Multi-label Tabular Synthetic Data Generation for Bundle Recommendation Problem

Aakash Swami
TCS Research, India

Tirumala V
TCS Research, India

2023 IEEE 2nd International Conference on Data, Decision and Systems (ICDDS) | 979-8-3503-1416-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDDS59137.2023.10434763

***Abstract*—Recommender systems are a great academic and industrial success in e-commerce and other areas. However, access to real-world historical interaction data for evaluating the recommender systems remains a challenge due to privacy, cost, and multiple issues. The state of the art techniques for generating synthetic data that is closest in relation to real world data, needs atleast a sample of real world data, which is also a problem especially when it comes to bundled data. We propose a novel way of generating multi-label synthetic data specifically for Bundle Recommender system problems, that does not need a sample of real world data but still is closest in relation to real world data. The proposed approach is hybrid as it uses a combination of process based and data based synthetic data generation methods. We demonstrate the feasibility of this approach by generating bundled data for Cosmetics product purchases. Also, synthetic data generated by the state-of-the-art method and our hybrid approach were compared, and it was found that the two sets of data are comparable in terms of statistical characteristics. The generated synthetic data can be used for evaluating bundle recommender systems for the industry.**

***Index Terms*—Synthetic data generation, Multi-label data, Hybrid approach, Bundle Recommendation, Cosmetics products**

I. Introduction

Many real-world Industrial applications recommend a bundle of items to users such as a travel package [1], a music playlist [2] so forth. Item bundling enhances user experience, raises average order value, lowers marketing and distribution expenses, or lowers inventory waste in these industrial applications. We refer to all of these applications in which a group of items are recommended as ”Bundle Recommendation” applications [3]. Datasets play a very important role in the development of the bundle recommendation system [4]. However, the availability of even a sample of real-world data remains a challenge due to privacy, cost, time taken, etc. Additionally, many a times concepts may be new and still not implemented, especially for bundling applications. This produces a need for the generation of synthetic bundle data for such applications. Several characteristics that relate to bundle data include 1. Each user is associated with multiple items. 2. Item-item interaction 3. Include both primary and product (auxiliary) items. For instance, if the primary item is a phone, the product item might be a case or a charger. The generation of primary item data can be compared to that of multi-class data, while the generation of product item data, which involves bundling of things, can be compared to that of multi-label data, where labels represent the items. The primary focus of this paper is the generation of multi-label product data for bundling.

Andre Goncalves et al. [7] presented in their related works a review of the state of art and progress of synthetic data generation methods. They broadly grouped them into process driven and data driven methods. Process-driven methods derive synthetic data from computational or mathematical models of an underlying physical process. Data-driven methods, on the other hand, derive synthetic data from generative models that have been trained on observed data. Liu et al. [8] developed a rule based synthetic data generator that takes the size of attributes and rules defined based on predefined criteria as input to generate samples. Sandro Mendonca et al. [11] developed a flexible synthetic data generator with lots of functionality included to generate data for different domains. Among the deep generative models available to generate synthetic data, Generative Adversarial Network (GAN) is a popular class of deep neural networks that generates synthetic data by learning distributions from the training data [5]- [6]. For multi-label data Jimena Torres et al. [10] presented a framework for generating multi-label data based on hypersphere and hypercube geometric shape strategies. It is necessary to include different aspects in multi-label synthetic data generation, such as feature distribution [12], feature generation [13], label distribution [15], label-label interaction [14], and feature label interaction [12], to improve its similarity to real-world data. In the literature, most of dataset generators produce single-label classification datasets [9] and the literature on multi-label synthetic data generators is very scant as multi-label classification is an emergent method [10]. Though there are multi-label synthetic data generators that are based on GAN, they also need a sample of real-world data.

