ALICE and the Caterpillar: A More Descriptive Null Model for Assessing Data Mining Results

Extended Version

Giulia Preti
CentAI
Turin, Italy
giulia.preti@centai.eu

Gianmarco De Francisci Morales
CentAI
Turin, Italy
gdfm@acm.org

Matteo Riondato
Amherst College
Amherst, MA, USA
mriondato@amherst.edu

"One side will make you grow taller, and the other side will make you grow shorter." — The Caterpillar, *Alice in Wonderland*

***Abstract*—We introduce a novel null model for assessing the results obtained by analyzing an observed transactional dataset (e.g., significant frequent itemsets) using statistical hypothesis testing. Our null model maintains more properties of the observed dataset than existing models. Specifically, we preserve the Bipartite Joint Degree Matrix of the bipartite graph corresponding to the dataset, which ensures that the number of caterpillars, i.e., paths of length three, is preserved, in addition to the item supports and the transaction lengths, which are the properties considered by previous works. We describe ALICE, a suite of two Markov-Chain Monte-Carlo algorithms for sampling datasets from our null model, based on a carefully defined set of states and efficient operations to move between them. The results of our experimental evaluation show that ALICE mixes fast and scales well, and that our null model finds different significant results than ones previously considered in the literature.**

***Index Terms*—Significant Pattern Mining, Swap Randomization, Hypothesis Testing, Markov Chain Monte Carlo Methods**

I. INTRODUCTION

Binary transactional datasets are the object of study in several areas, from marketing to network analysis, to genomics, where, for example, transactions represent individuals and the items in a transaction represent their gene mutations. Many fundamental data mining tasks can be defined on them, such as frequent itemset mining, clustering, and anomaly detection.

The goal of knowledge discovery from a dataset is not simply to analyze the dataset, but to obtain *new understanding* of the stochastic, often noisy, *process that generated the dataset*. Such novel insights can only be obtained by subjecting the results of the analysis to a rigorous validation, which allows to separate those results that give new information about the process from those that are due to the randomness of the process itself. This kind of validation is actually necessary in many scientific fields, for example in microbiology and genomics, when the observed dataset represents individuals with their gene mutations, or protein interactions [1, 2, 3].

The *statistical hypothesis testing* framework [4, Ch. 10] is a most rigorous validation process for the results obtained from an observed dataset. Hypotheses about the results are formulated, and then tested by comparing them (or appropriate statistics about them) to the distribution of the same results over the *null model*, i.e., a set of datasets enriched with a user-specified probability distribution (see Sect. III-A), that contains all and only the datasets that preserve a user-specified subset of the properties of the observed dataset (e.g., the size, or some cumulative statistics). The testing of hypotheses requires, in *resampling-based methods* [5], to be able to efficiently draw multiple datasets from the null model. These samples are then used to obtain an approximation of the distribution of results from the null model, to which the actually observed results are compared. When the probability of obtaining results as or more extreme than those observed is low, the observed results are deemed *statistically significant*, i.e., they give previously unknown information about the data generating process.

Informally, the properties preserved by the null model, and the sampling distribution, capture the existing or assumed knowledge about the process that generated the observed dataset. Testing the hypothesis can be understood as trying to ascertain whether the observed results can be explained by the existing knowledge. The choice of the null model must be made by the user, based on their domain knowledge, and should be deliberate. Null models that capture more properties of the observed dataset are usually more descriptive and therefore to be preferred. The challenge in using such models is the need for efficient computational procedures to draw datasets from the null model according to the user-specified distribution, as many such sampled datasets are necessary to test complex or multiple hypotheses.

Contributions

We study the problem of assessing results obtained from an observed transactional dataset (e.g., frequent itemsets and itemset distributions) by performing statistical hypothesis tests via resampling methods from a descriptive null model. Specifically, our contributions are the following.

