Synthetic Dataset Generation with Itemset-Based Generative Models

Christian Lezcano and Marta Arias

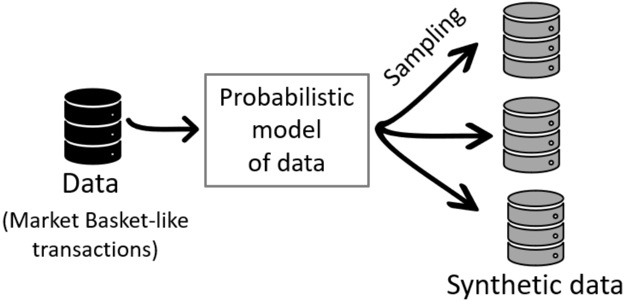
Universitat Politècnica de Catalunya, Barcelona, Spain

October 28, 2019

Synthetic data applications

- Provide data when in short supply.
- Synthetic data (based on statistical models) allows to choose the data volume as well as to generate as many copies as desired.
- Protect the confidentiality of real data (e.g., in software testing)

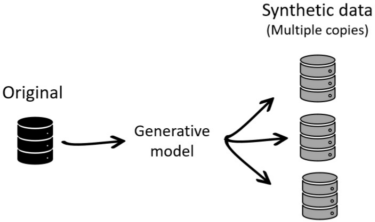
Data generation approach



Contributions

The contributions of this work are:

1. three synthetic transactional dataset generators using generative models based on itemsets.
2. quality evaluation of generated datasets based on various criteria in order to know the strengths and weaknesses of each model.



Dataset representation

Transaction ID	Items bought from a supermarket				
Customer 1	egg	bread	milk	pizza	
Customer 2	bread	beer	diapers	milk	butter
Customer 3	diapers	milk	butter		
Customer 4	egg	bread	beer	diapers	milk
Customer 5	beer	diapers	milk	butter	pizza

Market Basket transactions

X = {beer, diapers} example of frequent itemset ("pattern")

The support of an itemset $sup(X)$ is defined as the number of transactions that contain X.

$$sup(X) = |\{t \in D \mid X \subseteq t\}|$$

X is considered frequent if its support is greater than or equal to a minimum support *minsup* defined by the user, i.e., $sup(X) \geq minsup$.

IGM model

Now, we need a probabilistic model of a representative set of patterns.

IGM model[1] only models a specific pattern X and its power set 2^X:

$$T(X) = \begin{cases} X & \text{w.p.} \quad \theta \\ X' \subset X & \text{w.p.} \quad \left(\frac{1-\theta}{2^{|X|}-1}\right) \end{cases}$$

$$\mathrm{T}(\bar{\mathrm{X}}) = \mathrm{X}'' \subseteq \bar{X} \quad \text{w.p.} \quad \left(\frac{1}{2^{|I|-|X|}}\right)$$

IGM assumes a transaction is generated with only <u>one pattern</u> $T(X)$ and noise $T(\bar{X})$.

New transaction $T \leftarrow T(X) \cup T(\bar{X})$

[1] Laxman et al. (2007)

IGM-based generator

Algorithm 1: IGM-based generator

Generate dataset $(D_{ori}, minsup)$
    $D_{syn} \leftarrow \emptyset$
    $fi \leftarrow$ Mine frequent itemsets $(D_{ori}, minsup)$
    $fi^* \leftarrow$ Filter frequent itemsets (fi)
    while $|D_{syn}| < |D_{ori}|$ **do**
        $D_{syn} \leftarrow D_{syn} \cup$ Generate transaction(fi^*)
    return D_{syn}

Generate transaction (fi^*)
    $T \leftarrow \emptyset$
    $X \leftarrow$ Sample itemset from fi^*

$$T(X) = \begin{cases} X \\ X' \subset X & \text{w.p.} \left(\frac{1-\theta}{2^{|X|}-1}\right) \end{cases}$$

$$T(\bar{X}) = X'' \subseteq \bar{X} \quad \text{w.p.} \left(\frac{1}{2^{|I|-|X|}}\right)$$

$$T \leftarrow T(X) \cup T(\bar{X})$$

New transaction T

    return T

IIM model

IIM model[2] infers itemsets that represent best the data using structural EM.