To address this gap of generating synthetic data even when that sample data is also not available, generally rule-based synthetic data generation are used which is the next best in terms of its quality when compared with GAN based technique. We propose a novel way of generating multi-label synthetic data, that does not need a sample of real world data but still is closest in relation to real world data. The proposed approach uses a combination of rules and only a distribution of real world data that is available in survey based literature [23], [26] or other related literature. We show the feasibility by demonstrating this approach by generating multi-label synthetic data for Cosmetics products purchases and the

979-8-3503-1416-8/23/$31.00 ©2023 IEEE

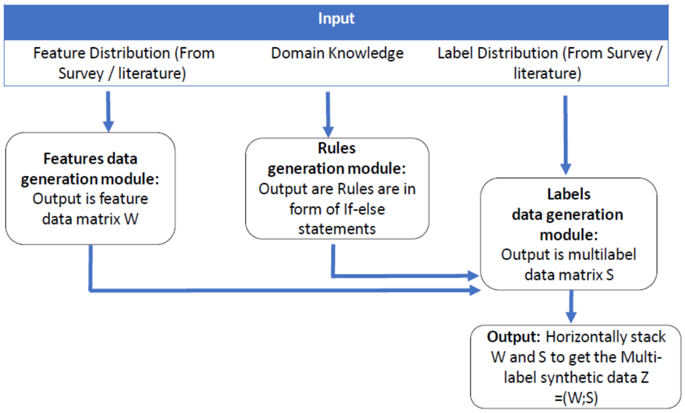

Input
Feature Distribution (From Survey / literature)
Domain Knowledge
Label Distribution (From Survey / literature)
Features data generation module: Output is feature data matrix W
Rules generation module: Output are Rules are in form of If-else statements
Labels data generation module: Output is multilabel data matrix S
Output: Horizontally stack W and S to get the Multi-label synthetic data Z =(W;S)

Fig. 1: Framework of the Multi-label synthetic data generator

results show that the distributions of generated synthetic data and real world data are almost similar. We also compared our approach to the state-of-the-art GAN model, and the results are shown in Table I. The rest of the paper is organized as follows: Concept and terminologies of datasets in section II, Problem formulation in section III, then methodology in section IV, demonstration of the feasibility of our proposed hybrid approach in section V, comparing the hybrid approach for multi-label synthetic data generation with state of art in section VI, followed by conclusion in section VII.

II. Concepts and terminologies of datasets

This section presents concepts and terminologies related to single and multi-label datasets as discussed in the literature to generate synthetic datasets.

- Feature distribution: The feature can take either standard or user-defined distribution [12]. Some well-known probability distributions are Normal, Log-Normal, Beta, Gamma, etc. which have a standard form.
- Label distribution: According to X.Geng [15] ”The label distribution covers a certain number of labels, representing the degree to which each label describes the instance ”.
- Feature generation: Feature generation is the process of creating new features from one or multiple features [19]. Dimensionality reduction and accuracy improvement are two goals of feature generation [20], [21]. When the goal is to enhance accuracy, the resulting feature space will almost certainly include more features than the original. Thus, Input space can be extended to include one or multiple features above the current feature.
- Feature-label interaction: The feature-label interaction is established based on rules [12].
- Label-label interaction: The label-label interaction describes how one label’s existence affects the presence of another. To improve the multi-label classification method, various researchers have attempted to model label-label interaction [14], [22].

III. Problem Formulation

Each instance in multi-label data consists of a feature vector and a labels vector associated with it. Let n denote the total number of samples or instances, d be the total number of features and m be the total number of labels. We denote an instance of multi-label synthetic data as z_i=(X_i,l_i), where i $\in$ 1,2,3...n. Here, $X_i \in R^d$ is a feature vector and $l_i \in R^m$ is label vector. In reality, getting access to real-world data remains difficult. However, distribution of real-world data is available in survey-based literatures for specific problems. A set of rules that defines the relationship between features and labels can also be defined for these problems based on domain knowledge. The goal is to create a multi-label synthetic data Z using input feature distributions, input label distributions A, and a set of rules. The input labels distributions is represented in the form A= $\{a_1, a_2, a_3.....a_m\}$ where a_i denote percentage of nonzero for particular label i. To formulate this problem, we first create feature data matrix W=$\{X_1, X_2, X_3.....X_n\}$, where feature vector X_i is formed by sampling data from input feature distributions. The relationship between features and labels is then defined by generating rules. The labels data matrix S = $\{l_1, l_2, l_3.....l_n\}$ is initialized with all 1 i.e $S_{i,j}$ =1 , where i $\in$ 1,2,3...n and j $\in$ 1,2,3...m. The labels data matrix S is then modified to match the input label distribution and the generated rules taking feature data matrix W as input. The multi-label synthetic data Z = (W;S) is formed by horizontally stacking feature data matrix W and modified label data Matrix S.

