Journal of the Korean Data & Information Science Society
2022, **00**(0), 1–24

한국데이터정보과학회지

신용카드 거래 데이터에 대한 재현 데이터 생성 방법 비교 연구[†]

정현우[1], 조윤상[2], 고건우[3], 송재익[4], 유동현[5]

[1,2,3,5]인하대학교 통계학과 [4]나이스지니데이타

접수 0000년 0월 1일, 수정 0000년 0월 0일, 게재확정 0000년 0월 0일

요 약

재현 데이터 생성은 개인정보 보호와 데이터 유용성 확보 측면에서 최근 많은 관심을 받는 통계적 노출 제어의 주요 분야이다. 본 연구에서는 고객의 신용카드 거래 데이터를 기반으로 범주형 변수와 연속형 변수가 혼합된 상황 하에서 최근 재현 데이터 생성에 많이 활용되는 synthpop, 변분 오토인코더, 생성적 적대 신경망 모형을 적용하고 재현 데이터의 노출 위험 및 유용성을 측정하여 비교하였다. 노출 위험 측도로는 외부 공격자 가정에 기반한 목표 속성 식별 확률을 고려하였으며 유용성 지표로는 성향점수 기반 평균제곱오차 및 관심 통계량의 비를 고려하였다. 노출 위험과 유용성 측도의 비교 결과로 synthpop은 노출 위험과 유용성이 가장 높게 나타났으며 변분 오토인코더는 저빈도의 범주와 다수의 범주를 지닌 변수를 지닌 신용 카드 거래 데이터에 대한 재현 성능이 유용성 측면에서 가장 낮게 나타났다. 조건부 벡터 기반의 생성적 적대 신경망 모형의 노출 위험은 synthpop과 비교 시 상대적으로 낮은 위험도와 유사한 유용성을 나타내었다.

주요용어: 변분 오토인코더, 신용카드 거래 데이터, 적대적 생성 신경망, 재현 데이터, synthpop.

1. 서론

최근 4차 산업혁명이라 명명될 정도로 하드웨어 및 소프트웨어의 발전에 따라 다양하고 방대한 양의 데이터가 생성되고 있으며 이에 대한 처리 및 분석 수요가 폭발적으로 증가하고 있다. 예를 들어, 웨어러블 기기로부터 수집 가능한 개인의 건강 정보 데이터, 사물 인터넷에 의해 기록되는 데이터, 온라인 스트리밍 사이트로부터 수집되는 시청 기록 데이터 등, 여러 종류의 데이터가 다양한 매체를 통해 수집되고 있다. 하지만 이러한 사용자와 밀접한 정보들은 개인정보 노출 및 개인 식별의 위험이 있으며 개인정보 보호법에 의거하여 수집된 자료의 활용 및 사업화에는 많은 제약이 존재하였다. 하지만 2020년 데이터 3법이라 지칭되는 개인정보 보호법, 정보통신망 이용촉진 및 정보보호 등에 관한 법률 및 신용정보의 이용 및 보호에 관한 법률이 개정되면서 통계 작성, 연구적 목적, 공익적 기록 보존 등을 위해 신용정보 주체의 동의 없이 가명 처리된 정보를 활용할 수 있게 되었고 가명 처리와 재현 데이터 생성 등 데이터를 활용하는 사업이 활성화 되고 있으며 연구 목적으로 다양한 데이터의 활용이 가능하게 되었다.

[†] **이 성과는 한국지능정보사회진흥원의 지원을 받아 수행된 연구임.**

[1] (22212) 인천광역시 미추홀구 인하로 100, 인하대학교 통계학과, 석사과정
[2] (22212) 인천광역시 미추홀구 인하로 100, 인하대학교 통계학과, 석박사통합과정
[3] (22212) 인천광역시 미추홀구 인하로 100, 인하대학교 통계학과, 석사과정
[4] (07242) 서울특별시 영등포구 은행로 30, 나이스지니데이타 주식회사, 미래혁신본부, 실장
[5] 교신저자: (22212) 인천광역시 미추홀구 인하로 100, 인하대학교 통계학과, 부교수. E-mail: dyu@inha.ac.kr

데이터 3법 개정 이전에는 개인정보에 관한 원 데이터의 정보는 공표할 수 없었기에 데이터의 특성을 수치적으로 요약하는 집계 형식의 매크로데이터 (macrodata) 방식으로 데이터가 공개 및 배포되었지만, 데이터 3법의 개정으로 인하여 가명 정보는 연구 목적으로 제한된 환경 내에서 공개할 수 있게 되었으며 익명 정보는 누구에게나 공표가 가능해졌다. 따라서 데이터를 제공하는 기관 또는 기업들은 데이터를 가명 처리 또는 익명 처리하여 데이터 3법에서 제시된 범위 내에서 이를 제공할 수 있게 되었다. 대부분의 가명 또는 익명 처리 방식은 통계청의 마이크로데이터 (microdata) 공개 시 적용된 통계적 노출 제어(statistical disclosure control, SDC) 기법인 그룹화, 잡음 추가, 식별자의 암호화 등의 매스킹 (masking) 기법을 사용하였다. 보다 상세한 기법의 소개와 이에 대한 적용 사례는 Park과 Kim (2016)에서 확인할 수 있다.

기존의 매스킹 기법은 원 데이터 자체를 활용하는 측면에서 여전히 노출 위험에 대한 우려가 있으며, 노출 위험을 낮추기 위해 원 데이터에 매스킹 기법을 적용함으로써 발생하는 정보 손실의 한계를 지닌다. 이를 극복하기 위하여 원 데이터의 결합 분포를 학습하고 이를 이용하여 결합 분포로부터 표본을 생성하는 재현 데이터 생성 방법 (synthetic data generation method)이 제안되었다. 초기 재현 데이터 생성 방법으로 전체 변수를 순차적으로 나열한 뒤 각각의 변수를 주어진 순서에 따라 조건부 분포를 학습하여 표본을 생성하는 절차가 제안되었으며 조건부 분포 기반 생성 모형으로 회귀 모형이 고려되었다 (Raghunathan, 2001). 회귀 모형에 기반한 순차적 재현 데이터 생성 방법은 회귀 모형의 기본 가정인 회귀 함수의 선형성 가정과 오차항의 분포적 가정의 제약이 있어 이러한 가정이 크게 위배될 경우, 생성된 재현 데이터는 원 데이터의 분포와 큰 차이를 보일 수 있으며 순차적 회귀 모형의 특성으로 각 단계의 불확실성이 누적되어 마지막으로 생성되는 변수는 큰 분산을 지니게 된다. 이러한 회귀 모형 기반의 재현 데이터 생성 방법의 한계를 보완하기 위하여 조건부 분포 학습 및 생성 모형으로 비모수적 모형을 적용하여 재현 데이터를 생성하는 절차가 제안되었다 (Drechsler와 Reiter, 2011). 최근에는 순차적 생성 모형에 기반한 모수적/비모수적 절차를 통합한 R 패키지인 `synthpop` (Nowok 등, 2016)이 개발되어 재현 데이터 생성이 보다 용이해졌다.

또한 딥러닝 기반의 생성 모형들이 이미지 데이터를 성공적으로 생성함에 따라 인공신경망 기반의 재현 데이터 생성 방법들도 개발되고 있으며 크게 변분 오토인코더 (variational autoencoder, VAE)와 생성적 적대 신경망 (generative adversarial network, GAN)에 기반한 방법들로 구분할 수 있다. 일반적으로 딥러닝 기반의 이미지 데이터 생성 모형들은 이미지 데이터를 $[-1, 1]$의 범위 내의 값을 갖는 연속형 값으로 변환하여 연속형 데이터를 생성하는 모형으로 범주형 데이터와 연속형 데이터가 혼합된 데이터에 바로 적용하기에는 한계가 있다. 이러한 한계를 보완하기 위하여 변분 오토인코더 기반의 방법으로는 혼합형 데이터 (mixed type data)에 적용 가능한 VAEM (VAE for heterogeneous mixed type data) 모형 (Ma 등, 2020)이 있으며 GAN 기반의 방법으로는 TableGAN (Park 등, 2018), CTGAN (Xu 등, 2019) 및 CTAB-GAN (Zhao 등, 2021)의 방법이 제안되었다.

일반적으로 재현 데이터의 생성은 데이터의 결합 분포를 추정한 후 추정된 분포를 기반으로 랜덤하게 표본을 생성하게 되므로 원 데이터의 개체 식별 위험은 낮다고 알려져 있다 (Drechsler, 2011). 특히 전체 데이터를 재현하는 완전 재현 데이터 생성에서 원 데이터가 연속형 분포로만 이루어진 경우에는 원 데이터와 동일한 레코드를 생성할 확률은 0이므로 이론적으로는 개체 식별 위험에서 자유롭다. 하지만 일반적으로 관측되는 데이터는 연속형 변수와 범주형 변수를 동시에 포함하는 혼합형 데이터이며 재현 데이터와 원 데이터는 범주형 변수 측면에서는 동일한 레코드가 존재할 수 있으므로 이에 따른 노출 위험을 측정해야 한다는 연구도 최근 발표되었다 (Taub 등, 2020).

따라서 본 논문에서는 Taub와 Elliot (2019)에서 재현 데이터의 노출 위험 측도로 제안된 목표 속성 식별 확률 (targeted corrected attribution probability, TCAP)을 기준으로 최근 개발되어 활용되는 `synthpop`, VAEM, TableGAN, CTGAN, CTAB-GAN 방법들에 대하여 재현 데이터의 노출 위험과

유용성을 비교하고자 한다. 재현 데이터 생성을 위한 원 데이터로는 신용카드의 거래 데이터를 고려하였다. 신용카드의 거래 데이터는 블록화된 지역 코드, 업종 코드, 거래 시간, 성별, 나이, 매출 금액 등을 포함하여 준 식별자가 존재하는 개인 식별 위험을 고려해야 하는 데이터이다. 본 연구에 활용되는 데이터는 나이스지니데이타에서 내부 보안 심의 후 제공하였으며 원 데이터에 가명 처리 및 그룹화 등 여러 비식별 처리한 데이터의 일부를 랜덤 추출한 데이터이다. 따라서 제공받은 데이터는 실제 개인에 대한 식별 위험은 통제하여 제공된 데이터이나 본 연구에서는 재현 데이터 생성 방법의 비교를 위하여 제공 받은 데이터를 실제 원 데이터로 간주하여 재현 데이터의 노출 위험 및 데이터 유용성을 비교하고자 한다.

본 논문의 구성은 다음과 같다. 2절에서는 통계적 모형 기반의 재현 데이터 생성 방법과 딥러닝 기반의 재현 데이터 생성 방법에 대하여 소개하였으며 3절에서는 제공 받은 신용카드 거래 데이터에 대한 변수 현황 및 특성과 재현 데이터 생성을 위한 데이터 전처리에 대하여 설명하였다. 4절에서는 신용카드 거래 데이터에 2절에서 소개한 재현 데이터 생성 방법론을 적용하고 노출 위험 측도와 유용성 측도를 통하여 재현 데이터 생성 방법들을 비교하였다. 마지막으로 5절에서는 본 연구를 전체적으로 요약하고 결론을 제시하였다.

2. 재현데이터 생성 방법론

본 절에서는 통계적 모형과 딥러닝 모형 기반의 재현 데이터 생성 방법들에 대하여 소개한다. 통계적 모형 기반의 방법은 조건부 확률분포의 곱으로 표현된 결합확률분포를 데이터로부터 모수적 또는 비모수적 모형으로 순차적으로 학습한 뒤, 이를 이용하여 데이터를 재현하는 방식으로 본 논문에서는 사용자 이용이 용이한 R 패키지 `synthpop`을 기준으로 통계 모형 기반의 재현 데이터 생성 방법을 소개한다. 딥러닝 기반 모형으로는 잠재 변수의 확률 분포를 변분 베이즈 기반으로 학습하는 VAE 기반 방법과 생성자와 판별자 구조를 기반으로 서로 적대적으로 학습하는 GAN 기반의 방법이 있으며 본 논문에서는 VAEM 방법과 TableGAN, CTGAN, CTAB-GAN을 고려하였다.