- We introduce a novel null model (Sect. IV) that preserves additional properties of the observed dataset than those preserved by existing null models [6]. Specifically, all datasets in our null model have the same *Bipartite Joint Degree Matrix (BJDM)* of the bipartite graph corresponding to the observed dataset (Sect. IV-A and IV-B). Maintaining the BJDM ensures that, in addition to dataset

size, transaction lengths, and item supports, also the number of *caterpillars* in the observed dataset is preserved (Lemma 1), which captures additional "structure" of the dataset. We also explain why more natural properties, such as the supports of itemsets of length two, are not as informative as one may think.
- We present ALICE,[1] a suite of two Markov-Chain-Monte-Carlo algorithms for sampling datasets from our null model according to a user-specified distribution. ALICE-A (Sect. V-A) is based on *Restricted Swap Operations (RSOs)* on biadjacency matrices, which preserve the BJDM. Our contributions include a sampling algorithm to draw such RSOs much more efficiently than with the natural rejection sampling approach. Our second algorithm, ALICE-B, (Sect. V-B) adapts the CURVEBALL approach [7] to RSOs, to essentially perform multiple RSOs at every step, thus leading to faster mixing.
- The results of our experimental evaluation show that ALICE mixes fast, it is scalable as the dataset grows, and that our new null model differs from previous ones, as it marks different results as significant.

II. RELATED WORK

The need for statistically validating results from transactional datasets was understood immediately after the first efficient algorithm for obtaining these results was introduced [8, 9]. A long line of works also studies how to filter out uninteresting patterns, or directly mine *interesting* ones [10]. This direction is orthogonal to the study of the *statistical validity* of the results, which is our focus.

Most work has been on mining *significant frequent itemsets*, tiles, or association rules [11, 12, 13]. The survey by Hämäläinen and Webb [14] presents many of these works in depth. The most relevant to ours are those by Gionis et al. [6] and Hanhijärvi [15], who present resampling methods for drawing transactional datasets from a null model which preserves the number of transactions, the transaction lengths, and the item supports as in an observed dataset. These approaches, like ours, can be used for testing any result from transactional datasets, not just for significant pattern mining. We present a null model that is more descriptive than the ones studied in these works, because it preserves additional properties of the observed dataset. Bie [16] proposes a method to *uniformly* sample datasets from a null model that preserves, *in expectation*, the same constraints. While it can partially be extended to preserve the constraints exactly, it cannot be used to sample according to any user-specified distribution, which we believe to be a fundamental ingredient of the null model, as it includes already available knowledge of the data generating process *in addition to* the constraints.

Beyond binary transactional datasets, resampling methods for assessing data mining results have been proposed for sequence datasets [17, 18, 19] graphs [20, 21, 22, 23], real-valued and mixed-valued matrices [24], and database tables [25]. None of these works proposes a null model similar to the one we introduce, nor presents similar sampling algorithms. Our approach can be a starting point to develop more descriptive null models for these richer types of data.

ALICE, our algorithm for sampling from a null model of datasets, can also be seen as sampling from the set of bipartite graphs with a prescribed BJDM, according to a desired sampling distribution. In this sense, our contributions belong to a long line of works that studies how to generate (bipartite) graphs with prescribed properties and according to a desired probability distribution. The surveys by Cimini et al. [26] and Greenhill [27] give a complete coverage of this field. These approaches have been studied in the context of complex networks, while we use *bipartite* graphs to represent transactional datasets, and our main goal is to statistically assess results obtained from such datasets, not to study the properties of the graphs.

No previous work on sampling bipartite graphs deals with the question we study. Saracco et al. [28] presents a configuration model to sample bipartite networks that, *in expectation*, have the same degree sequences as a prescribed one. ALICE *exactly* maintains the BJDM, which preserves the exact degree sequences, and also other additional properties (see Sect. IV); thus our null model preserves more characteristics of the observed dataset. Aksoy et al. [29] proposes a method to generate bipartite networks that preserve also the clustering coefficient, which is not related to the BJDM. Amanatidis et al. [30] gives necessary and sufficient conditions for a matrix to be the BJDM of a bipartite graph. We always start from such a matrix, so we do not have to address its realizability. The concept of *Restricted Swap Operation (RSO)* was introduced by Czabarka et al. [31], but not for the purpose used in ALICE. Boroojeni et al. [32] presents randomized algorithms to generate a bipartite graph from a BJDM, but there is no proof that their approaches can generate all possible graphs with that BJDM nor there is an analysis on the probability that such a graph is generated. Both aspects are important in order to use the samples for statistical hypothesis testing (see Sect. III-A), and ALICE achieves these goals.