IV. Proposed Hybrid approach for Multi-label Synthetic data generation

The proposed hybrid approach involves generating multi-label synthetic data from both process based and data based methods. It includes the generation of multi-label data from labels distributions, where these distributions data is available from various survey data in the literature, for example [23], [24], [25] and the generation of feature/attribute data from features distributions, which are also easily available from the literature, for example [26]. These are part of distribution based methods from the data based methods of synthetic data generation [7]. The approach also includes rules to link the generated features and multi-labels data, which is a rule based method from the process based method of synthetic data generation [7]. The proposed approach is a hybrid because we are integrating both process based and data based methods of synthetic data generation. In the current work, it is assumed that each instance takes at least one label. Figure 1 shows the framework of the data generator. Features data generation module generates W; Rules generation module generates rules; Labels data generation module generates label data matrix S taking feature data matrix W and rules as input. The multi-label synthetic data Z = (W;S) is formed by horizontally stacking feature data matrix W and modified labels data Matrix S.

A. Features data generation module

The input features distributions are used to generate data for each feature. These input features distributions can follow either a standard or a customized probability distribution as per the standard literature. Feature vector X_i , where i $\in$ 1,2,3...n is created by randomly sampling data for the individual feature from the input features distributions. All of these feature vector's combines to form feature data matrix W.

B. Rules generation module

The rules are generated utilizing domain knowledge and criterion to establish the relationship between features and rules, as specified by Liu et al. [8]. The goal is to define a set of rules that can determine which feature vectors take label value as 0, for all the labels. Consider a scenario where there are three features (Feats_1,Feat_2, and Feat_3) and two labels (label_1 and label_2). A ruleset for label_1 taking value 0 would look like this: [[Feat_1=>9.0 and Feat_3=>21.0]] or [Feat_1=>9.0 and Feat_2=17.0-19.0 ...]]. For label 2, a ruleset can be defined similarly.

C. Labels data generation module

The flowchart to generate labels data are shown in Figure 2. The method iterates through all the multiple labels m and modifies label value till it matches the required input label distribution for each label and the rules applied. The label data matrix S is initialized with all ones. In multi-label data, each label will take either 0 or 1. Because the label data matrix S is initialized with 1 there is no need to include rules for which label takes value 1, only rules for which label takes value 0 are sufficient, which is what we have generated in the previous section. Then, the parameter p, which specifies individual label distribution at each run is calculated by finding the number of non-zero values in the label vector v_k. Here, $v_k \in R^n$ represents a column-wise label vector in S with k $\in$ 1, 2, 3, and m. Then 'While loop' is created with a condition to check if p is more than the required input label distribution value a_k. Here, a_k is the input label distribution for label k. Because all of the values in the label vectors v_k are set to 1, p is 100% at the start, and the criterion to enter the loop is always met at the start for each label. The algorithm then choses a random index q between 1 to n. Because each of the feature vector takes at least one label, the condition to count the number of non-zero values in label vector l_q denoted as c is considered before continuing to execute the further steps.

The parameter val is created, which takes value '0' $h1$ % of time and '1' $h2$ % of time. For example if $h1$ is taken as 90 and $h2$ is taken as 10. This means that 90 % of the time condition to establish the rule is implemented and 10 % cases where the rule is not followed are taken. This val parameter realizes two purposes: First, labels for which the rules are not followed fully but to the extent that distribution is maintained it makes sure that the loop does not enter the infinity loop. Second, it adds noise to the data. The noise refers to a data sample where the rule is not followed. If the parameter val has a value of zero, the feature vector at index q, X_q, is checked to see if it satisfies the ruleset for that specific label. If rule is satisfied then the value of the label vectore l_q for that particular label k is modified to 0 and if the value of parameter val is one then also the value of the label vectore l_q for that particular label k is modified to 0. In the end, the value of p is updated by finding the number of non-zero values in the modified label vector v_k. The loop iterates till p gets below the required input distribution value for each label. Thus, using the proposed approach we generated the Modified S matrix that has label distribution based on the rules applied and the input label distribution.