IIM allows to obtain a probabilistic distribution over a set of patterns.

$$Y_x \sim \text{Bernoulli}(p_x)$$

$$\text{New transaction } T = \bigcup_{X \mid Y_x = 1} X$$

[2] Fowkes and Sutton (2016)

IIM-based generator

Algorithm 2: IIM-based generator

Generate database *(D_{ori})*
- $D_{syn} \leftarrow \emptyset$
- $II, p \leftarrow$ Learn IIM model (D_{ori})
- **while** $|D_{syn}| < |D_{ori}|$ **do**
  - $D_{syn} \leftarrow D_{syn} +$ Generate transaction(II, p)
- **return** D_{syn}

Generate transaction *(II, p)*
- $T \leftarrow \emptyset$
- **foreach** *itemset* X *in* II **do**
  - $Y_x \sim$ Bernoulli(p_x)
  - $T = \bigcup_{X \mid Y_x = \mathbf{1}} X$
- **return** T

} New transaction T

LDA model[3]

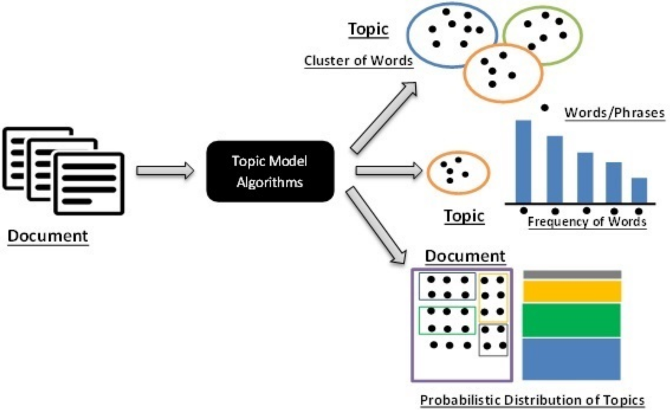
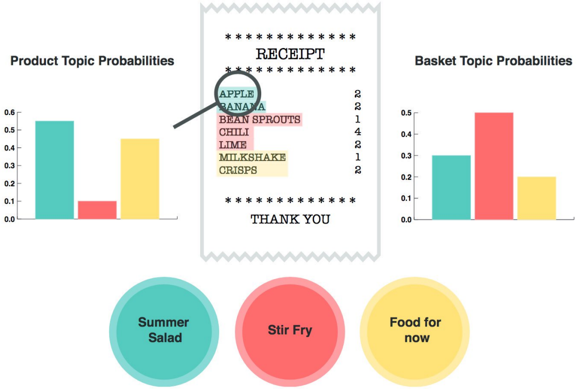


Image credit: Christine Doig

[3] Blei et al. (2003)

LDA model



One topic represents a specific pattern

Image taken from Hornsby et al. (2019)

LDA-based generator

Algorithm 3: LDA-based generator

Generate dataset (D_{ori}, K)
    $D_{syn} \leftarrow \emptyset$
    $\theta_i,\ \varphi_t \leftarrow$ Learn LDA model $(D_{ori},\ K)$
    while $|D_{syn}| < |D_{ori}|$ **do**
        $T \leftarrow \emptyset$
        while $|T| < N_i$ **do**
            $t \leftarrow$ Sample topic from θ_i
            $w_j \leftarrow$ Sample word from φ_t
            $T \leftarrow T \cup w_j$
        } New transaction T
        $D_{syn} \leftarrow D_{syn} + T$
    return D_{syn}

1. For each document d_i, $1 \leq i \leq M$, choose its own probability distribution of topics θ_i from a Dirichlet distribution with parameter α.
2. For each topic t, $1 \leq t \leq K$, choose its probability distribution of words φ_t from a Dirichlet distribution with parameter β. The number of topics K is defined by the user.
3. For each word in a document, that is, for each word w_j in a document d_i, first (a) select a topic t from θ_i and, then (b) select a word w_j from φ_t.