2.1. 순차적 조건부 확률분포 학습 기반 생성 방법

통계적인 모형 기반의 재현 데이터 생성 방법은 먼저 원 데이터로부터 데이터의 결합확률분포를 추정한 뒤, 추정된 결합확률분포로부터 표본을 추출하는 방법을 고려한다. 일반적으로 원 데이터의 결합확률분포는 재현 데이터 생성의 용이성과 차원의 저주 (curse of dimensionality) 문제를 고려하여 조건부 확률분포의 곱으로 재표현되며 재표현된 조건부 확률분포를 순차적으로 추정하여 전체 결합확률분포를 추정하게 된다. 각각의 조건부 확률분포를 추정하기 위하여 회귀모형, 로지스틱 회귀모형, 의사결정나무 등 다양한 모수적/비모수적 모형들을 고려할 수 있으며 최근 이러한 순차적 조건부 확률분포 학습 기반의 여러 모형들을 구현하여 제공하는 R 패키지 `synthpop` (Nowok 등, 2016)가 개발되어 통계 기반의 재현 데이터 생성이 보다 용이하게 되었다. 순차적 조건부 확률분포 학습 기반의 생성 방법을 보다 자세히 설명하기 위하여 재현하고자 하는 원 데이터는 $\mathbf{X} = (X_1, X_2, X_3, X_4)$로 4개의 변수로 구성되었다고 가정한다. 이 때, 조건부 확률분포 기반 변수의 재현 순서를 $X_4 \rightarrow X_1 \rightarrow X_3 \rightarrow X_2$를 고려하면 전체 4개의 변수에 대한 결합확률분포 $p(X_1, X_2, X_3, X_4)$는 다음과 같이 표현된다.

$$p(X_1, X_2, X_3, X_4) = p(X_4) \cdot p(X_1|X_4) \cdot p(X_3|X_1, X_4) \cdot p(X_2|X_1, X_3, X_4) \tag{2.1}$$

식 (2.1)의 우측에 표현된 조건부 확률분포 $p(X_4)$, $p(X_1|X_4)$, $p(X_3|X_1, X_4)$, $p(X_2|X_1, X_3, X_4)$는 원 데이터를 기반으로 각각 추정되며, 조건부 확률분포의 목표 확률 변수의 형태에 따라 회귀모형, 다항 로

지스틱 회귀모형 등의 모수적 방법과 의사결정나무 등의 비모수적 방법이 추정 방법으로 고려된다. 본 연구에서 적용하는 R 패키지 `synthpop`은 기본 설정으로 처음 생성되는 변수에 대한 확률 분포를 추정하지 않고 원 데이터의 해당 변수를 랜덤 복원 추출을 적용하여 생성한다. 즉, $p(X_4)$의 분포를 추정하지 않고 원 데이터의 X_4에서 랜덤 복원 추출을 수행한다. 이후의 조건부 확률 분포는 대상 확률 변수의 특징에 따라 추정 방법이 결정된다. 예를 들어, 모수적 방법의 경우에 연속형 변수는 선형 회귀 모형을 적용하여 추정되며 범주형 변수는 범주의 수에 따라 로지스틱 회귀 모형 (범주의 수가 2인 경우)과 다항 로지스틱 회귀모형 (범주의 수가 3 이상인 경우)이 적용된다. `synthpop`에서 비모수적 방법 설정 시 대상 확률 변수의 특징과 관계없이 분류회귀나무 (classification and regression tree, CART) 모형의 적용이 기본 설정으로 되어 있다. 기본 설정인 CART 모형을 적용하여도 보통 우수한 재현 성능을 나타내지만, CART 모형은 의사결정나무 학습 시 표본의 모든 가능한 값에 대해 엔트로피 또는 지니지수를 계산하기 때문에 표본 수가 많은 경우 계산 복잡도가 크게 증가하여 대용량의 데이터 학습은 제한적이다.

학습된 조건부 분포를 기반으로 재현 데이터를 생성 시, `synthpop`은 주어진 순서에 따라 학습한 모형을 기반으로 순차적으로 생성한다. 본 논문의 예로 살펴보면, `synthpop`은 X_4를 랜덤 복원 추출 한 뒤에 추출된 X_4의 값을 기반으로 $\hat{p}(X_1|X_4)$을 이용하여 X_1을 생성한다. 이러한 절차를 $\hat{p}(X_3|X_1, X_4)$와 $\hat{p}(X_2|X_1, X_3, X_4)$에 적용하여 X_3와 X_2를 순차적으로 생성한다. 순차적 조건부 확률분포 학습을 통하여 결합확률분포를 학습한다는 점에서 차원의 저주를 피하고 재현 데이터 생성이 용이하나 데이터의 수와 학습 모형의 복잡도 등에 따라 분포 추정 성능이 크게 영향을 받는다. 따라서 식별자를 지우거나 암호화하는 등의 원 데이터 비식별화 기법과는 다르게 재현 데이터는 원 데이터에서 의미적으로 타당하지 않은 데이터의 생성 가능성도 존재하므로 이에 대해 확인할 필요가 있다. 예를 들어, 나이와 결혼 여부에 대한 변수가 있다고 가정하였을 때, 재현 데이터 생성 기법에 어떠한 제약 조건도 부여하지 않을 경우 (나이, 결혼 여부)를 나타내는 변수에 대하여 (10, 결혼)과 같이 의미적으로 존재할 수 없는 데이터가 생성될 수 있다. `synthpop`의 경우, 재현데이터 생성 시 제약 조건을 부여하여 이러한 문제점을 해결할 수 있으나 원 데이터의 변수가 많은 경우, 이러한 제약 조건의 조합도 크게 증가하여 제약 조건을 부여하는데 한계가 존재한다.

2.2. 변분 오토인코더 기반 방법

변분 오토인코더(VAE) 기반의 재현 데이터 생성 방법을 소개하기에 앞서, Hinton 등 (2006)에 의해 제안된 저차원의 잠재 변수로 원 데이터를 축약하고 복원하는 구조를 갖는 인공신경망 모형인 오토인코더 (autoencoder, AE)를 설명하고자 한다. 오토인코더는 저차원의 잠재 벡터로 원 데이터를 축약하는 인코더 (encoder) 부분과 축약된 잠재 변수로부터 원 데이터를 복원하는 디코더 (decoder) 부분으로 구성되어 있다. 오토인코더는 원 데이터와 축약 후 복원된 데이터 사이의 거리를 손실함수로 사용하며, 모형의 학습 시 인코더는 원 데이터의 특성을 잘 요약하는 잠재 변수를 찾는 방향으로, 디코더는 축약된 잠재 변수로부터 복원된 데이터가 원 데이터와 최대한 가까워지는 방향으로 학습을 진행한다. 오토인코더는 비선형적 차원 축소 방법으로 컴퓨터 비전 분야에서 개발되었으며 사람의 얼굴 이미지로부터 눈, 코, 입의 위치와 같은 특징 추출 (feature extraction) 방법으로 많이 활용되고 있으며 이미지의 잡음 제거에도 활용된다 (Li 등, 2019; Song 등, 2020).

VAE의 소개를 위해 입력 데이터 $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$은 n개의 연속형 또는 범주형 변수의 랜덤 샘플이라 가정하며, $\mathbf{z}$를 잠재 변수, θ를 $\mathbf{x}$와 $\mathbf{z}$의 결합 분포의 모수라고 가정한다. VAE는 전체적인 구조는 오토인코더와 유사하나 오토인코더의 인코더로 축약된 잠재 변수의 확률 분포를 변분 베이즈(variational Bayes) 방법을 기반으로 근사하여 학습하는 점에서 차이가 있다. 보다 구체적으로 살펴 보면, VAE는

인코더를 이용하여 잠재 변수로 재표현하는 과정을 잠재 변수에 대한 사후분포 (posterior distribution)의 확률 밀도 함수인 $p_\theta(\mathbf{z}|\mathbf{x})$로 학습하는 과정으로 인식하고 디코더를 이용하여 원 데이터를 복원하는 과정을 $p_\theta(\mathbf{x}|\mathbf{z})$를 학습하는 과정으로 인식한다. 하지만 사후 분포 $p_\theta(\mathbf{z}|\mathbf{x})$는 원 데이터에 대한 분포 가정과 이에 대응하는 사전 분포 가정이 없는 상황 하에서 추정하기 어렵기 때문에 변분 베이즈에 기반한 근사 사후분포 $q_\phi(\mathbf{z}|\mathbf{x})$를 사용하여 학습한다. 여기서 ϕ는 변분 모수 (variational parameter)로 원 데이터 $\mathbf{x}$로 부터 근사 사후 분포의 모수를 출력하는 인공신경망의 가중치 (weight) 및 편의 (bias)로 정의된다. Figure 2.1 은 위의 과정을 도식화하여 표현한 것이며, 점선과 실선은 각각 인코더와 디코더를 나타낸다.

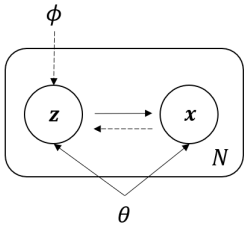


Figure 2.1 Conceptual structure of VAE

VAE는 원 데이터의 분포를 잠재 변수를 통하여 학습하기 위하여 아래의 관계식 (2.2)을 이용한다.

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x})] \\ &= \mathbb{E}_{\mathbf{z}}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) + D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x}))\end{aligned} \tag{2.2}$$

여기서 $p_\theta(\mathbf{x})$는 원 데이터의 확률 밀도 함수를 나타내며 $D_{KL}(P\|Q)$ 은 두 확률분포 P와 Q 사이의 차이를 측정하는 측도인 쿨백-라이블러 발산 (Kullbak-Leibler divergence)을 의미한다. 식 (2.2)의 마지막 항은 쿨백-라이블러 발산의 특성으로 항상 비음 (non-negative)의 값을 갖게 되므로 아래의 관계식 (2.3)이 성립한다.

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})) \tag{2.3}$$

식 (2.3)의 부등호의 우측 항들은 $\log p_\theta(\mathbf{x})$의 하한으로 고려할 수 있으며 VAE 모형에서는 ELBO (evidence lower bound)라 정의하고 인공신경망의 손실 함수로 고려하여 ELBO를 최대화 하는 방향으로 모형의 학습을 진행한다. ELBO의 최대화를 통해 추정한 근사 사후 분포 $q_\phi(\mathbf{z}|\mathbf{x})$와 잠재 변수 기반 원 데이터 복원에 대한 조건부 확률 밀도 함수 $\log p_\theta(\mathbf{x}|\mathbf{z})$를 기반으로 원 데이터의 확률 분포를 근사한다. 일반적으로 연속형 확률 분포에 대하여 잠재 변수의 사전 분포 $p_\lambda(\mathbf{z})$와 근사 사후 분포 $q_\phi(\mathbf{z}|\mathbf{x})$ 모두 정규 분포를 적용하며 사전 분포에는 표준 정규 분포 $N(0, I)$, 근사 사후 분포에는 독립인 정규 분포 $N(\mu, \Sigma)$를 가정하여 학습을 진행한다. 여기서 $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$을 나타낸다.

하지만 이러한 구조를 고려한 VAE 모형의 학습을 위해 ELBO를 최대화하는 과정에서 잠재 변수 $\mathbf{z}$는 확률 과정에 의해 생성되어 미분이 불가능하므로 이를 통해 분포의 모수를 학습할 수 없는 문제가 발생한다. VAE 방법은 이러한 문제를 해결하기 위하여 재모수화 트릭 (reparametrization trick)을 적용하여 문제점를 해결하였다. 재모수화 트릭은 $\mathbf{z}$를 $q_\phi(\mathbf{z}|\mathbf{x})$로부터 추출을 하는 대신, 보조 변수 ϵ을 표준 정규 분포에서 추출하고 이를 변환하여 $\mathbf{z}$가 정의 되도록 하였다. VAE에서는 $\mathbf{z}$가 $q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mu, \Sigma)$의 분포를 따른다고 가정 했을 때, 보조 변수를 이용하여 잠재 변수를 다음과 같이 표현할 수 있다.

$$\mathbf{z} = \mu + \Sigma^{1/2}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{2.4}$$

여기서 $\Sigma^{1/2} = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$을 나타낸다. 재모수화 트릭을 통하여 μ와 Σ를 인공신경망 내의 가중치로 정의가 가능하여 이를 통해 역전파 학습 (back-propagation)이 가능하게 되었다.

기존의 VAE는 원 데이터의 변수들이 동일한 유형이거나 통계적으로 유사한 분포적 성질을 지닐 때 학습이 원활하게 진행된다. 하지만 원 데이터의 변수들이 범주형과 연속형이 혼합되어 동일한 유형의 변수들을 갖지 않는 경우에는 원래 제안된 VAE의 학습 성능이 크게 저하됨이 알려져 있다 (Kendall 등, 2018). 이러한 문제를 해결하기 위하여 VAEM 모형이 Ma 등 (2020)에 의하여 제안되었다. VAEM 모형의 설명을 위하여 $\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_d^T)^T$는 원 데이터의 한 표본으로 정의하고 $\mathbf{z} = (z_1, \ldots, z_d)^T$는 이에 대응하는 잠재 변수 벡터로 정의한다. 여기서 $\mathbf{x}_j$ $(1 \le j \le d)$는 원 데이터의 j번째 변수에 대응하며 연속형 변수의 경우 1차원의 스칼라 (scalar)로 표현되며 범주형 변수의 경우 해당 범주형 변수의 범주 수에 해당하는 0과 1로 구성된 벡터이다. 예를 들어 원 데이터의 X_j 변수가 5개의 범주를 갖는다고 가정하고 $X_j = 3$이면, $\mathbf{x}_j = (0, 0, 1, 0, 0)^T$으로 정의된다. 하지만 잠재 변수 z_j는 1차원의 스칼라로 정의되는 점에서 $\mathbf{x}_j$와 차이가 있다. VAEM 모형은 아래의 두 단계를 통하여 모형을 학습한다.