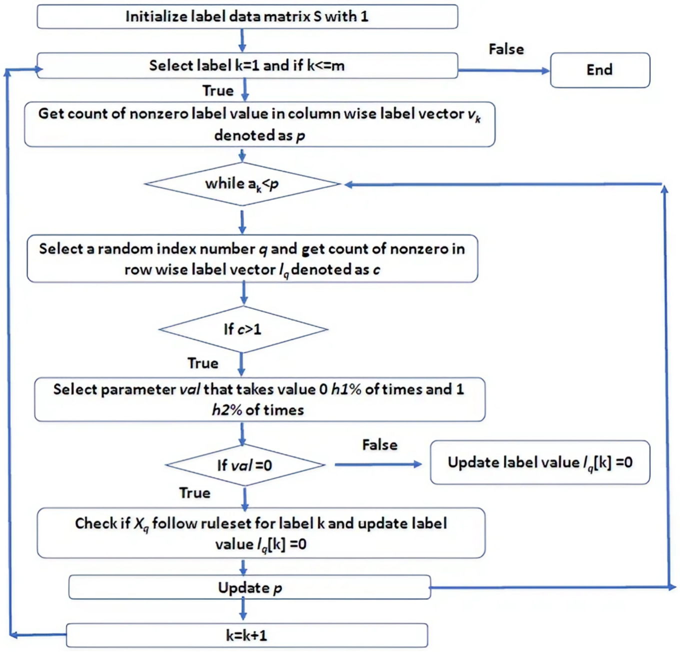
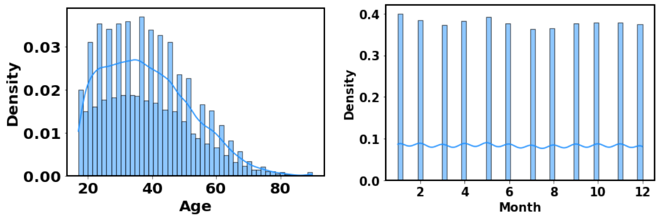

Initialize label data matrix S with 1
Select label k=1 and if k<=m
False
End
True
Get count of nonzero label value in column wise label vector v_k denoted as p
while a_k<p
Select a random index number q and get count of nonzero in row wise label vector l_q denoted as c
If c>1
True
Select parameter val that takes value 0 $h1$% of times and 1 $h2$% of times
False
If val=0
Update label value l_q[k] =0
True
Check if X_q follow ruleset for label k and update label value l_q[k] =0
Update p
k=k+1

Fig. 2: Flowchart depicting step by step process of generating labels data

The multi-label synthetic data Z = (W;S) is now formed by horizontally stacking feature data matrix W and modified labels data Matrix S.

V. Generating Cosmetics products' synthetic data using our hybrid approach

The proposed hybrid approach for data generation is demonstrated by generating data on the user preference for Cosmetics products based on user features. We consider two independent user features of the Cosmetics products' data and these are 'Age' and 'Month'. The input feature distribution for these two independent user features is shown in Figure 3. We obtained the distribution of age features from Demographic data available on Kaggle [26]. The month follows a uniform distribution. As a part of the demonstration of our hybrid approach, product distribution as shown in Figure 5(a) is used as input label distribution. We have considered three labels i.e., 'Sunscreen product', 'Anti aging product' and 'Moisturizer product'. Three rules that are used in the data generation are as follows: The first rule is applied to 'Sunscreen product'. The



