TabMT: Generating Tabular data with Masked Transformers

Kyungseon Lee

January 8, 2024

Seoul National University

Outline

1. Introduction
2. Related Work
3. Methodology
4. Experiment
5. Conclusion

Introduction

Introduction

1. Privacy Protection
   - Prevents personal information exposure by analyzing synthetic data where personal details are unidentifiable.
2. Data Augmentation
   - Generates additional datasets when data is insufficient.

Introduction

1. Challenge
   - Handling missing data.
   - Due to the diversity in data types, we encounter challenges in modeling joint distributions and relationships among variables.
2. Solution :
   - BERT architecture
   - Novel masking method.

Introduction

The previous attempt at solving the challenges.

- GANs, VAEs
  - Challenges of robustness and scalability across different datasets.
- Diffusion model
  - Privacy leakage problem.
- No model has a solution for missing data.

Related Work

Related Work - Attention

	안녕	난	널	좋아해
Hello	0.8	0.1	0.05	0.05
I	0.1	0.6	0.2	0.1
love	0.05	0.2	0.65	0.1
you	0.2	0.1	0.1	0.6

Figure 1: $Q * K^t$: The example of Attention matrix.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^t}{\sqrt{d_k}}\right) V$$

- l : The number of words in one input sentence.
- d : Embedding dimension.
- Q, K, V : $\mathbb{R}^{l \times d}$ matrix, Query, Key, Value. Each matrix is uniquely determined for each sentence and can be updated.

Related Work - Transformer

Process of generating the t-th word in the translated sentence.

$$\hat{Y}_t = D_2(\mathcal{E}(X), D_1(Y_0 \oplus \hat{Y}_1 \oplus ... \oplus \hat{Y}_{t-1}))$$

- $t \in \{1, 2, ..., T+1\}$: T is the number of words in one output sentence.
- X : Embedded sentence to be translated.
- $\hat{Y}_t$: Predicted t-th embedded word token.
- Y : True translated embedded sentence.
- Y_0 , $\hat{Y}_{T+1}$: Start token, End token
- $\mathcal{E}(X) : \mathbb{R}^{I \times d} \rightarrow \mathbb{R}^{I \times d}$, Encoder function.
- $D_1(X) : \mathbb{R}^{t \times d} \rightarrow \mathbb{R}^{T \times d}$, The first layer of decoder function.
- $D_2(X, Y) : \mathbb{R}^{I \times d} \times \mathbb{R}^{t \times d} \rightarrow \mathbb{R}^{I \times d}$, The second layer of decoder function.

BERT-Bidirectional Encoder Representations from Transformers

Before	After
My dog is **hairy**	My dog is **[MASK]**

Figure 2: The example of masked X.

$$\hat{Y}_t = \mathcal{E}(X_m)$$

$$\mathcal{L} = L(\hat{Y}_t, Y_t) \text{ ; loss function}$$

- $X_m \in \mathrm{F}^m$: $\mathbb{R}^{l \times d}$ matrix, F^m is a set of masked fields.
- Y_t : A true t-th vector.
- $\hat{Y}_t$: A prediction vector of t-th word.

Methodology

Methodology

How is the BERT utilized to generate tabular data?

- BERT input : embedded sentence
- TabMT input
  - Categorical variable : Same embedding method of BERT.

$$x_{i,j} \sim N(0, I_d)$$

- $x_{i,j} \in \mathbb{R}^k$: embedded vector of i-th category, j-th class.
- d : embedding dimension.

Methodology

How is the BERT utilized to generate tabular data?

- TabMT input
  - ▶ Numerical variable : K-means Quantizing and Ordered embedding.
  - ▶ After k-means clustering on the values of the nth variable, then replace each value with the mean of the cluster it belongs to.

$$NE(x_n) = \mathrm{r}_n \cdot \mathrm{W}_n^t + b_n$$

$$Q(x) = \underset{\mu}{argmin} \sum_{i=1}^{\alpha} \sum_{\mathrm{x} \in S_i} \|\mathrm{x} - \mu_i\|^2 \quad , \quad \mathrm{r}_n = \frac{Q(x_n) - \min(\mathrm{Q}(\mathrm{x_n}))}{\max(\mathrm{Q}(\mathrm{x_n})) - \min(\mathrm{Q}(\mathrm{x_n}))}$$

- $x_n \in \mathbb{R}^k$ input, unmasked variable of n-th row.
- k : The number of unique unmasked input variables.
- $NE(x_n) : \mathbb{R} \rightarrow \mathbb{R}^{k \times d}$ The embedding function of numerical variable.
- α : The hyperparameter of k-means clustering.

How is the BERT utilized to generate tabular data?

1. In training process, masking probability is not fixed but sampling in uniform distribution.

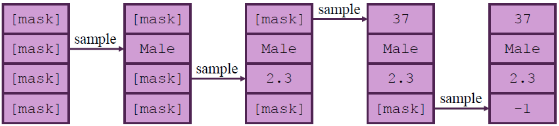
$$P\left(p_m = p\right) \sim U(0,1)$$

$$P\left(|\mathrm{F}_i^m| = k\right) = \int_0^1 \begin{pmatrix} l \\ k \end{pmatrix} p^k (1-p)^{l-k} dp = \frac{1}{l+1}$$

- For each row F_i, a set of unmasked fields F_i^u and a set of masked fields F_i^m
- $\mathrm{p}_m = P\left(\mathrm{F}_{i,j} \in \mathrm{F}_i^m\right)$; masking probability.
- Training uniformly across subset sizes.