(**Stage 1**)
첫 번째 단계에서는 각 변수에 대하여 독립적으로 VAE 모형을 학습하며 사전 분포로 $p_{\lambda_j}(z_j) = (2\pi)^{-1/2} e^{-\frac{1}{2} z_j^2}$으로 표준 정규 분포를 고려한다. 즉, 원 데이터의 변수가 연속형이나 범주형 변수를 나타내더라도 각 변수에 대응하는 잠재변수는 1차원의 스칼라로 정의하며 사전 분포와 근사 사후 분포가 정규 분포를 따르도록 한다. 각 변수에 대한 VAE의 모형의 최적화 문제는 다음과 같다.

$$(\theta_j^*, \phi_j^*) = \underset{\theta_j, \phi_j}{\mathrm{argmax}}\, \mathbb{E}_{q_{\phi_j}(z_j|\mathbf{x}_j)}\big[\log p_{\theta_j}(\mathbf{x}_j|z_j)\big] - D_{KL}\big(q_{\phi_j}(z_j|\mathbf{x}_j) \| p_{\lambda_j}(z_j)\big), \tag{2.5}$$

여기서 $q_{\phi_j}(z_j|\mathbf{x}_j)$는 j번째 변수의 잠재 변수에 대한 근사 사후 분포를 나타내는 VAE 모형의 인코더 부분이며 $p_{\theta_j}(\mathbf{x}_j|z_j)$는 잠재 변수가 주어졌을 때 원 데이터의 j번째 변수에 대한 조건부 분포를 나타내는 VAE 모형의 디코더 부분이다. 원 데이터의 변수의 형태에 따라 연속형의 경우 $p_{\theta_j}(\mathbf{x}_j|z_j)$는 연속형 출력값과 평균제곱오차의 손실함수를 갖는 인공신경망 모형을 이용하여 추정하며, 범주형인 경우 해당 범주의 수의 출력값과 소프트맥스 (soft-max) 손실함수를 갖는 인공신경망 모형을 이용하여 학습한다.

(**Stage 2**)
첫 번째 단계에서 각 변수를 독립적인 VAE 모형을 이용하여 학습하며 원 데이터 변수들간의 종속성이 고려되지 않았다. 두 번째 단계에서는 원 데이터의 종속성을 유지할 수 있도록 첫 번째 단계에서 표현된 잠재 변수들을 이용하여 다시 VAE 모형을 적합한다. 즉, VAEM 모형은 이형적 분포를 갖는 원 데이터를 정규분포를 따르는 잠재 변수로 표현한 뒤 원 데이터의 종속성 유지를 위해 잠재 변수들의 잠재 변수 구조를 고려하는 이중 VAE 모형이라 볼 수 있다. 이를 표현하기 위해 $\mathbf{z} = (z_1, \ldots, z_d)^T$는 첫 번째 단계를 통하여 표현된 1단계 잠재 변수로 정의하고 $\mathbf{h} = (h_1, \ldots, h_k)^T$는 두 번째 단계에서 학습하는 1단계 잠재 변수의 잠재 변수 벡터를 나타낸다. 여기서 k $(1 \le k \le d)$는 잠재 변수의 차원을 의미한다. 두 번째 단계의 VAE 모형의 최적화 문제를 표현하면 다음과 같다.

$$\begin{aligned} &\mathbf{x} \sim p_{data}(\mathbf{x}), z_j \sim q_{\phi_j}(z_j|\mathbf{x}_j), j = 1, \ldots, d, \\ &(\psi^*, \eta^*) = \underset{\psi, \eta}{\mathrm{argmax}}\ \mathbb{E}_{q_\eta(\mathbf{h}|\mathbf{x},\mathbf{z})}\big[\log p_\psi(\mathbf{z}|\mathbf{h})\big] - D_{KL}\big(q_\eta(\mathbf{h}|\mathbf{x},\mathbf{z}) \| p_\tau(\mathbf{h})\big), \end{aligned} \tag{2.6}$$

여기서 $p_\tau(\mathbf{h})$는 2단계 잠재변수의 사전 분포를 나타내며 정규 혼합 분포 (Gaussian mixture distribution) 또는 VAMP (variational mixture of posterior) 사전 분포 (Tomczak 와 Welling,

2017)를 고려할 수 있다. VAEM 모형에서는 VAMP 사전 분포를 고려한 모형이 제안되었다.

마지막으로 전체 VAEM 모형을 요약하면 다음과 같이 표현할 수 있다.

$$p_\theta(\mathbf{x}) = \mathbb{E}_{(\mathbf{z},\mathbf{h})\sim p_\tau(\mathbf{h})p_\psi(\mathbf{z}|\mathbf{h})}\big[\prod_j p_{\theta_j}(\mathbf{x}_j|z_j)\big]. \tag{2.7}$$

VAEM 모형을 이용하여 $p_\theta(\mathbf{x})$를 따르는 재현 데이터의 생성은 학습 과정과 동일하게 두 단계를 거쳐 생성된다. 먼저 2단계의 잠재 변수 $\mathbf{h}$를 학습된 $q_{\eta^*}(\mathbf{h}|\mathbf{x},\mathbf{z})$에서 생성한 뒤에 디코더 $p_{\psi^*}(\mathbf{z}|\mathbf{h})$를 이용하여 $\mathbf{z}$로 변환된다. 생성된 $\mathbf{z}$를 이용하여 각각의 변수에 대하여 학습된 디코더 $\log p_{\theta_j^*}(\mathbf{x}_j|z_j)$를 이용하여 $\mathbf{x}_j$를 얻게 된다.

2.3. 생성적 적대 신경망 기반 방법

GAN 모형은 임의의 난수를 추출하여 원 데이터와 유사하도록 생성하는 모형으로 생성자 (generator)와 판별자 (discriminator)로 구성되어 있는 인공신경망 모형이다 (Goodfellow 등, 2014). GAN 모형의 생성자는 임의의 난수를 원 데이터의 형태로 변환하는 역할을 하며 판별자는 원 데이터와 난수로부터 생성된 재현 데이터를 입력받아 원 데이터와 재현 데이터를 구분하는 역할을 한다. GAN 모형은 원 데이터와 유사한 재현 데이터의 생성을 위하여 생성자와 판별자를 경쟁적으로 학습 시키는 모형으로 다음의 목적 함수의 최적화를 통하여 학습이 진행된다.

$$\min_G \max_D V(G,D) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\big[\log D(\mathbf{x})\big] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[\log(1-D(G(\mathbf{z})))\big], \tag{2.8}$$

여기서 $\mathbf{x}$는 원 데이터, $\mathbf{z}$는 보통 균등 분포 (uniform distribution) 또는 정규 분포 (standard normal distribution)에서 생성된 난수를 나타내며, $G(\mathbf{z})$는 생성자를 통하여 변환된 합성 데이터, $D(\mathbf{x})$는 원 데이터를 원 데이터로 판단할 확률, $D(G(\mathbf{z}))$는 합성 데이터를 원 데이터로 판단할 확률을 나타낸다. 따라서 위의 목적함수는 판별자는 원 데이터를 원 데이터로 판단할 확률을 최대화 하도록 학습하고 생성자는 판별자가 합성 데이터를 합성 데이터로 판단하는 확률을 최소화하도록 학습하여 서로 경쟁적으로 학습하게 된다. 이러한 학습이 반복되면서 생성자가 변환하는 합성 데이터가 실제 데이터의 분포와 유사해지며, 이론적인 최적의 상태는 판별자가 원 데이터와 합성 데이터를 구분하지 못하는 상태, 즉 확률이 0.5인 상태가 최적이 된다.

GAN 모형은 최소최대 최적화 문제로 일반적인 최소화, 최대화 최적화 문제보다 해를 구하는 것이 어렵다는 것이 알려져 있으며 완전 연결 계층 (fully connected layer)으로 구성된 GAN 모형은 학습이 불안정하다고 알려져 있다 (Arjovsky 와 Bottou, 2017). 이를 보완하기 위하여 이미지 데이터에 대해 DCGAN (deep convolutional generative adversarial network) 모형이 제안되었다 (Radford 등, 2015). DCGAN 모형은 원래의 GAN 모형에서 고려하였던 완전 연결 계층을 합성곱 신경망 (convolutional neural network, CNN)의 계층으로 대체하고 판별자에 strided 합성곱 계층 적용과 계층의 활성화 함수에 LeakyReLU (Maas 등, 2013) 함수를 적용하여 GAN 모형의 학습을 안정화 시킨 모형이다. DCGAN에서도 활용한 CNN 계층 구조는 이미지 형식의 데이터에 대하여 우수한 성능을 지님이 여러 응용 분야에서 확인되었고 이미지 잡음 제거에도 활용되고 있다 (Kim 등, 2020). DCGAN 모형의 생성자에서는 전치 합성곱 계층 (transposed convolution layer)과 배치 정규화 (batch normalization) 방법 (Ioffe 와 Szegedy, 2015) 및 ReLU 활성화 함수가 적용되었다.

이러한 GAN 기반의 모형이 이미지 생성을 성공적으로 수행함에 따라, 이미지 데이터뿐만 아니라 일반적인 관측 데이터에 대한 생성으로 GAN 모형이 확장되고 있다. TableGAN (Park 등, 2018)은 테이블 형식의 데이터에 대하여 DCGAN 모형을 적용하여 재현 데이터를 생성하는 모형이다. 기존의

DCGAN 모형의 구조를 적용하기 위하여 TableGAN 모형은 테이블의 각각의 행을 정방 행렬의 형태로 변환하여 2차원 이미지 형태의 자료를 고려한다. TableGAN 모형은 테이블의 열의 수가 정방 행렬을 구성하기 어려운 경우 추가적인 패딩 (padding)을 통하여 차원을 일치시키는 전처리가 수행되며 원 데이터의 변수들의 관계를 유지하며 학습하기 위하여 정보 손실 함수 (information loss function)와 분류 손실 함수 (classification loss function)를 도입하여 원 데이터 내의 변수들의 관계와 재현 데이터의 변수들의 관계가 유사하도록 학습을 진행한다. 정보 손실 함수는 판별자의 마지막 계층에서 표현된 벡터를 특징 벡터 (feature vector)로 고려하여 원 데이터의 특징 벡터와 생성된 재현 데이터의 특징 벡터의 평균과 표준편차의 차이로 정의되며 분류 손실 함수는 분류자 (classifier)를 도입하여 원 데이터의 레이블 예측을 학습하고 원 데이터로 부터 학습된 분류자로 생성된 데이터에서의 예측 오차를 손실 함수로 정의한다. TableGAN의 최종 손실함수는 원래의 GAN 모형의 손실 함수인 식 (2.8)에서 정의된 $V(G,D)$와 정보 손실 함수 및 분류 손실함수의 합으로 정의한다. 정보 손실 함수 $\mathcal{L}_{info}^{G}$는 원 데이터와 생성 데이터의 판별자의 마지막 계층의 특징 벡터를 각각 $\mathbf{f_x}$와 $\mathbf{f}_{G(\mathbf{z})}$로 나타낼 때, 다음과 같이 정의된다.

$$\begin{aligned}
\mathcal{L}_{mean} &= ||\,\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\mathbf{f_x}] - \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\mathbf{f}_{G(\mathbf{z})}]\,||_2 \\
\mathcal{L}_{sd} &= ||\,\mathbb{SD}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\mathbf{f_x}] - \mathbb{SD}_{\mathbf{z}\sim p(\mathbf{z})}[\mathbf{f}_{G(\mathbf{z})}]\,||_2 \\
\mathcal{L}_{info}^{G} &= \max(0, \mathcal{L}_{mean} - \delta_{mean}) + \max(0, \mathcal{L}_{sd} - \delta_{sd})
\end{aligned} \tag{2.9}$$

여기서 $\mathbb{SD}[\cdot]$은 주어진 벡터를 이용하여 표준 편차를 계산하는 함수이다. 분류 손실 함수 $\mathcal{L}_{class}$는 원 데이터에 대한 분류 손실 함수 $\mathcal{L}_{class}^{C}$와 생성 데이터에 대한 분류 손실 함수 $\mathcal{L}_{class}^{G}$를 구분하여 아래와 같이 정의한다.

$$\begin{aligned}
\mathcal{L}_{class}^{C} &= \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\,|\,\ell(\mathbf{x}) - C(remove(\mathbf{x}))\,|\,] \\
\mathcal{L}_{class}^{G} &= \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\,|\,\ell(G(\mathbf{z})) - C(remove(G(\mathbf{z})))\,|\,],
\end{aligned} \tag{2.10}$$

여기서 $l(\mathbf{x})$는 원 데이터 $\mathbf{x}$에서 관심인 레이블을 산출하는 함수이며 $remove(\mathbf{x})$는 원 데이터에서 관심인 레이블 정보를 제거한 데이터를 의미한다. 원 데이터에 대한 분류 손실 함수는 분류자를 학습할 때 적용되며 생성 데이터에 대한 분류 손실 함수는 학습된 분류자를 통해 산출되며 생성자의 학습에 적용된다. TableGAN 모형의 전체 개념적 구조를 Figure 2.2에 제시하였다.