(a) Age (b) Month

Fig. 3: Input feature distribution

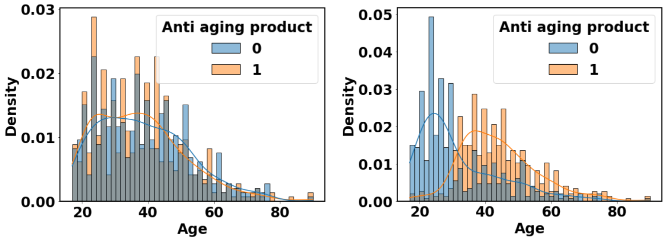
rule says that if 'Month' is May, June, or July then 'Sunscreen product' =1 else 'Sunscreen product' =0. The second rule is applied to 'Anti aging product', the rule says that if 'age' >= 30 then 'Anti aging product' =1 else 'Anti aging product' =0. The third rule is applied to 'Moisturizer product'. The rule says that if the 'Month' is May, June, or July then 'Moisturizer product' =1 else 'Moisturizer product' =0. Thus, using these above inputs, the data is generated for customers who selected at least one product.

The generated data is examined to see if the distribution has been maintained and whether rules have been applied. Because the data is generated for validation purposes, only around 5000 samples are generated. h1 is taken as 90% and h2 is taken as 10%. Figure 5(a) compares the input label distributions with the label distribution in the generated data. The two distributions follow closely with each other with an error of less than 0.1%. To measure the quality of feature data we use the KL divergence metric also used by [7] to find the similarity between the statistical property of the original feature data and the feature data generated by the algorithm. The KL test is computed over the pair of original and generated probability mass functions for a given feature. The value is close to zero when both distributions are similar. The larger value of KL divergence indicates a larger discrepancy. The KL divergence of two PMFs, P and Q for a given feature is computed as follows:

$$KL(P||Q) = \sum_{i=1}^{n} P(x) \ln(\frac{P(x)}{Q(x)}) \tag{1}$$

The KL divergence value for both features is less than 0.01 and hence it is concluded that the feature distributions of generated data are similar to input feature distributions data. The result of the application of the second rule is shown in Figure 4. As seen in Figure 4(b) the distribution pattern shifts to follow this rule. The results of the application of the first and third rule is intuitively same as for second rule.

The density of the label vector (1,0,1) displayed in the box in Figure 5(b) increases after the application of the rules. This is because the same rule is applied to both 'Sunscreen product' and 'Moisturizer product'. The same rule applied to two labels increases the likelihood of them existing together and hence label-label interaction in the data is established as per the rule applied.



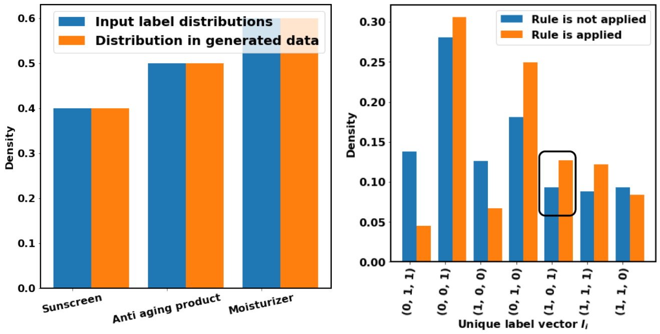
(a) Distribution when rule is not applied (b) Distribution when rule is applied

Fig. 4: Implementation of Rule 1 showing the distribution of 'age' compared across label 'Anti aging product'

The concept and terminologies of dataset discussed in section II are taken into consideration in our approach in the following way: First, because feature data is generated by random sampling from an input feature distribution, the feature distribution is considered. Second, the proposed approach generates multiple label data matching input label distribution thus considering label distribution. Third, the algorithm provides flexibility to extend input space to include new features thus considering feature generation. Fourth, the rules are used to bring feature label interaction. The rules change the distribution of labels to make them conform to the rules applied. Thus, considering feature-label interaction. Last, the use of rules has an impact on the possibility of labels coexisting, resulting in label-label interaction. To illustrate consider the following two rules. The first rule states that label 1 takes value 1 if feature 1 $>$ 4, and the second rule states that label 2 takes value 1 if feature 1 $>$ 4. Thus, the likelihood of label 1 and label 2 existing together is influenced by the rule.