Methodology

❷ In generating process, the order of generated variable is not fixed but random.



$$P\left(\mathrm{F}_i^m = s\right) = \frac{1}{\binom{l}{|s|} \cdot l} \qquad P\left(\mathrm{F}_i^m = s\right) = \frac{t! \cdot (l-t)!}{l!} = \frac{1}{\binom{l}{t}}.$$

(left) Distribution of training process. (right) Distribution at generation step $0 \leq t \leq l$

▶ Since we encounter each t exactly once, this overall distribution is identical to the masking distribution encountered during training.

Methodology

How is the BERT utilized to generate tabular data?

- TabMT output $\hat{Y} \in \mathbb{R}^k$: prediction vector of each field.(masked field and unmasked field)
- The generated quantized value determines the final value $\hat{Y}$ based on the distribution of the cluster.

Experiment

DS	**TVAE**	**CTabGAN+**	**RealTab.**	**TabDDPM**	**TabMT**	Real
AB	0.433±0.008	0.467±0.004	0.504±0.011	**0.550±0.010**	0.535±0.004	0.556±0.004
AD	0.781±0.002	0.772±0.003	0.811±0.002	0.795±0.001	**0.814±0.001**	0.815±0.002
BU	0.864±0.005	0.884±0.005	0.928±0.003*	0.906±0.003	**0.908±0.002**	0.906±0.002
CA	0.752±0.001	0.525±0.004	0.808±0.003	0.836±0.002	**0.838±0.002**	0.857±0.001
CAR	0.717±0.001	0.733±0.001	-	0.737±0.001	**0.738±0.001**	0.738±0.001
CH	0.732±0.006	0.702±0.012	-	**0.755±0.006**	0.741±0.005	0.740±0.009
DI	0.714±0.039	0.734±0.020	0.732±0.027	0.740±0.020	**0.769±0.018**	0.785±0.013
FB	0.685±0.003	0.509±0.011	0.771±0.004	0.713±0.002	**0.798±0.002**	0.837±0.001
GE	0.434±0.006	0.406±0.009	-	0.597±0.006	**0.605±0.008**	0.636±0.007
HI	0.638±0.003	0.664±0.002	-	0.722±0.001	**0.727±0.001**	0.724±0.001
HO	0.493±0.006	0.504±0.005	-	**0.677±0.010**	0.619±0.004	0.662±0.003
IN	0.784±0.010	0.797±0.005	-	0.809±0.002	**0.811±0.003**	0.814±0.001
KI	0.824±0.003	0.444±0.014	-	0.833±0.014	**0.876±0.011**	0.907±0.002
MI	0.912±0.001	0.892±0.002	-	0.936±0.001	**0.938±0.001**	0.934±0.000
WI	0.501±0.012	0.798±0.021	-	**0.904±0.009**	0.881±0.009	0.898±0.006

Figure 3: ML Utility score and std across techniques.

- ML Utility score was obtained by using CatBoost trained on synthetic data to predict the original test data.
- The score was computed as the f1-score for classification datasets and as R^2 for regression datasets.
- TabMT performs better than all methods except TabDDPM.

Experiment - Privacy

Table 2: DCR score comparison between TabDDPM and TabMT. Corresponding MLE scores are in parentheses.

DS	TabDDPM	**TabMT**
AB	0.050(0.550)	**0.249**(0.533)
AD	0.104(0.795)	**1.01**(0.811)
BU	0.143(0.906)	**0.165**(0.908)
CA	0.041(0.836)	**0.117**(0.832)
CAR	0.012(0.737)	**0.041**(0.737)
CH	0.157(0.755)	**0.281**(0.758)
DI	0.204(0.740)	**0.243**(0.740)
FB	0.112(0.713)	**0.252**(0.787)

DS	TabDDPM	**TabMT**
GE	0.059(0.597)	**0.234**(0.599)
HI	0.449(0.722)	**0.483**(0.727)
HO	0.086(0.677)	**0.151**(0.607)
IN	0.041(0.809)	**0.061**(0.816)
KI	0.189(0.833)	**0.335**(0.868)
MI	0.022(0.936)	**0.026**(0.936)
WI	0.016(0.904)	**0.063**(0.881)

- DCR score: Average of the distance between synthetic data and original data.
- The tabular data generator has the trade-off between privacy and data quality, so the paper compared its privacy score only with TabDDPM, which had similar ML utility scores.
- TabMT performs better privacy score than TabDDPM.

Experiment - Missing data

DS	ML Utility score	Delta
AD	0.813	-0.001
KI	0.868	-0.008

Figure 4: ML Utility score of TabMT when training with 25% of values missing. Delta represents the difference in ML Utility score from training with no missing values.

- Other generators need to either drop rows with missing values or find ways to impute the missing values when a row contains missing data.
- Using TabMT's masking procedure, TabMT can inherently handle arbitrary missing data.

Conclusion

Conclusion

- Superior data quality
  - Our model achieves state-of-the-art generation quality.
- Missing data robustness
  - The quality remains consistent even in the presence of missing data.
- Privacy preserving generation
  - Our model achieves superior privacy.