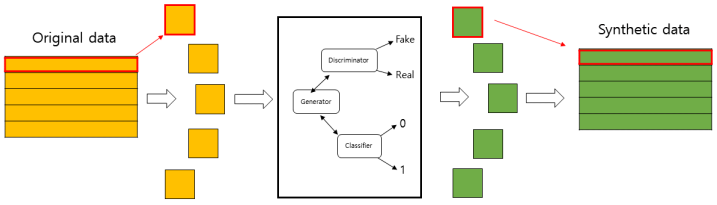


Figure 2.2 Conceptual structure of TableGAN

앞서 살펴본 TableGAN의 경우, DCGAN의 이미지 데이터의 성공적 생성과 실제 테이블 형식의 데이터의 변수들의 관계 유지를 위한 손실 함수 등이 추가적으로 고려되었으나 기존 DCGAN 모형의 구

조를 적용하기 위해 각 변수를 $[-1, 1]$의 범위를 갖도록 변환하게 되면서 기울기 소실 문제가 여전히 발생할 수 있다. 또한 범주형 변수와 연속형 변수가 혼합된 혼합형 데이터에 대하여 범주형 변수의 범주의 불균형이 존재할 경우, 원래의 TableGAN의 학습 구조를 적용하게 되면 재현 데이터 생성 시 소수의 범주가 누락될 가능성이 높다. 이러한 문제를 해결하기 위하여 CTGAN (conditional tabular GAN) 모형이 제안되었다 (Xu 등, 2019). CTGAN 모형은 변수들의 범위 전처리를 통한 기울기 소실 문제의 해결을 위하여 정규 혼합 분포 추정에 기반한 최빈값 기준 정규화 (mode-specific normalization)을 제안하였으며 범주형 변수의 성공적인 생성을 위하여 조건부 벡터 (conditional vector)와 조건부 벡터 기반 표본 추출에 따른 학습 절차를 제안하였다. 먼저 혼합 분포에 기반한 최빈값 기준 정규화 과정을 살펴보면 아래의 절차를 따른다.

(Step 1) 각각의 연속형 변수들에 대하여 변분 정규 혼합 분포 모형 (Bishop, 2006) 을 적용하여 혼합 분포를 추정하여 혼합 분포의 수, 가중치 및 분포 모수를 추정한다. 예를 들어, X_j 변수에 대해 $q_j = 3$개의 분포로 구성된 정규 혼합 분포가 식별되었다고 가정하면 X_j의 정규 혼합 분포는 $P_{X_j}(x_{ij}) = \sum_{k=1}^{3} w_k f(x_{ij}; \hat{\mu_k}, \hat{\sigma}_k^2)$로 표현할 수 있다. 여기서 $f(x; \mu, \sigma^2)$는 평균이 μ이며 분산이 σ^2인 정규 분포의 확률 밀도 함수를 나타낸다.

(Step 2) 주어진 변수의 값에 대하여 Step 1에서 추정된 정규 혼합 분포를 기준으로 각각 분포에 따른 확률 밀도 함수와 가중치의 곱을 산출하여 이 중 가장 높은 값을 갖는 분포를 선택한다. 예를 들어 3개의 분포로 구성된 정규 혼합 분포에서 x_{ij}의 값에 대한 비교 시, $\rho_k = w_k f(x_{ij}; \hat{\mu_k}, \hat{\sigma}_k^2)$를 산출하여 가장 높은 성분을 선택한다. 즉, 선택된 성분은 $m^* = \text{argmax}_k \, \rho_k$로 표현된다.

(Step 3) Step 2에서 선택된 분포 성분을 기준으로 $\alpha_{ij} = (x_{ij} - \hat{\mu}_{m^*})/(4\hat{\sigma}_{m^*})$의 정규화를 수행한다. 또한 각 연속형 변수에 대하여 혼합 분포의 성분에 대한 정보를 활용하기 위하여 β_{ij}를 도입하여 어떤 혼합 분포에 해당하는지 0과 1을 이용하여 표현한다. 예를 들어, 3개의 성분으로 구성된 혼합 분포에서 3번째 성분에 해당하는 ρ_k의 값이 가장 높게 나타났을 경우, $\beta_{ij} = (0, 0, 1)$의 값으로 정의된다.

CTGAN 모형은 연속형 데이터에 최빈값 기준 정규화를 적용하여 정규화된 α와 구성 성분 정보를 갖는 β의 정보로 재표현하고 범주형 변수들은 지시 변수를 활용하여 모두 0과 1의 값을 갖도록 변환한다.

CTGAN 모형은 범주형 데이터에 대한 생성을 위하여 조건부 벡터를 도입하여 생성자를 설계한다. 보다 자세히 살펴 보면, 다른 GAN 기반의 모형과 다르게 CTGAN 모형은 고려한 분포로부터 임의의 난수를 추출한 뒤 정의한 조건부 벡터를 결합하여 생성자의 입력값으로 고려한다. 즉, $\mathbf{z}$를 생성한 난수 벡터라 하고 $\mathbf{c}$를 조건부 벡터라 할 때, 입력 벡터는 $\mathbf{a} = (\mathbf{z}^T, \mathbf{c}^T)^T$로 정의한다. 생성자의 입력에서 범주에 대한 조건부 벡터를 미리 설정하여 학습을 수행하게 되므로 원 데이터와 생성 데이터를 판별하는 판별자의 학습에도 원 데이터를 랜덤하게 추출하여 학습하는 것이 아니라 원 데이터의 분포도 조건부 벡터에 대한 조건부 분포를 학습할 수 있도록 수정되어야 한다. CTGAN 모형에서는 조건부 분포의 학습을 위하여 생성자에 적용되는 조건부 벡터에 따라 원 데이터에서 주어진 조건부 벡터에 대응하는 표본들을 샘플링하여 학습을 진행하게 된다. 예를 들면, 원 데이터에 C_1, C_2의 연속형 변수와 3개의 범주를 갖는 D_1과 2개의 범주를 갖는 D_2의 범주형 변수가 있다고 가정할 때, CTGAN 모형이 $D_2 = 1$에 대한 조건부 분포를 학습하는 과정을 개념적으로 도식화하면 Figure 2.3와 같다. 예시에서는 범주형 변수 D_1과 D_2의 조합에 따른 조건부 분포가 아니라 $D_2 = 1$인 조건부 분포에 대한 학습으로 D_1에 대한 조건을 고려하지 않아 조건부 벡터는 $(0, 0, 0, 1, 0)$으로 정의하였다.

CTGAN 모형은 TableGAN 모형과 다르게 정방 행렬로의 표현이 요구되지 않으며 생성자와 판별자에 완전 연결 계층을 고려하였다. CTGAN의 생성자 $G(\mathbf{z}, \mathbf{c})$는 다음의 구조를 갖는다.

1. $\mathbf{h}_0 = \mathbf{z} \oplus \mathbf{c}$
2. $\mathbf{h}_1 = \mathbf{h}_0 \oplus ReLU(BN(FC_{|\mathbf{z}|+|\mathbf{c}|\to 256}(\mathbf{h}_0)))$
3. $\mathbf{h}_2 = \mathbf{h}_1 \oplus ReLU(BN(FC_{|\mathbf{z}|+|\mathbf{c}|+256\to 256}(\mathbf{h}_1)))$
4. $\alpha_j = \tanh(FC_{|\mathbf{z}|+|\mathbf{c}|+512\to 1}(\mathbf{h}_2)) \quad 1 \le j \le N_c$
5. $\beta_j = \text{gumbel}_{0.2}(FC_{|\mathbf{z}|+|\mathbf{c}|+512\to q_j}(\mathbf{h}_2)) \quad 1 \le j \le N_c$
6. $\mathbf{d}_j = \text{gumbel}_{0.2}(FC_{|\mathbf{z}|+|\mathbf{c}|+512\to |D_j|}(\mathbf{h}_2)) \quad 1 \le j \le N_d$

여기서 N_c는 연속형 변수의 수, N_d는 범주형 변수의 수, $\oplus$는 벡터의 연결 결합을 의미하며 $ReLU(\cdot)$은 ReLU 활성화 함수, BN은 배치 정규화, gumbel_p는 검벨 소프트 맥스 함수 (Gumbel softmax function), $FC_{p\to q}(x)$는 입력 차원 p와 출력 차원 q를 갖는 완전 연결 계층의 출력값을 나타낸다.

CTGAN의 판별자 $C(\cdot)$는 PacGAN (Lin 등, 2018)의 구조를 기반으로 다음과 같이 정의된다. 현재 정의한 구조는 PacGAN의 구조에서 pac의 크기를 10으로 한 경우이다.

1. $\mathbf{h}_0 = \mathbf{r}_1 \oplus \cdots \oplus \mathbf{r}_{10} \oplus \mathbf{c}_1 \oplus \cdots \oplus \mathbf{c}_{10}$
2. $\mathbf{h}_1 = Drop(leaky_{0.2}(FC_{10|\mathbf{r_1}|+10|\mathbf{c}_1|\to 256}(\mathbf{h}_0)))$
3. $\mathbf{h}_2 = Drop(leaky_{0.2}(FC_{256\to 256}(\mathbf{h}_1)))$
4. $C(\cdot) = FC_{256\to 1}(\mathbf{h}_2)$

여기서 $Drop$은 Dropout (Srivastava 등, 2014) 절차를 의미하며 $leaky_p()$는 Leaky ReLU 활성화 함수를 나타낸다.

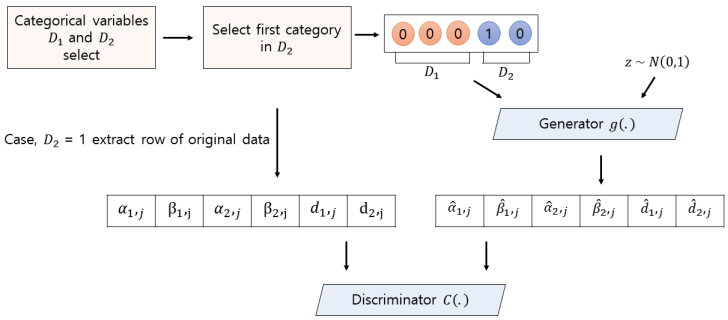


Figure 2.3 Conceptual structure of CTGAN

마지막으로 설명하고자 하는 GAN 기반의 재현 데이터 생성 모형은 CTAB-GAN 모형 (Zhao 등, 2021)으로 앞서 설명한 TableGAN 모형과 CTGAN 모형의 구조를 결합한 형태를 갖는다. 기본적으로는 CTGAN 모형의 구조를 모두 적용하여 최빈값 기준 정규화와 조건부 벡터 및 조건부 벡터 기반 표본 추출에 따른 학습을 적용한다. CTAB-GAN 모형은 CTGAN 모형의 구조를 기반으로 손실함수 부분만 TableGAN 모형에서 고려되었던 정보 손실 함수 및 분류 손실 함수를 추가적으로 반영한 모형이다. CTAB-GAN 논문에서는 TableGAN과 CTGAN 각각의 모형을 적용할 때와 비교하여 재현 성능이 증가함을 여러 데이터를 활용하여 확인하였다. 전체적인 CTAB-GAN의 개념적 구조는 Figure 2.4에 제시하였다.

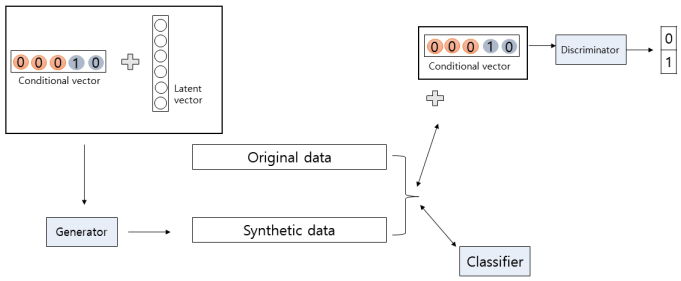


Figure 2.4 Conceptual structure of CTAB-GAN

3. 신용카드 거래 데이터

3.1. 신용카드 거래 데이터의 소개

본 연구에서 사용한 데이터는 나이스 지니데이타에서 가명 처리 및 비식별 처리 후 제공된 데이터로 본 연구의 목적으로만 활용 가능한 데이터이다. 나이스 지니데이타에서 보유한 원 데이터는 2019년 신용카드 고객 매출 데이터로 약 5억 3,000만건이 있으며 이 중에서 서울특별시의 1월 고객 매출 데이터는 약 851만 건이다. 본 연구에 활용되는 데이터는 서울특별시의 1월 고객 매출 데이터 중 5만 건을 랜덤하게 추출하여 제공된 데이터이다. 원 데이터와 비교하여 제공받은 데이터의 수는 상대적으로 매우 작다고 볼 수 있으나 신용카드 거래 데이터는 고객의 나이, 성별, 매출 업종, 매출 가맹점 위치 기반 블록 코드, 결제 시간, 집 우편번호, 결제 금액으로 구성되어 개인 식별이 어느 정도 가능한 준 식별자들이 포함된 데이터이므로 원 데이터의 노출 위험 및 개인정보보호법에 대한 법적 검토와 보안 심의 후 5만 건으로 연구를 진행하기로 논의되었다. 최종적으로 제공 받은 신용카드 거래 데이터의 변수 현황 및 특성은 Table **3.1**에 요약하였다.