VI. Comparison of hybrid approach and state of the art

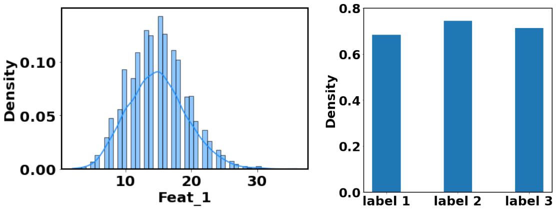
Even though the proposed approach is unique as it does not require a sample of real-world data. But for the sake of comparison with the state-of-the-art method which need sample data we made an assumption that the sample test data is available. We used the CTGAN (Conditional Tabular GAN) [17] data generation model for comparison. The CTGAN method uses GAN to model the distribution of tabular real data and then sample rows of synthetic data from the distribution. To start with, a small sample of test data is first created. We used datasets produced using standard multi-label classification libraries available under Scikit-learn for the generation of test data. Then, we learn this test data using CTGAN and produced a sample of synthetic data. The proposed hybrid multi-label synthetic data generator is also used to produce a sample of synthetic data. The TableEvaluator library [16] is then used to assess how closely the synthetic dataset resembles the test data. We have considered 3 features and 3 labels. There are 4,000 test data samples that were taken. The generated synthetic data contains around 10,000 samples. Our hybrid data generation approach requires the distribution of features, the distribution of labels, and a set



(a)

(b)

Fig. 5: a) Comparison between input label distribution and label distribution in the generated data b) Comparison of unique label l_i present in the data generated, with and without the application of the rule



(a)

(b)

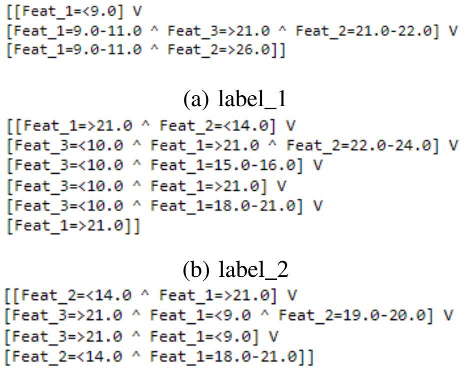
Fig. 6: a) Distribution of feature 1 b) Input label distribution

of rules as input. The distribution of features and labels are extracted from the test data. The input feature distribution for one of the feature is shown in Figure 6(a) and input label distribution is shown in Figure 6(b). The rules are extracted using the algorithm RIPPER (Repeated Incremental Pruning to Induce Error Reduction) [18]. Our aim with ruleset extraction using RIPPER is to create a set of rules that can identify which feature vectors take label as 0 for all the 3 labels. The rule extracted from the original data will be large. But to be more realistic we consider a subset of full rule because we won't be able to capture every rule in the data based on our domain expertise. The rules extracted using RIPPER, where labels takes value 0 is shown in Figure 7, where "V" stands for "or" and " ^ " for "and."

TABLE I: All metric and corresponding value

Metric	CTGAN	Hybrid
Basic statistical evaluation	0.98	0.99
Correlation correlation	0.96	0.76
Correlation distance : RMSE	0.046	0.11
Correlation distance : MAE	0.034	0.083
Mean correlation between fake and real	0.914	0.99
Row distance	(0.29,0.29)	(0.26,0.26)

The metrics used to compare the data are those that Bauke et al. [16] define in his thesis. The specifics of various metrics are briefly described below.


```
[[Feat_1=<9.0] V
[Feat_1=9.0-11.0 ^ Feat_3=>21.0 ^ Feat_2=21.0-22.0] V
[Feat_1=9.0-11.0 ^ Feat_2=>26.0]]
```

(a) label_1

```
[[Feat_1=>21.0 ^ Feat_2=<14.0] V
[Feat_3=<10.0 ^ Feat_1=>21.0 ^ Feat_2=22.0-24.0] V
[Feat_3=<10.0 ^ Feat_1=15.0-16.0] V
[Feat_3=<10.0 ^ Feat_1=>21.0] V
[Feat_3=<10.0 ^ Feat_1=18.0-21.0] V
[Feat_1=>21.0]]
```