List of datasets generated

	Dataset	Model	Levels of support (%)	Generated datasets
1.	forests	LDA	$\langle 60, 70, 80, 90 \rangle$	$\langle for_{LDA}60, for_{LDA}70, for_{LDA}80, for_{LDA}90 \rangle$
2.	forests	IGM	$\langle 70, 80, 90 \rangle$	$\langle for_{IGM}70, for_{IGM}80, for_{IGM}90 \rangle$
3.	forests	IIM		$\langle for_{IIM} \rangle$
4.	bogPlants	LDA	$\langle 10, 20, 30, 40, 50, 60 \rangle$	$\langle bog_{LDA}10, bog_{LDA}20, bog_{LDA}30, \ldots, bog_{LDA}60 \rangle$
5.	bogPlants	IGM	$\langle 10, 20, 30, 40, 50, 60 \rangle$	$\langle bog_{IGM}10, bog_{IGM}20, bog_{IGM}30, \ldots, bog_{IGM}60 \rangle$
6.	bogPlants	IIM		$\langle bog_{IIM} \rangle$

Benchmarking datasets forest and bogPlants taken from W. Hamalainen[4]

We generate 10 datasets for each synthetic dataset representation, e.g., $for_{LDA}60$ actually represents a set of 10 generated databases.

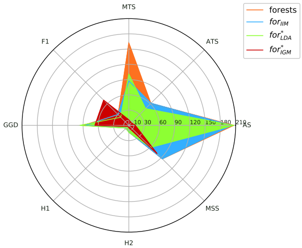
[4] http://www.cs.uef.fi/~whamalai/datasets.html (accessed September 1, 2017)

Characteristic metrics

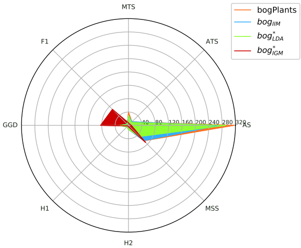
	Dataset	DS	AS	ATS	MTS	F1 (%)	GGD (%)	H1	H2	MSS (%)
1.	forests	246	206.00	61.26	162.00	29.74	89.88	7.07	13.24	93.09
2.	for^*_{LDA}	246	205.70	46.45	100.85	22.58	95.52	7.41	13.84	61.04
3.	for^*_{IGM}	246	12.67	7.07	10.93	69.98	66.67	2.74	4.75	78.46
4.	for_{IIM}	246	202.60	61.59	87.40	30.40	85.32	7.06	13.13	93.09
5.	bogPlants	377	315.00	14.65	39.00	4.65	16.57	6.56	11.56	65.25
6.	bog^*_{LDA}	377	290.52	12.49	29.55	4.32	25.19	6.87	12.22	47.02
7.	bog^*_{IGM}	377	8.67	4.86	7.77	67.75	83.33	2.49	3.92	72.46
8.	bog_{IIM}	377	270.80	15.03	28.90	5.55	24.73	6.50	11.77	64.85

Each value represents the average between all the databases generated by each benchmarking dataset and model.

Evaluation on characteristics: IIM is the best.