Table 3.1 Summary of variables in credit card transaction data

No.	Item	Variable	Type	#(Null)	#(Category)	Range
1	Store type code	TYPE	Categorical	0	158	(1, 385)
2	Block code (location)	BLK	Categorical	0	4322	(1718, 367057)
3	Customer's sex	SEX	Categorical	244	2	(1=Male, 2=Female)
4	Customer's age	AGE	Discrete	244	18	(15, 120)
5	Time of transaction	TIME	Discrete	0	8	(0, 21)
6	Transaction count	CNT	Discrete	0		(1, 10)
7	Amount of transaction	AMT	Continuous	0		(1010, 264840)
8	Customer's home zip code	ZIP	Categorical	1018	11043	(01000,63644)
9	Day of transaction	DAY	Categorical	0	7	(1,7)

Table **3.1**에 제시된 변수 중 거래 요일에 대한 데이터는 원 데이터에서 매출월, 매출일 정보를 기반으로 산출되었으며 매출월, 매출일이 특정된 거래 정보는 개인 정보 노출 위험을 높일 수 있어 본 연구에서는 요일 정보를 유지하고 매출월과 매출일의 정보는 제외하였다. 각각의 변수에 대하여 살펴보면, 먼저 업종 코드는 고객이 신용카드를 사용한 가맹점을 상업 분류에 따라 분류하여 코드를 부여한 값이며, 블록 코드는 매출 가맹점의 지리적 위치인 위도와 경도에 따라 블록화하여 부여한 코드를 의미한다. 실

제 블록 코드와 지리적 위치 매핑 정보는 나이스 지니데이타 내부 자료로 외부에 공개되지는 않는다. 나이 변수는 15세부터 5세 단위로 통합되어 집계 처리되어 제공되었으며, 결측치는 999로 정의되었다. 결제 시간 변수도 정확한 매출 시간이 아닌 0시부터 24시까지의 시간을 3시간 단위로 통합한 뒤 합산 집계되었다. Table **3.1**의 나이 및 결제 시간의 정보는 연속적인 값이 구간화되어 제공된 변수들이므로 목적에 따라 연속형 변수 또는 이산형 변수로 처리가 가능하다. 결제 시간의 정보는 매출월, 매출일 정보와 결합하면 연속적 시간 변수로 활용이 가능하나 본 연구에서는 매출월, 매출일 정보를 활용하지 않으므로 결제 시간은 범주형 변수로 처리하였다.

3.2. 신용카드 거래 데이터의 전처리

2절에서 소개한 재현 데이터 생성 방법을 적용하기에 앞서 원 데이터의 결측치 및 특이치를 확인하고 각 변수의 조합에 따른 유일한 레코드를 확인하여 원 데이터 자체의 노출 위험을 조절한 뒤 재현 데이터 생성 방법을 적용하고자 한다. 3.1절에서 살펴본 바와 같이 서울특별시의 1월 신용카드 거래 데이터 약 851만 건 중 5만건을 추출하여 전체의 0.59% 정도이며 원 데이터에서 표본 추출 시 원 데이터 자체에서 유일한 레코드는 제외되어 본 연구에서 활용되는 데이터의 유일성이 실제 원 데이터의 유일성을 의미하지 않는다. 하지만 본 연구에서는 재현 데이터 생성 방법의 성능 비교를 목적으로 제공 받은 5만건의 데이터를 원 데이터로 가정하고 재현 데이터에 대한 노출 측정 및 유용성 측도를 계산하였다. 또한 재현 데이터의 성능 비교에 초점을 두고 있어 추가적인 결측치 처리 방법을 적용하지 않고 결측치 발생 표본은 재현 대상 데이터에서 제외하였다. 따라서 원 데이터의 집 우편번호가 누락된 1018개의 행을 원 데이터에서 제외하였다. 참고로 집 우편번호가 누락된 1018개의 행에 성별 및 나이가 결측된 244개의 표본이 모두 포함되어 추가적인 결측치 제외는 발생하지 않았다. 결측치가 제거된 데이터 내에 나이 변수의 경우 15세 1명, 95세 1명, 120세 1명으로 신용 카드 거래 데이터 내에 준 개인 식별자로 인식 될 수 있어 3개의 행을 추가적으로 제외하였다. 이 결과의 재현 대상 데이터는 48,979건이다. 추가로 개별 고객의 거래 데이터로 3시간 단위로 Table **3.1**의 변수에 따라 합산 집계가 되었으나 결제 횟수 변수의 경우 48,979개의 거래 데이터 중 47,062개의 거래, 즉 96.08%의 거래가 결제 횟수가 1로 나타났으며 결제 횟수가 2와 3인 경우까지 포함할 경우, 전체 거래 데이터의 99.92%에 해당한다. 본 연구에서는 유일한 레코드가 나타날 가능성을 낮추기 위하여 결제 횟수의 특이값들을 제거하여 표본을 줄이기 보다는 활용성을 높이기 위해 결제 횟수가 1보다 큰 경우, 결제 금액을 결제 횟수로 나눈 금액을 결제 금액으로 산정하고 결제 횟수 만큼의 레코드를 재현 대상 데이터에 추가한 뒤 기존 합산 집계된 데이터를 제외하는 형식으로 전처리를 진행하였다. 전처리 후 결제 횟수는 모두 1이 되므로 실제 재현 데이터 생성 시 생성 모형에서 결제 횟수 변수는 제외되었다. 위의 전처리 후 데이터의 수는 48,979개에서 51,133개로 2,154개의 레코드가 증가하였다.

신용카드 거래 데이터에서 업종 코드, 블록 코드와 집 우편번호의 경우, 매우 큰 범주의 수가 존재하여 노출 위험이 높은 범주라 볼 수 있다. 많은 범주의 수를 갖는 변수들에 대하여 변수의 조합별로 유일하게 존재하는 레코드의 수를 확인하여 Table **3.2**에 요약하였다. 업종 코드, 블록 코드, 집 우편번호 3가지의 변수만 조합하여도 51,133개의 데이터 중 37,770개의 레코드가 유일하게 나타나 범주의 조합만으로도 노출의 위험이 높은 것을 확인할 수 있다. 이를 완화하기 위하여 블록 코드는 뒤의 3자리 코드를 "000"으로 변환하고 집 우편번호의 뒤의 2자리를 "00"으로 처리하여 상위 범주로 통합하여 표현하는 절차를 적용하였다. 통합 결과, 블록 코드는 4322개의 범주에서 253개의 범주로 통합되었으며 집 우편번호는 11043개의 범주에서 457개의 범주로 통합되었다. 상위 범주로 통합 처리 후의 변수 조합에 따른 유일한 레코드의 수를 Table **3.2**에 나타내었다. 원 데이터와 비교 시, 개별 변수의 유일 레코드 수는 크게 감소하였으나 여전히 변수 조합에 따른 유일한 레코드는 높게 나타났다. 실제 개인 정보 노출 위험

이 높은 데이터에서는 유일한 레코드가 발생하지 않도록 제외 처리를 하거나 더 높은 상위 범주로의 통합을 고려해야 한다. 하지만 본 연구는 원 데이터의 0.59%에 해당하는 샘플이며 원 데이터 내에 유일하지 않은 레코드만 표본으로 추출하여 실제 노출 위험은 낮으므로 재현 데이터 생성의 노출 위험 측도 측정 및 유용성 측도 비교를 위하여 추가적인 비식별 처리를 진행하지는 않았다. 전처리 후의 재현 대상 데이터에 대한 변수의 현황을 Table **3.3**에 요약하였다.

Table 3.2 Number of unique records in original data

No.	Variable	Original	Modified
1	TYPE	12	12
2	BLK	765	7
3	ZIP	3881	37
4	(TYPE, BLK)	1375	380
5	(TYPE, ZIP)	20552	3066
6	(BLK, ZIP)	36316	6418
7	(TYPE, BLK, ZIP)	37770	16588

Table 3.3 Summary of variables in credit card transaction data after preprocessing.

No.	Variable name	Type	#(Category)	Range
1	TYPE	Categorical	158	(1, 385)
2	BLK	Categorical	253	(1000, 367000)
3	SEX	Categorical	2	(1=Male, 2=Female)
4	AGE	Discrete	15	(20, 90)
5	TIME	Discrete	8	(0, 21)
6	AMT	Continuous		(600, 264840)
7	ZIP	Categorical	457	(01000,63600)
8	DAY	Categorical	7	(1,7)

4. 신용카드 거래 데이터에 대한 재현 데이터 생성 방법론의 비교

4.1. 재현 데이터 생성 성능 비교 측도

재현 데이터의 성능을 비교하는 측도들은 특정 레코드의 식별 가능성을 나타내는 노출 위험 (disclosure risk) 측도와 원 데이터와 비교하여 생성된 재현 데이터의 활용성 또는 정보 손실을 측정하는 유용성 (utility) 지표로 구분된다. 노출 위험의 측도로는 k-익명성 및 l-다양성 등의 측도가 정의되어 있으며 보통 원 데이터에 비식별 처리 후 제공되는 가명 데이터의 노출 위험을 측정하는데 주로 활용된다. 하지만 재현 데이터의 경우, 원 데이터의 분포를 추정한 뒤, 추정된 분포에서 랜덤하게 난수를 추출한다는 관점에서 노출 위험에 대한 우려가 낮다고 인식되었다. 이러한 인식은 원 데이터가 모두 연속형 데이터인 경우, 원 데이터와 같은 값을 갖는 레코드가 나타날 확률이 이론적으로 0이라는 사실에 기인한다. 하지만 범주형 변수만 포함된 데이터에 대해서는 동일한 범주로 구성된 원 데이터와 동일한 레코드가 나타나게 되므로 Taub와 Elliot (2019)에서는 재현 데이터에서도 노출 위험을 측정해야 하며 목표 대상 특성 식별 확률 (targeted corrected attribution probability, TCAP)을 통하여 재현 데이터의 노출 위험을 측정할 것을 제안하였다. 따라서 본 논문에서는 각 재현 데이터 생성 방법을 신용카드 거래 데이터에 적용하고 TCAP을 이용하여 노출 위험을 측정하였다. 또한 재현 데이터의 유용성 지표로는 Woo 등 (2009)에서 제안한 성향 점수 기반의 pMSE (propensity score-based mean squared error)와 ROE (ratio of estimates)를 고려하였다.

4.1.1. 노출 위험 측도

본 연구에서 노출 위험 측도로 고려한 TCAP은 다음과 같은 외부 공격자를 전제로 정의한다. 외부 공격자는 생성된 재현 데이터에 대한 모든 정보와 일부 변수에 대하여 원 데이터와 동일한 정보를 지니고 있다고 가정한다. 여기서 외부 공격자가 알고 있는 원 데이터의 변수들을 식별자 변수 (key variable)로 정의하고 재현 데이터와 원 데이터의 변수 정보를 활용하여 알아 내고자 하는 정보를 지닌 변수를 목표 변수 (target variable)라 정의한다.

Taub와 Elliot (2019)에서 제안된 TCAP은 외부 공격자가 재현 데이터와 원 데이터의 식별자 변수 정보를 통하여 원 데이터의 목표 변수의 정보를 얻는 노출 위험을 측정하는 측도로 외부 공격자가 재현 데이터에서 제공된 목표 변수의 정보를 통해 공격 대상을 축소하고 이를 통해 원 데이터의 정보 획득 가능성을 높이는 상황 하에서 목표 변수 특성 식별 확률을 측정한다. 보다 자세히 살펴보면, 재현 데이터에서 제공된 정보를 통하여 공격 대상을 축소하기 위해 외부 공격자는 동일 식별자 내 특성 확률 (within equivalence class attribution probability, WEAP)을 고려한다. 동일 식별자 내 특성 확률은 재현 데이터 정보만 활용하며 다음과 같이 정의된다.

$$WEAP_j = \frac{\sum_{i=1}^{n} I(T_{syn,i} = T_{syn,j}, K_{syn,i} = K_{syn,j})}{\sum_{i=1}^{n} I(K_{syn,i} = K_{syn,j})} \tag{4.1}$$

여기서 $I(A)$는 조건 A를 만족할 경우 1의 값을 나타내며 만족하지 않을 경우 0의 값을 나타내는 지시함수, $K_{syn,i}$는 재현 데이터의 i번째 레코드의 식별자 변수의 정보, $T_{syn,i}$는 재현 데이터의 목표 변수의 정보를 나타낸다. WEAP는 각각의 레코드마다 계산되며 식별자가 동일한 데이터들 내에 목표 변수의 정보가 모두 동일할 경우에 1의 값을 갖는 지표이다.