(b) label_2

```
[[Feat_2=<14.0 ^ Feat_1=>21.0] V
[Feat_3=>21.0 ^ Feat_1=<9.0 ^ Feat_2=19.0-20.0] V
[Feat_3=>21.0 ^ Feat_1=<9.0] V
[Feat_2=<14.0 ^ Feat_1=18.0-21.0]]
```

(c) label_3

Fig. 7: The ruleset for labels taking value 0

TABLE II: ML Accuracy and F1 score

Model	Real Data Acc;F1	CTGAN Acc;F1	Hybrid Acc;F1
RF	0.34;0.80	0.32;0.75	0.37;0.81
KNN	0.35;0.80	0.31;0.76	0.38;0.82
ANN	0.38;0.82	0.35;0.78	0.40;0.85

Basic statistical evaluation: Calculate the correlation coefficient between the basic properties of real and fake data like mean, median, standard deviation, and variance using Spearman's Rho.

Correlation correlation: Find the correlation coefficient between the association matrix of real and fake data.

Correlation distance: Compute the distance between association matrices using certain metrics like RMSE or MAE.

Row distance: Compute Mean and standard deviation between fake and real data.

Mean correlation between fake and real/ Column correlation : Column correlation between real and fake data

The result of the comparison between CTGAN and Hybrid data generator is shown in Table I. The two generator compares well in terms of statistical characteristics. To assess the effectiveness of generated synthetic data for application in machine learning, we also tested it on the ML algorithms RF (Random Forest), KNN (K-Nearest Neighbour), and ANN. Table II shows the result on metric 'Accuracy' and 'F1-score'. The values are comparable.

VII. Conclusion

In this work, we proposed a hybrid approach for multi-label synthetic data generation, which uses a combination of process driven and data driven synthetic data generation methods. Through evaluation metrics and considering various aspects of multi-label synthetic data, our hybrid approach showed impressive results for both features and labels of multi-label dataset. Due to various difficulties in getting access even for a sample of real world data, we expect this hybrid approach to make a good contribution to the development and evaluation of bundle recommendation algorithms for various industrial applications. The hybrid approach also provides industry analyst flexibility to include specific needs or certain conditions that may not be found in the real data to simulate changing market/customer behavior.

References

[1] Ge, Yong, Hui Xiong, Alexander Tuzhilin, and Qi Liu. ”Cost-aware collaborative filtering for travel tour recommendations.” ACM Transactions on Information Systems (TOIS) 32, no. 1 (2014): 1-31.

[2] Cao, Da, Liqiang Nie, Xiangnan He, Xiaochi Wei, Shunzhi Zhu, and Tat-Seng Chua. ”Embedding factorization models for jointly recommending items and user generated lists.” In Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, pp. 585-594. 2017.

[3] Zhu, Tao, Patrick Harrington, Junjun Li, and Lei Tang. ”Bundle recommendation in ecommerce.” In Proceedings of the 37th international ACM SIGIR conference on Research and development in information retrieval, pp. 657-666. 2014.

[4] He, Xiangnan, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. ”Neural collaborative filtering.” In Proceedings of the 26th international conference on world wide web, pp. 173-182. 2017.

[5] Radford, Alec, Luke Metz, and Soumith Chintala. ”Unsupervised representation learning with deep convolutional generative adversarial networks.” arXiv preprint arXiv:1511.06434 (2015).

[6] Goodfellow, Ian. ”Nips 2016 tutorial: Generative adversarial networks.” arXiv preprint arXiv:1701.00160 (2016).

[7] Goncalves, Andre, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Ana Paula Sales. ”Generation and evaluation of synthetic patient data.” BMC medical research methodology 20, no. 1 (2020): 1-40.

[8] Liu, Runzong, Bin Fang, Yuan Yan Tang, and Patrick PK Chan. ”Synthetic data generator for classification rules learning.” In 2016 7th International Conference on Cloud Computing and Big Data (CCBD), pp. 357-361. IEEE, 2016.