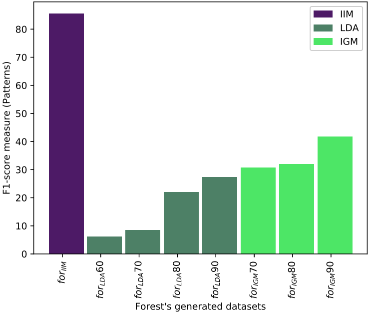


(a) forest

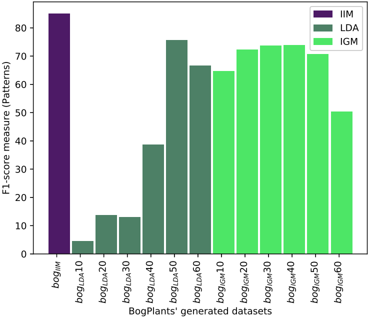


(b) bogPlants

Preservation of frequent itemsets: IIM is the best.



(c) forest



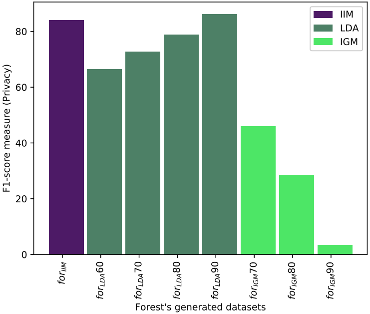
(d) bogPlants

precision $p_X(Y) = \frac{|X \cap Y|}{|Y|}$; $p(FI_{syn}) = \frac{1}{|FI_{syn}|} \sum_{Y \in FI_{syn}} \max_{X \in FI_{ori}} \{p_X(Y)\}$

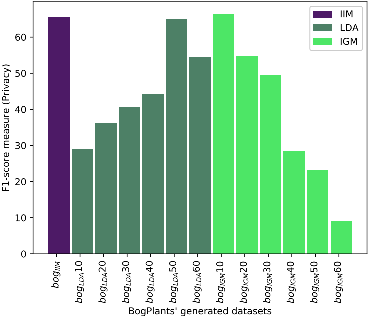
recall $r_X(Y) = \frac{|X \cap Y|}{|X|}$; $r(FI_{syn}) = \frac{1}{|FI_{ori}|} \sum_{X \in FI_{ori}} \max_{Y \in FI_{syn}} \{r_X(Y)\}$

F_1-score $= \frac{2 * precision * recall}{precision + recall}$

Evaluation on privacy: IGM is the best.



(e) forest



(f) bogPlants

precision $p(D_{syn}) = \frac{1}{|D_{syn}|} \sum_{Y \in D_{syn}} \max_{X \in D_{ori}} \{p_X(Y)\}$

recall $r(D_{syn}) = \frac{1}{|D_{ori}|} \sum_{X \in D_{ori}} \max_{Y \in D_{syn}} \{r_X(Y)\}$.

Runtime evaluation

Table 1: Learning fase runtime in seconds.

Model	forest	bogPlants
IGM	0.02	0.03
IIM	546.29	102.24
LDA	1654.79	228.53

Table 2: Generation fase runtime in seconds.

Model	forest	bogPlants
IIM	0.43	0.62
LDA	6.50	1.98
IGM	400.43	119.89

Conclusion and future work

1. We presented in this work several types of generators to create synthetic transactional datasets which are based on generative models.
2. It was observed experimentally that each one possesses specific abilities according to several criteria.
3. As future work, we plan on using a larger set of benchmarking datasets, and we are in the process of introducing new generator algorithms

References

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Fowkes, J. M. and Sutton, C. A. (2016). A bayesian network model for interesting itemsets. In Frasconi, P., Landwehr, N., Manco, G., and Vreeken, J., editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II*, volume 9852 of *Lecture Notes in Computer Science*, pages 410–425. Springer.

Hornsby, A. N., Evans, T., Riefer, P. S., Prior, R., and Love, B. C. (2019). Conceptual organization is revealed by consumer activity patterns. *Computational Brain & Behavior*.

Laxman, S., Naldurg, P., Sripada, R., and Venkatesan, R. (2007). Connections between mining frequent itemsets and learning generative models. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*, pages 571–576. IEEE Computer Society.

Thank you for your attention