III. PRELIMINARIES

Let $\mathcal{I} \doteq \{a_1, \dots, a_{|\mathcal{I}|}\}$ be a finite alphabet of *items*. W.l.o.g., we can assume $\mathcal{I} = \{1, \dots, |\mathcal{I}|\}$. Any $A \subseteq \mathcal{I}$ is an *itemset*. A *dataset* $\mathcal{D}$ is a finite bag of itemsets, which are known also as *transactions* when considered as the elements of a dataset. The *size* $|\mathcal{D}|$ of the dataset is the number of transactions it contains. The *length* $|t|$ of a transaction $t \in \mathcal{D}$ is the number of items in it. For any itemset $A \subseteq \mathcal{I}$, the *support* $\sigma_{\mathcal{D}}(A)$ *of* A *in* $\mathcal{D}$ is the number of transactions of $\mathcal{D}$ which contain A:

$$\sigma_{\mathcal{D}}(A) \doteq |\{t \in \mathcal{D} : A \subseteq t\}| \ .$$

The support is a natural (albeit not without drawbacks) measure of interestingness. A foundational knowledge discovery task requires to find, given a *minimum support threshold* $\theta \in [0, |\mathcal{D}|]$, the collection $\mathsf{FI}_\theta(\mathcal{D})$ of *Frequent Itemsets (FIs) in* $\mathcal{D}$ *w.r.t.* θ: $\mathsf{FI}_\theta(\mathcal{D}) \doteq \{A \subseteq \mathcal{I} : \sigma_{\mathcal{D}}(A) \geq \theta\}$ [33].

[1] Like the eponymous character of *Alice in Wonderland*, our algorithms explore a large strange world, and interact with caterpillars.

A. Null Models and Hypothesis testing

The statistical hypothesis testing framework [4, Ch. 10] allows to rigorously understand whether the results obtained from an *observed dataset* $\mathring{\mathcal{D}}$ (e.g., the collection of frequent itemsets, or its size, among many others) are actually interesting or are just due to randomness in the (unknown, at least partially) data generation process. Informally, the observed results are compared to the distribution of results that would be obtained from a *null model* (see below); if results as or more extreme than the observed ones are sufficiently unlikely, the observed results are deemed *statistically significant*.

A *null model* $\Pi = (\mathcal{Z}, \pi)$ is a pair where $\mathcal{Z}$ is a set of datasets, and π is a (user-specified) probability distribution over $\mathcal{Z}$. The datasets in $\mathcal{Z}$ are all and only those that share some descriptive characteristics with an *observed dataset* $\mathring{\mathcal{D}}$, which also belongs to $\mathcal{Z}$.[2] Null models in previous works [6, 16] preserve the following two *fundamental properties*:

- the distribution of the transaction lengths, i.e., for any possible transaction length $\ell \in [1, |\mathcal{I}|]$, $\mathcal{D} \in \mathcal{Z}$ contains the same number of transactions of length ℓ as $\mathring{\mathcal{D}}$;[3] and
- the support of the items, i.e., for any $i \in \mathcal{I}$ and $\mathcal{D} \in \mathcal{Z}$, $\sigma_{\mathcal{D}}(i) = \sigma_{\mathring{\mathcal{D}}}(i)$.