[9] Abufadda, Mohammad, and Khalid Mansour. ”A survey of synthetic data generation for machine learning.” In 2021 22nd International Arab Conference on Information Technology (ACIT), pp. 1-7. IEEE, 2021.

[10] Tomás, Jimena Torres, Newton Spolaôr, Everton Alvares Cherman, and Maria Carolina Monard. ”A framework to generate synthetic multi-label datasets.” Electronic Notes in Theoretical Computer Science 302 (2014): 155-176.

[11] Mendonca, Sandro De Paula, Yvan Pereira Dos Santos Brito, Carlos Gustavo Resque Dos Santos, Rodrigo Do Amor Divino Lima, Tiago Davi Oliveira De Araujo, and Bianchi Serique Meiguins. ”Synthetic datasets generator for testing information visualization and machine learning techniques and tools.” IEEE Access 8 (2020): 82917-82928.

[12] Seidlová, R., J. Poživil, J. Seidl, and L. Malecl. ”Synthetic data generator for testing of classification rule algorithms.” Neural Network World 27, no. 2 (2017): 215.

[13] Islamaj, Rezarta, Lise Getoor, and W. John Wilbur. ”A feature generation algorithm for sequences with application to splice-site prediction.” In Knowledge Discovery in Databases: PKDD 2006: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings 10, pp. 553-560. Springer Berlin Heidelberg, 2006.

[14] Kasinikota, Anusha, P. Balamurugan, and Shirish Shevade. ”Modeling label interactions in multi-label classification: a multi-structure SVM perspective.” In Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part I 22, pp. 43-55. Springer International Publishing, 2018.

[15] Geng, Xin. ”Label distribution learning.” IEEE Transactions on Knowledge and Data Engineering 28, no. 7 (2016): 1734-1748.

[16] Brenninkmeijer, Bauke, A. de Vries, E. Marchiori, and Youri Hille. ”On the generation and evaluation of tabular data using GANs.” PhD diss., Radboud University, 2019.

[17] Xu, Lei, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. ”Modeling tabular data using conditional gan.” Advances in Neural Information Processing Systems 32 (2019).

[18] Cohen, William W. ”Fast effective rule induction.” In Machine learning proceedings 1995, pp. 115-123. Morgan Kaufmann, 1995.

[19] S. Van den Bosch ”Automatic feature generation and selection in predictive analytics solutions”. Master's thesis, Faculty of Science, Radboud University. 2017;3(1):3-1.

[20] ML.Raymer, WF. Punch, ED. Goodman, LA. Kuhn, AK. Jain. ”Dimensionality reduction using genetic algorithms”. IEEE transactions on evolutionary computation. 2000 Jul;4(2):164-71.

[21] R.Islamaj, L.Getoor, WJ.Wilbur ”A feature generation algorithm for sequences with application to splice-site prediction”. InEuropean Conference on Principles of Data Mining and Knowledge Discovery Springer, Berlin, Heidelberg. 2006 Sep 18 (pp. 553-560).

[22] Z.Younes, F.Abdallah, T.Denoeux, H.Snoussi ”A dependent multilabel classification method derived from the k-nearest neighbor rule”. EURASIP Journal on Advances in Signal Processing. 2011 Dec;2011:1-4.

[23] ”Cosmetic industry in India size & share analysis - growth trends & forecasts (2023 - 2028)” mordorintelligence.com https://www.mordorintelligence.com/industry-reports/india-cosmetics-products-market-industry (accessed Aug. 8, 2023)

[24] ”14 Trends Changing The Face Of The Beauty Industry In 2021” cbinsights.com https://www.cbinsights.com/research/report/beauty-trends-2021/ (accessed Aug. 8, 2023)

[25] ”Cosmetic Market in Japan: Key Research Findings 2017” yanoresearch.com https://www.yanoresearch.com/en/press-release/show/press_id/1760 (accessed Aug. 8, 2023)

[26] ”Demographic data” kaggle.com https://www.kaggle.com/code/advaitchavan/demographic-data/input (accessed July 15,2023)