비용과 시간이 필요할 수 있다. 셋째, 주어진 쿼리에 대해서도 그 복잡성에 따라 적절한 매개변수 ε의 값을 다르게 결정해야 하는데 이 또한 어려운 문제이다. 마지막으로, 크기가 작은 데이터에 DP 메커니즘을 적용하기가 어렵다는 점을 들 수 있다.

11) 자세한 내용은 Abadi, Martin, et al.(2016); Ji, Zhanglong, Zachary C. Lipton, and Charles Elkan(2014)를 참고 바람

3. 한계 및 대안의 필요성

이 장에서는 현재 널리 사용되는 가명·익명처리 기법들에 대해 살펴보았다. 마스킹 기법은 몇 가지 분명한 한계점이 존재한다. 우선, 마스킹된 데이터가 완전히 익명화되지는 않는다. 즉 마이크로데이터의 경우 일부 데이터 행에서 여전히 개인이 식별된다는 의미이다. 예를 들어, 마스킹 기법을 사용한 후에도 나이, 성별, 직업, 지역 등의 정보를 조합하면 여전히 개인을 식별할 가능성이 존재한다. 또 다른 한계점으로, 마스킹된 데이터는 원데이터보다 유용성이 떨어질 수 있다는 점을 들 수 있다. 이는 물론 모든 가명·익명처리 기법에 적용되는 단점이겠지만 마스킹의 수준이 높아질수록 원데이터의 중요한 정보들이 훼손되어 어느 시점에서는 더 이상 쿼리나 분석을 통해 의미있는 결과를 얻지 못하게 된다. 현재 국내의 익명데이터로 분류되는 정보들이 대표적인 예이다.

DP의 경우에도 다양한 방향으로의 이론적 확장과 일반화가 진행되고 있지만 위에서 서술한 한계점들로 인해 아직 광범위하게 사용되고 있지는 않다.

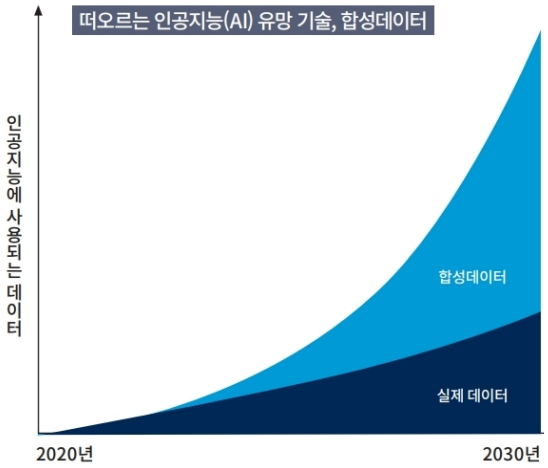
이러한 기존의 방법론들을 대신할 수 있는 가명·익명처리를 위한 기술들에 대해 개인정보보호위원회(2022. 4)의 가명정보 처리 가이드라인은 부록에서 표본 추출, 해부화, 동형비밀분산(혹은 동형암호기법)등을 제시하고 있다. 이 중 데이터의 훼손을 최소화하면서도 높은 수준의 개인정보보호가 가능한 기술은 동형암호기법과 재현데이터 정도이다. 나머지 기술들의 경우 원데이터의 무작위 복원 추출이나 데이터의 분리를 이용함으로 사용의 범위에 있어 제한적이다.

최근 주목을 받고 있는 동형암호기법은 나머지 방법론과 전혀 다른 방식으로 개인정보를 보호한다. 동형암호의 개념은 1970년대 논문으로 소개되었고 이후 2009년 크레이그 젠트리(Craig Gentry)의 연구에 기반해 가명·익명정보처리를 위한 기술로써 사용되고 있다. 동형암호기술은 원데이터를 암호화한 후 그 상태에서 다양한 쿼리와 분석을 진행하는 방식을 취한다. 암호화된 데이터를 이용해 분석하고 결과를 얻는다는 점에서 개인정보보호가 유지되면서도 데이터의 정보는 그대로 보존되는 장점을 가지지만, 아직 분석의 복잡성에 따른 계산량의 문제, 그리고 쿼리의 종류와 범위에 따른 암호화 설계의 어려움이 존재한다고 알려져 있다.

이러한 배경에서 본 보고서는 기존 방법론에 대한 대안으로 재현데이터에 대해 주목하고자 한다. 재현데이터(Synthetic data)[12]는 익명데이터로서 원데이터의 주요 통계적 특성

들을 유지하면서 생성해 낸 모의 데이터이기 때문에 특정 개인을 매칭하는 것이 불가능하고 데이터가 사외로 유출되는 경우에도 큰 충격이 없다. 방법론적으로도 민감변수와 식별변수를 선택적으로 배제할 수 있고 개발된 기법들을 쉽고 적은 비용으로 적용할 수 있다는 점에서 현실적인 대안으로 적합하다.

〈그림 Ⅱ-1〉 재현데이터의 미래



자료: Gartner(https://www.gartner.com/en)

미국의 정보 기술 연구 및 자문 회사 가트너는 2021년 6월 Maverick Research을 인용한 〈그림 Ⅱ-1〉에서 보이듯이 2030년경에는 모형의 학습에 사용되는 데이터 대부분이 합성(재현)데이터가 될 것으로 예측하고 있다. 재현데이터에 대한 자세한 내용은 다음 장에 이어 다루겠다.

12) 인공지능 분야에서는 합성데이터라고 부르며, 비정형데이터를 이용한 딥러닝모형의 학습에 널리 사용되고 있음