TCAP은 외부 공격자가 높은 WEAP를 갖는 레코드를 공격 대상으로 식별하는 상황을 가정하여 다음과 같이 TCAP을 정의한다.

$$TCAP_j = \frac{\sum_{i=1}^{n} I(T_{obs,i} = T_{syn,j}, K_{obs,i} = K_{syn,j})}{\sum_{i=1}^{n} I(K_{obs,i} = K_{syn,j})} \tag{4.2}$$

여기서 K_{obs}, T_{obs}은 각각 원 데이터의 식별자 및 목표 변수 정보, K_{syn}, T_{syn}은 각각 재현 데이터의 식별자 및 목표 변수 정보를 나타낸다. $TCAP_j$는 재현 데이터의 j번째 레코드와 동일한 식별자 변수의 값을 갖는 원 데이터의 레코드들 중에서 목표 변수의 값까지 동일한 레코드의 비율을 측정하므로 기존의 l-다양성 (Machanavajjhala 등, 2007) 측도 측면에서 공격 대상이 된 레코드들의 목표 변수가 *1*-다양성일 가능성을 측정하는 것으로 이해할 수 있다. 즉, TCAP 값이 1에 가까워지면 해당 레코드의 목표 변수의 노출 위험이 높아지는 것을 의미한다. 반대로 TCAP가 0에 가까워지면 재현 데이터와 식별자가 동일한 원 데이터의 레코드들이 여러 목표 변수의 값을 갖는 다는 것을 의미하여 재현 데이터의 노출 위험이 낮다는 것을 의미한다.

본 연구에서는 신용카드 거래 데이터에 대한 재현 데이터의 노출 위험을 측정하기 위하여 목표 변수로는 집 우편 번호 (ZIP)을 고려하였으며 식별자 변수로는 $\{T, B, S, A, TI\}$, $\{T, B, S, A, TI, D\}$ $\{T, B, S, A, TI, D, AM(100)\}$, $\{T, B, S, A, TI, D, AM(1000)\}$, $\{T, B, S, A, TI, D, AM(10000)\}$[6] 로 5개의 변수 조합을 고려하였다. 여기서 $AM(100)$, $AM(1000)$, $AM(10000)$는 AMT 변수에서 각각 변수의 값이 100원 단위, 1000원 단위, 10000원 단위가 되도록 반올림 연산이 적용된 것을 나타낸다. 공격 대상 식별에 사용되는 WEAP의 기준 값으로는 1.0을 적용하였다.

[6] T: TYPE, B: BLK, S: SEX, A: AGE, TI: TIME, D: DAY, AM: AMT

4.1.2. 유용성 측도

본 논문에서는 재현 데이터의 유용성을 측정하기 위하여 Woo 등 (2009)의 pMSE와 관심 대상 통계량 값들의 비율인 ROE를 고려하였다. 먼저 pMSE 측도를 살펴 보면, 원 데이터와 재현 데이터의 특정 통계량의 유사도를 계산하는 것이 아니라 전체적인 관점에서의 유사성을 판단하기 위해 원 데이터와 재현 데이터의 성향 점수 (propensity score)를 기준으로 유용성을 측정한다. 여기서 성향 점수는 주로 관찰 연구 또는 인과 추론에서 사용되는 개념으로 설명 변수 (X)가 주어졌을 때 반응 변수 (Y)의 조건부 확률을 의미하며 pMSE 측도에서 반응 변수는 재현 데이터인 경우 1, 원 데이터인 경우 0의 값을 갖는 이진 변수로 정의된다. 원 데이터와 재현 데이터가 동일한 레코드 수 n을 갖는다고 가정할 때, pMSE 측도는

$$pMSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - 0.5)^2$$

로 정의된다. 여기서 $p_i = \mathbb{P}[Y_i = 1 \,|\, X_i]$로 i번째 레코드의 성향 점수를 나타낸다. 여기서 $N = 2n$으로 원 데이터와 재현 데이터가 통합된 데이터의 레코드 수를 나타낸다. 위의 정의를 살펴 보면, 원 데이터와 재현 데이터의 분류 확률이 0.5에 가까울 경우에는 두 데이터를 구분할 수 없는 상황으로 두 데이터의 유사도가 높다고 판단하여 pMSE가 낮은 값을 갖도록 하고 어느 하나의 데이터로 분류가 될 확률이 높아지는 상황 하에서는 높은 pMSE의 값을 갖도록 설계되었다. 실제 데이터에 적용 시 pMSE를 산출 하는 과정은 다음과 같다.

(1) 원 데이터와 재현 데이터를 하나로 통합한 뒤 원 데이터와 재현 데이터를 0과 1로 구분하는 지시 변수를 추가한다.

(2) 통합된 데이터의 변수들을 설명 변수 (X)로 고려하고 원 데이터와 재현 데이터를 구분하는 지시 변수를 반응 변수 (Y)로 하여 로지스틱 회귀모형을 적용한다.

(3) 추정된 로지스틱 회귀모형을 이용하여 각각의 레코드마다 $\hat{p}_i$를 산출한다.

(4) 산출된 $\hat{p}_i$를 이용하여 pMSE를 계산한다.

본 연구에서는 고려한 노출 위험 측도 TCAP은 0과 1 사이의 값을 나타내므로 유용성 측도를 노출 위험과 비교하기 위하여 pMSE를 직접 비교하는 것이 아니라 $1 - 4 \cdot pMSE$의 지표를 산출하여 비교하였다. $1 - 4 \cdot pMSE$의 값은 pMSE의 값이 최솟값 0일 경우 1의 값을 갖으며 최댓값인 0.25일 경우 0의 값을 갖는 지표로 높을 수록 유용성이 높다고 판단하는 지표이다.

추가적으로 특정 통계량에 대하여 원 데이터와 재현 데이터의 비율을 비교하는 측도인 ROE 측도를 고려하였다. ROE 측도를 정의하기 위하여 $U_{org,k}$와 $U_{syn,k}$는 각각 원 데이터와 재현 데이터로부터 계산된 통계량 U_k를 나타낸다. 이를 이용하여 유용성 측도 ROE는

$$ROE_k = \frac{\min(U_{org,k}, U_{syn,k})}{\max(U_{org,k}, U_{syn,k})}$$

로 정의되며 0과 1 사이의 값을 갖는다.

본 연구에서는 ROE를 산출하기 위하여 통계량으로 업종별 평균 결제 금액, 블록별 평균 결제 금액, 결제 시간별 평균 결제 금액 요일별 평균 결제 금액, 성별 구성 비율, 나이 구성 비율을 고려하였다. 최종적인 유용성 지표는 $1 - 4 \cdot pMSE$와 ROE 측도들의 평균으로 산출하였다.

4.2. 재현 데이터 생성 및 성능 비교 결과

이 절에서는 3.2절에서 설명한 전처리 과정을 적용한 신용카드 거래 데이터를 원 데이터로 고려하여 2절에서 소개한 5가지의 재현 데이터 생성 방법론을 적용하고자 한다. 또한 각 방법에 따라 생성

된 재현 데이터를 이용하여 4.1절에서 소개한 TCAP 노출 위험 측도와 유용성 지표를 통하여 재현 데이터 생성 방법의 성능을 비교하고자 한다. 본 연구에서는 2절에서 설명한 재현 데이터 생성 방법들의 실제 적용을 위하여 구현된 라이브러리를 활용하였고 이에 대하여 간략히 소개하고자 한다. 먼저 재현 데이터 생성 방법으로 고려한 synthpop은 R 패키지로 구현된 `synthpop`을 이용할 수 있으며 파이썬으로 구현된 `py-synthpop`[7]도 사용할 수 있다. 본 연구에서는 인공신경망 기반의 모형들과 비교의 용이성 및 플랫폼 영향을 배제하기 위하여 동일하게 파이썬 환경에서 재현 데이터 생성 방법들을 적용하였다. VAEM은 원 논문의 저자가 제공하는 라이브러리가 현재의 환경에서 잘 호환 되지 않으며 저자의 GitHub에서 제공되는 코드의 오류를 확인하고 PyTorch로 직접 구현하여 적용하였다. TableGAN과 CTGAN 모형은 DataCebo에서 개발하여 배포하는 SDGym 라이브러리[8]를 활용하였다. CTAB-GAN은 원 논문의 저자가 구현하여 배포하는 라이브러리[9]를 이용하여 재현 데이터를 생성하였다.

각각의 재현 데이터 생성 방법의 적용 시 고려한 사항을 요약하면 다음과 같다. `py-synthpop` 적용 시에는 업종 코드, 블록 코드, 성별, 나이, 결제 시간, 결제 금액, 집 우편번호, 요일의 순서로 순차적으로 학습하고 재현 데이터를 생성하였다. VAEM의 경우, 직접 구현한 PyTorch의 클래스를 이용하여 학습하였으며 Adam 알고리즘 (Kingma와 Ba, 2014)과 학습률 0.0002를 적용하였고 각 변수에 대한 VAE 모형과 잠재 변수에 대한 VAE 모형에서는 2개의 은닉층을 적용하였다. TableGAN과 CTGAN의 경우, SDGym 라이브러리에서 제공하는 기본 설정값을 대부분 적용하였으며 다수의 범주에 대한 모형 학습을 용이하게 하기 위하여 에포크 (epoch)와 미니 배치 크기 (mini-batch size)를 각각 500과 5000 정도로 크게 조정하여 학습을 진행하였다. CTAB-GAN에서는 원 논문의 저자가 고려한 기본 설정을 유지하며 TableGAN, CTGAN과 유사하게 미니 배치의 크기를 1024로 고려하고 에포크는 500으로 설정하여 학습을 진행하였다.

본 연구에서 고려한 신용카드 거래 데이터의 경우, 대부분의 변수가 범주형 변수로 구성되어 있어 변수들의 다변량적 특성의 확인을 위하여 가능한 모든 조합의 테이블을 나타내기에는 제약이 있다. 먼저 생성된 재현 데이터가 원 데이터의 주변 분포와 유사한지 확인하기 위하여 Figure 4.1과 Figure 4.2를 통하여 확인하였다. Figure 4.1과 Figure 4.2에서 다수의 범주를 갖는 변수의 경우는 가장 빈도가 높은 5개의 범주에 대하여 막대 그래프를 통하여 분포를 확인하였으며 소수의 범주 (10 미만)인 경우는 모든 범주에 대한 분포를 확인하였다. 연속형 변수인 결제 금액 변수는 로그 변환 후 추정된 확률 밀도 함수의 그래프를 나타내었다.

Figure 4.1에는 다수의 범주를 갖는 변수에 대한 분포를 확인할 수 있으며 `synthpop`이 원 데이터와 가장 유사하게 재현 데이터를 생성하였음을 확인할 수 있다. VAEM 방법은 다수의 범주를 갖는 업종 코드, 블록 코드, 집 우편번호에서 성능 저하가 심하게 나타남을 확인할 수 있었으며 AGE 및 Figure 4.2의 소수의 범주와 연속형 변수에 대해서는 재현 성능 저하가 크게 나타나지 않음을 확인하였다. VAEM 모형의 성능은 잠재 변수로 축약하고 복원하는 과정에서 사전 분포에 대한 거리와 복원에 대한 손실함수의 최적화에 있어서 저빈도 범주가 일정 부분 무시되어 학습된 결과라 해석된다. GAN 구조 기반의 TableGAN, CTGAN, CTAB-GAN 중에서는 전반적으로 CTGAN이 주변 분포 재현 관점에서 상대적으로 우수하게 나타났으며 CTAB-GAN은 다수의 범주에서는 TableGAN 보다 성능이 우수하게 나타나지만 소수의 범주에서는 TableGAN이 원 데이터의 분포와 더 유사하게 나타났다.

본 연구에서는 분포의 유사성을 시각적으로 확인하는 것에 더해 4.1절에서 소개한 노출 위험 측도인 TCAP을 계산하여 Table **4.1**과 Table **4.2**에 요약하였으며 유용성 측도인 pMSE 및 ROE를 계산하여

[7] https://github.com/hazy/synthpop/
[8] https://github.com/sdv-dev/SDGym
[9] https://github.com/zhao-zilong/CTAB-GAN

Table **4.3**에 제시하였다. 노출 위험 측도인 TCAP을 Table **4.1**과 Table **4.2**에 나누어 제시한 이유는 다음과 같다. 실제 재현 데이터 생성 방법 적용 후 정리하는 과정에서 나타난 현상으로 본 연구에서 고려한 신용 카드 거래 데이터는 각 레코드 조합에 대한 빈도의 수가 작아 하나의 변수의 값만 바뀌어도 해당 조합이 원 데이터에 존재하지 않는 경우가 다수 발생함을 확인하였다. 이러한 이유로 생성된 재현 데이터의 대부분의 레코드에서 TCAP이 0으로 나타나면서 산출된 TCAP 측도의 값이 매우 낮게 나타나게 된다. 원래의 의미적으로는 외부 공격자가 WEAP로 식별한 목표 대상의 식별 확률의 관점에서는 Table **4.1**에서 제공한 수치가 타당한 의미이나 Figure 4.3에서 나타낸 것처럼 다수의 0을 포함하는 분포를 보여 0이 아닌 TCAP을 갖는 레코드들에 대한 평균, 즉 외부 공격자가 식별하였다고 생각한 목표 특성이 원 데이터에 존재하는 경우로만 특정하여 TCAP을 계산한 결과가 Table **4.2**에 제시된 결과이다. TCAP의 계산에 있어 본 연구에서는 신용카드 고객의 매출이 일어난 업종 코드, 블록 코드, 결제 시간, 결제 요일 정보를 외부 공격자가 지니고 있다고 가정하고 고객의 성별, 나이, 집 우편 번호를 식별할 수 있는지에 대한 노출 위험을 측정하였다. 노출 위험 관점에서는 `synthpop`이 모든 케이스에 대하여 5가지의 방법 중 가장 높은 값을 나타냈으며 GAN 기반의 방법들이 평균적으로 VAEM 보다 높은 위험도를 나타내었다. 하지만 VAEM과 TableGAN의 경우, 업종코드, 블록코드, 집 우편번호와 같이 다수의 범주를 갖는 경우에 위험도가 낮으나 상대적으로 유용성 지표에서 매우 낮게 나타남을 Table **4.3**를 통하여 확인할 수 있다.

Table **4.3**에 제시한 유용성 지표로 pMSE를 산출하였고 ROE의 기준으로 각 변수에 대한 주변 분포의 유사도를 측정하는 범주별 빈도에 대한 원 데이터와 재현 데이터의 비율 (예: ROE-T, 업종별 상대 빈도), 각 범주형 변수의 범주에 따른 결제 금액의 평균 (예: ROE-T/AM100, 업종별 결제 금액의 평균)을 고려하였다. 전체적인 노출 위험과 유용성 측도의 비교를 위하여 각 표에서 제시한 노출 위험 및 유용성 지표의 평균을 통하여 위험도-유용성 Figure (risk-utility plot)을 Figure 4.4에 제시하였다. Figure 4.4의 (b)에서 나타낸 0이 아닌 TCAP의 평균과 유용성을 확인하면 노출 위험과 유용성이 서로 상반 (trade-off) 관계에 있음을 확인할 수 있으며 노출 위험이 높을수록 높은 유용성을 지니는 것을 확인할 수 있다. 특히 위험도-유용성 그림을 통하여 확인할 수 있는 사실은 신용카드 거래 데이터에 대하여 CTAB-GAN 방법보다 CTGAN 방법이 더 낮은 위험과 높은 유용성을 보임을 확인할 수 있다. Figure 4.4의 (a)에서 제시한 TCAP은 공격 대상 레코드들의 전체적이 관점에서의 TCAP으로 공격 대상 중 임의의 한 레코드를 선택하여 목표 대상의 특성을 식별할 위험을 나타내는 측도이다. 따라서 본 연구에서 진행한 신용카드 거래 데이터의 경우에는 전체적인 관점에서는 재현 데이터를 사용할 경우 노출 위험이 낮다고 판단할 수 있다.

5. 결론

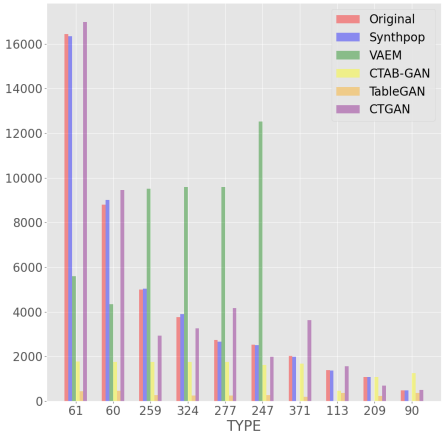
본 연구에서는 최근 개발된 재현 데이터 생성 방법론에 대하여 소개하고 신용카드 거래 데이터에 대하여 재현 데이터 생성 방법을 적용하였다. 재현 데이터 생성 방법들의 비교를 위하여 재현 데이터에 대한 노출 위험을 측정하는 목표 대상 특성 식별 확률과 재현 데이터의 유용성을 측정하는 성향 점수 기반의 MSE, 통계량의 비를 고려하였으며 보다 용이한 비교를 위하여 위험도-유용성 그림에 각 방법을 나타내었다. 본 연구의 비교 결과, 범주의 수가 많으며 저빈도 범주가 다수 존재하는 신용카드 거래 데이터에 대하여 `synthpop`이 노출 위험은 높으나 유용성 측면에서도 가장 높은 값을 나타냈으며 인공신경망 기반의 모형에서는 CTGAN이 높은 유용성을 나타냈으며 전체 방법 중 두 번째로 높은 노출 위험을 나타내었다. 또한 CTGAN에 기반한 CTAB-GAN의 경우에는 CTGAN과 유사한 성능을 보이긴 하였으나 전반적으로 낮은 위험도와 낮은 유용성을 나타내었다. VAEM은 잠재 변수로 축약하고 재복원하는 과정에서 저빈도 범주에 대한 복원 성능이 낮아 전체적으로 유용성이 낮게 나타났음을 확인하였으며 이에

Table 4.1 Summary of average of TCAP

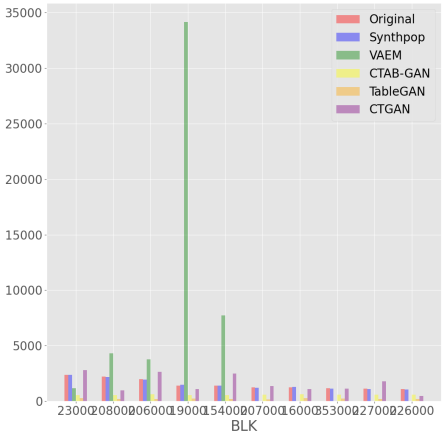
Key variable	Target	synthpop	VAEM	TableGAN	CTGAN	CTAB-GAN
(T,B,TI,D)	(S)	0.8004	0.6375	0.7575	0.7046	0.6924
(T,B,TI,D)	(A)	0.7223	0.3231	0.4790	0.4422	0.4377
(T,B,TI,D)	(Z)	0.7736	0.0750	0.3919	0.4512	0.2796
(T,B,TI,D)	(S,A)	0.7240	0.2416	0.4164	0.4229	0.3950
(T,B,TI,D)	(S,Z)	0.7391	0.0670	0.5833	0.3674	0.2197
(T,B,TI,D)	(A,Z)	0.7269	0.0584	-	0.5827	0.1830
(T,B,TI,D)	(S,A,Z)	0.7603	0.0714	-	0.0235	0.0828
(T,B,TI,D, AM100)	(S)	0.9590	0.9328	1.0000	0.9434	0.9786
(T,B,TI,D, AM100)	(A)	0.9456	0.8437	1.0000	0.9342	1.0000
(T,B,TI,D, AM100)	(Z)	0.9533	0.8333	-	0.7348	0.5833
(T,B,TI,D, AM100)	(S,A)	0.9510	0.7771	1.0000	0.9133	1.0000
(T,B,TI,D, AM100)	(S,Z)	0.9532	0.1667	-	0.5139	0.5833
(T,B,TI,D, AM100)	(A,Z)	0.9579	-	-	-	-
(T,B,TI,D, AM100)	(S,A,Z)	0.9602	-	-	-	-
(T,B,TI,D, AM1000)	(S)	0.9056	0.8569	0.9046	0.8559	0.8804
(T,B,TI,D, AM1000)	(A)	0.8744	0.7096	0.8333	0.7341	0.7441
(T,B,TI,D, AM1000)	(Z)	0.9085	0.4189	-	0.8785	0.5694
(T,B,TI,D, AM1000)	(S,A)	0.8821	0.7076	0.7500	0.7247	0.6972
(T,B,TI,D, AM1000)	(S,Z)	0.9052	0.3208	-	0.8733	0.3056
(T,B,TI,D, AM1000)	(A,Z)	0.9075	-	-	0.6667	-
(T,B,TI,D, AM1000)	(S,A,Z)	0.9094	-	-	0.3333	-
Average		0.0696	0.0176	0.0005	0.0148	0.0026

Table 4.2 Summary of average of nonzero TCAP

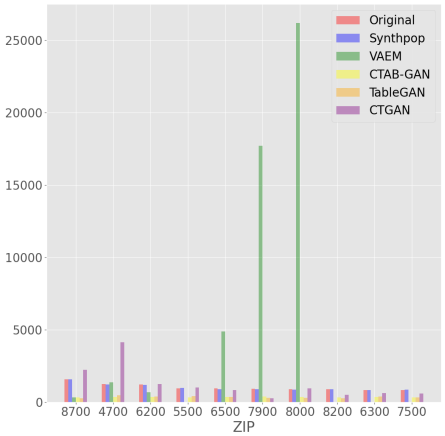
Key variable	Target	synthpop	VAEM	TableGAN	CTGAN	CTAB-GAN
(T,B,TI,D)	(S)	0.8004	0.6375	0.7575	0.7046	0.7038
(T,B,TI,D)	(A)	0.7223	0.3231	0.4790	0.4422	0.4587
(T,B,TI,D)	(Z)	0.7736	0.0750	0.3919	0.4512	0.3719
(T,B,TI,D)	(S,A)	0.7240	0.2416	0.4164	0.4229	0.4162
(T,B,TI,D)	(S,Z)	0.7391	0.0670	0.5833	0.3674	0.3726
(T,B,TI,D)	(A,Z)	0.7269	0.0584	-	0.5827	0.3562
(T,B,TI,D)	(S,A,Z)	0.7603	0.0714	-	0.0235	0.4333
(T,B,TI,D, AM100)	(S)	0.9590	0.9328	1.0000	0.9434	0.9500
(T,B,TI,D, AM100)	(A)	0.9456	0.8437	1.0000	0.9342	1.0000
(T,B,TI,D, AM100)	(Z)	0.9533	0.8333	-	0.7348	-
(T,B,TI,D, AM100)	(S,A)	0.9510	0.7771	1.0000	0.9133	1.0000
(T,B,TI,D, AM100)	(S,Z)	0.9532	0.1667	-	0.5139	-
(T,B,TI,D, AM100)	(A,Z)	0.9579	-	-	-	-
(T,B,TI,D, AM100)	(S,A,Z)	0.9602	-	-	-	-
(T,B,TI,D, AM1000)	(S)	0.9056	0.8569	0.9046	0.8559	0.8968
(T,B,TI,D, AM1000)	(A)	0.8744	0.7096	0.8333	0.7341	0.7302
(T,B,TI,D, AM1000)	(Z)	0.9085	0.4189	-	0.8785	0.7500
(T,B,TI,D, AM1000)	(S,A)	0.8821	0.7076	0.7500	0.7247	0.7130
(T,B,TI,D, AM1000)	(S,Z)	0.9052	0.3208	-	0.8733	0.7500
(T,B,TI,D, AM1000)	(A,Z)	0.9075	-	-	0.6667	-
(T,B,TI,D, AM1000)	(S,A,Z)	0.9094	-	-	0.3333	-
Average		0.8676	0.4730	0.7378	0.6369	0.5666



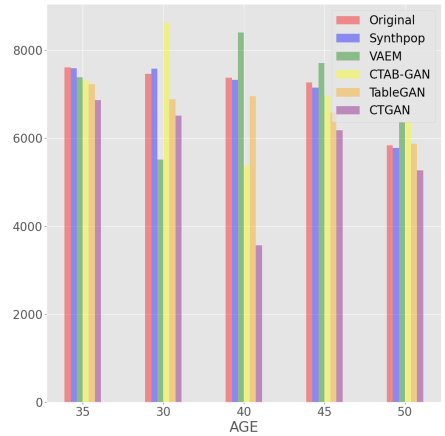
(a) TYPE (Top 5)



(b) BLK (Top 5)

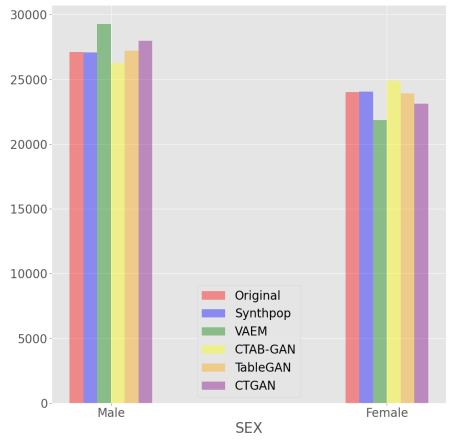


(c) ZIP (Top 5)

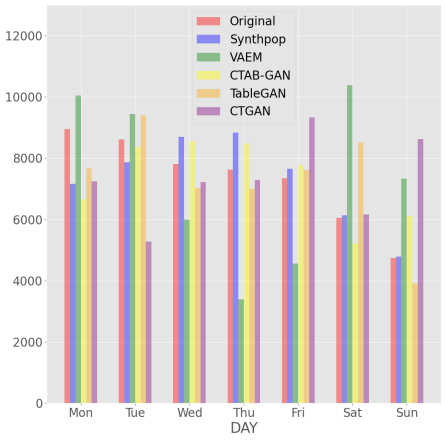


(d) AGE (Top 5)

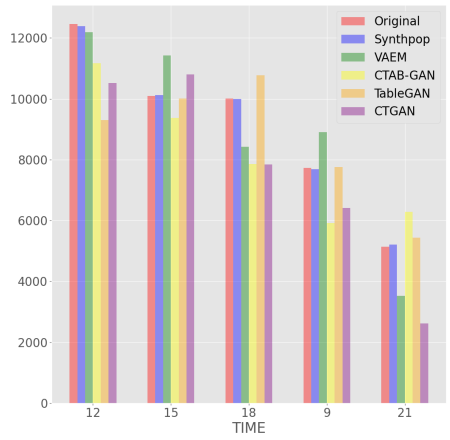
Figure 4.1 Comparison of distributions for TYPE, BLK, ZIP and AGE



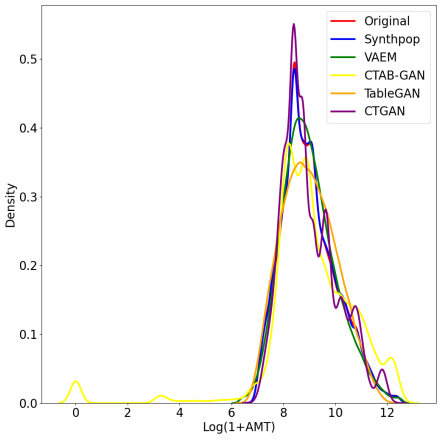
(a) SEX



(b) DAY



(c) TIME (Top 5)

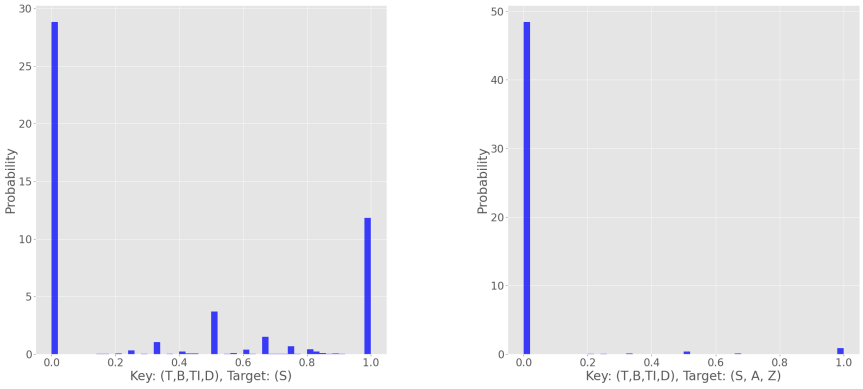


(d) Log(AMT)

Figure 4.2 Comparison of distributions for SEX, DAY, TIME and Log(AMT)

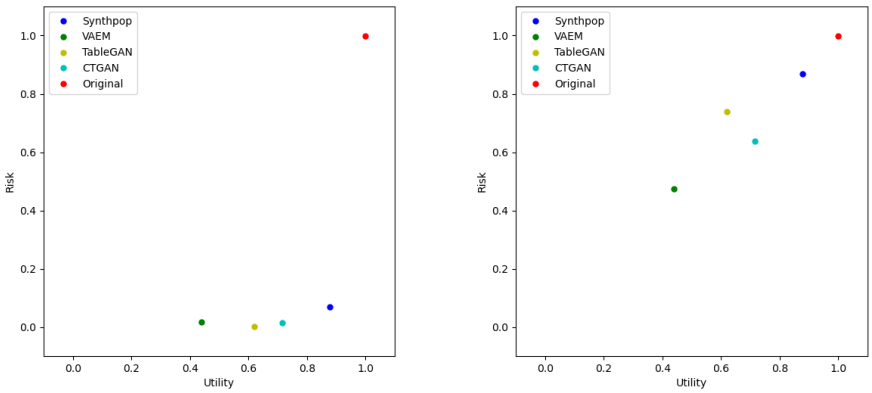
Table 4.3 Summary of utility measures

Utility	synthpop	VAEM	TableGAN	CTGAN	CTAB-GAN
pMSE	0.9967	0.0231	0.1838	0.7322	0.3428
ROE-T	0.7906	0.0142	0.1532	0.6889	0.1853
ROE-B	0.8599	0.0069	0.3052	0.5698	0.3935
ROE-S	0.9984	0.9181	0.9961	0.9661	0.9705
ROE-A	0.7914	0.7064	0.5447	0.6331	0.4600
ROE-TI	0.9925	0.7778	0.6978	0.6768	0.7396
ROE-D	0.9883	0.6375	0.8457	0.8089	0.9298
ROE-Z	0.8122	0.0045	0.3946	0.6173	0.3732
ROE-T/AM100	0.6513	0.0243	0.5352	0.5142	0.5251
ROE-B/AM100	0.7918	0.0143	0.6830	0.5399	0.6746
ROE-S/AM100	0.9928	0.9689	0.9327	0.9828	0.9088
ROE-A/AM100	0.7874	0.6105	0.6672	0.6551	0.5948
ROE-TI/AM100	0.9758	0.8952	0.8009	0.8153	0.8706
ROE-D/AM100	0.9788	0.9570	0.9337	0.9303	0.9048
ROE-Z/AM100	0.7614	0.0129	0.6284	0.6033	0.7095
Average	0.8780	0.4381	0.6201	0.7156	0.6387



(a) Target: (S) (b) Target: (S,A,Z)

Figure 4.3 Distribution of TCAP with key variables TYPE, BLK, TIME and DAY



(a) Average of TCAPs (b) Average of nonzero TCAPs

Figure 4.4 Risk-Utility Plot

따라 노출 위험도 낮게 나타났다. TableGAN의 경우, DCGAN 구조 기반의 모형으로 연속형 변수가 다수 포함될 경우, 원 데이터의 분포 학습이 잘 이루어진다고 알려져 있으나 본 연구에서 적용한 신용카드 거래 데이터와 같이 다수의 범주와 저빈도를 갖는 범주로 구성된 혼합형 데이터에서는 전반적으로 낮은 성능을 나타내었다. 하지만 GAN 기반의 모형은 최소최대 목적함수를 기반으로 학습하여 학습 데이터와 최적화 알고리즘 및 조율 모수에 민감하므로 다른 데이터의 재현 데이터 생성에 대한 성능 비교로 일반화하여 판단하는 것은 주의가 필요하다.

추가로 본 연구에서 고려한 TCAP 측도는 외부 공격자가 l-다양성이 낮은 키 변수 조합에 대하여 공격 대상으로 식별함을 가정하고 산출하는 재현 데이터에 대한 노출 위험 측도로 최근에 개발된 노출 위험 측도이다. 본 연구에서 고려한 신용카드 거래 데이터와 같이 다수의 범주를 갖는 범주형 변수가 많이 포함되고 범주 조합에 대하여 소수의 레코드만 존재할 경우, 공격 대상 변수의 범주 값이 조금만 바뀌어도 TCAP 측도는 다수의 0으로 산출된다. 4절에서 살펴본 바와 TCAP은 0으로 산출된 레코드들과 0이 아닌 레코드들의 서로 떨어져 2개의 그룹으로 나타나게 되므로 단순히 전체 레코드의 평균을 이용하기 보다는 분포적 특성을 고려하여 요약하는 지표를 고려할 필요가 있으며 이를 추후 연구 주제로 고려하고 있다.

참고문헌

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer, New York.

Drechsler, J. and Reiter, J. P. (2011). An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets. *Computational Statistics and Data Analysis*, **55**, 3232–3243.

Drechsler, J. (2011). *Synthetic datasets for statistical disclosure control*. Lecture Notes in Statistics, **201**.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville A. and Bengio Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 2672–2680.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, **313**, 504–507.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, 448–456.

Kendall, A., Gal, Y. and Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7482–7491.

Kim, J., Song, J. and Lim, D. (2020). CT image denoising using inception model. *Journal of the Korean Data Information Science Society*, **31**, 487–501.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980.*

Li, B., Xu, K., Feng, D., Mi, H., Wang, H. and Zhu, J. (2019). Denoising convolutional autoencoder based B-mode ultrasound tongue image feature extraction. *International Conference on Acoustics, Speech and Signal Processing*, 7130–7134.

Lin, Z., Khetan, A., Fanti, G. and Oh, S. (2018). Pacgan: The power of two samples in generative adversarial networks. *Advances in Neural Information Processing Systems*, **31**.

Ma, C., Tschiatschek, S., Hernández-Lobato, J. M., Turner, R. and Zhang, C. (2020). VAEM: A deep generative model for heterogeneous mixed type data. *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, 11237–11247.

Maas, A. L., Hannun, A. Y. and Ng. A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the International Conference on Machine Learning*, **30**.

Machanavajjhala, A., Kifer, D. and Gehrke J. (2007). L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, **1**, 3–es.

Nowok, B., Raab, G, M. and Dibben C. (2016). Synthpop: Bespoke creation of synthetic data in R. *Journal of Statistical Software*, **74**, 1–26.

Park, M-J. and Kim, H. (2016). Statistical disclosure control for public microdata: Present and future. *The Korean Journal of Applied Statistics*, **29**, 1041–1059.

Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H. and Kim, Y. (2018). Data synthesis based on generative adversarial networks. *Proceedings of the VLDB*, **11**, 1071–1083.

Radford, A., Metz, L. and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434.*

Raghunathan, T. E., Lepkowski, J. M., Van Hoewyk, J. and Solenberger, P. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, **27**, 85–95.

Song, J., Kim, J. and Lim, D. (2020). Image restoration using convolutional denoising autoencoder in images. *Journal of the Korean Data Information Science Society*, **31**, 25–40.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The journal of machine learning research*, **15**, 1929–1958.

Taub, J. and Eliot, M. (2019). The synthetic data challenge. *UNECE: Conference of European Statisticians.*

Taub, J., Eliot, M. and Sakshaug, J. W. (2020). The impact of synthetic data generation on data utility with application to the 1991 UK samples of anonymised records. *Transactions on Data Privacy*, **13**, 1–23.

Tomczak, J. and Welling, M. (2017). VAE with a vamp prior. *International Conference on Artificial Intelligence and Statistics*, 1214–1223.

Woo, M-J., Reiter, J. P., Oganian, A. and Karr A. F. (2009). Global measures of data utiliy for microdata masked for disclosure limitation. *The Journal of Privacy and Confidentiality*, **1**, 111–124.

Xu, L., Skoularidou, M., Cuesta-Infante, A. and Veeramachaneni, K. (2019). Modeling tabular data using conditional GAN. *NIPS'19: Proceedings of the 33th International Conference on Neural Information Processing Systems*, 7335–7345.

Zhao, Z., Kunar, A., Birke, R. and Chen, L. Y. (2021). Ctab-gan: Effective table data synthesizing. *Asian conference on machine learning*, 97–112.

Journal of the Korean Data & Information Science Society
2022, **00**(0), 1–24

한국데이터정보과학회지

Comparison study of synthetic data generation methods for credit card transaction data†

Hyunwoo Jung[1] · Younsang Cho[2] · Geonwoo Ko[3] · Jae-ik Song[4] · Donghyeon Yu[5]

[1,2,3,5]Department of Statistics, Inha University [4]NexGen Innovation, NICE ZiniData Co., Ltd.

Received 1 0000, revised 0 0000, accepted 0 0000

Abstract

Synthetic data generation is one of the main topics in data privacy and statistical disclosure control. In this paper, we apply popular synthetic data generation methods such as synthpop, variational autoencoder (VAE), and generative adversarial network (GAN) models to credit card transaction data. We consider the targeted corrected attribution probability (TCAP) for the disclosure-risk measure, and we also consider propensity-score-based mean squared errors (pMSE) and ratio-of-estimates (ROE) for the data utility. As a result, the synthetic data by the synthpop has high disclosure risk and high data utility, while the VAE has the lowest disclosure risk and data utility. For GAN-based models, the conditional tabular GAN (CTGAN) has a relatively lower disclosure risk and similar data utility compared to the synthpop.

Keywords: Credit card transaction, generative adversarial network, synthetic data, synthpop, variational autoencoder.

† This work was supported by the National Information Society Agency.
[1] Master course student, Department of Statistics, Inha University, Incheon 22212, Korea.
[2] Integrated Ph.D. program student, Department of Statistics, Inha University, Incheon 22212, Korea.
[3] Master course student, Department of Statistics, Inha University, Incheon 22212, Korea.
[4] Manager, NexGen Innovation, NICE ZiniData Co., Ltd., Seoul 07242, Korea.
[5] Corresponding author: Associate Professor, Department of Statistics, Inha University, Incheon 22212, Korea. E-mail: dyu@inha.ac.kr