The intuition behind wanting to preserve some properties of $\mathring{\mathcal{D}}$ is that these properties, together with π, capture what is known or assumed about the process that generated the data, and the goal is to understand whether the results obtained from $\mathring{\mathcal{D}}$ are, informally, "typical" for datasets with these characteristics. Formally, given $\mathring{\mathcal{D}}$ and a null model $\Pi = (\mathcal{Z}, \pi)$, one formulates a *null hypothesis* H_0 involving Π and a result $R_{\mathring{\mathcal{D}}}$ obtained from $\mathring{\mathcal{D}}$. For example, let $R_{\mathring{\mathcal{D}}} = |\mathsf{FI}_\theta(\mathring{\mathcal{D}})|$, and

$$H_0 \doteq \text{“} \mathop{\mathbb{E}}_{\mathcal{D}\sim\pi} [|\mathsf{FI}_\theta(\mathcal{D})|] = R_{\mathring{\mathcal{D}}} \text{”}.^{4} \tag{1}$$

The hypothesis is then tested by computing the *p-value* $p_{\mathring{\mathcal{D}},H_0}$ *of* H_0, defined as the probability that, in a dataset $\mathcal{D}'$ sampled from $\mathcal{Z}$ according to π, the results $R_{\mathcal{D}'}$ (e.g., $|\mathsf{FI}_\theta(\mathcal{D}')|$) are *more extreme* (e.g., larger) than $R_{\mathring{\mathcal{D}}}$, i.e.,

$$p_{\mathring{\mathcal{D}},H_0} \doteq \Pr_{\mathcal{D}'\sim\pi} (R_{\mathcal{D}'} \text{ more extreme than } R_{\mathring{\mathcal{D}}}) \ . \tag{2}$$

The notion of "more extreme" depends on the nature of $R_{\mathring{\mathcal{D}}}$. When $p_{\mathring{\mathcal{D}},H_0}$ is *not larger* than a user-specified *critical value* α, then the observed results $R_{\mathring{\mathcal{D}}}$ are deemed to be *statistically significant*, i.e., unlikely to be due to random chance (in other words, the null hypothesis H_0 is rejected as not sufficiently supported by the available data).

Computing the p-value $p_{\mathring{\mathcal{D}},H_0}$ from (2) exactly is often impossible, thus an empirical estimate $\tilde{p}_{\mathring{\mathcal{D}},H_0}$ is obtained as follows and used in place of $p_{\mathring{\mathcal{D}}}$ when testing the hypothesis [5]. Let $\mathcal{D}_1, \ldots, \mathcal{D}_T$ be T datasets *independently sampled* from $\mathcal{Z}$ according to π, then

$$\tilde{p}_{\mathring{\mathcal{D}},H_0} \doteq \frac{1 + |\{\mathcal{D}_i : R_{\mathcal{D}_i} \text{ is more extreme than } R_{\mathring{\mathcal{D}}}\}|}{1+T} \ . \tag{3}$$

Thus, efficiently drawing random datasets from $\mathcal{Z}$ according to π plays a key role in statistical hypothesis testing.

B. Markov Chain Monte Carlo Methods

ALICE follows the *Markov chain Monte Carlo (MCMC) method*, and uses the *Metropolis-Hastings (MH) algorithm* [34, Ch. 7 and 10]. Next is an introduction tailored to our work.

Let $G = (V, E)$ be a directed, weighted, strongly connected, aperiodic graph, potentially with self-loops. The *Metropolis-Hastings (MH) algorithm* gives a way to sample an element of V according to a user-specified probability distribution ϕ. Let $v \in V$ be any state, chosen arbitrarily. We first draw a neighbor $u \in \Gamma(v)$ of v according to the distribution ξ_v. Then we "move" from v to u with probability

$$\min\left\{1, \frac{\phi(u)\xi_u(v)}{\phi(v)\xi_v(u)}\right\}, \tag{4}$$

otherwise we stay in v. After a sufficiently large number of steps t, the state v_t is (either approximately or exactly) distributed according to ϕ and can be taken as a sample.

In summary, to be able to use MH, one must define the graph $G = (V, E)$, the neighbor-sampling probability ξ_v for every $v \in V$, a procedure to sample a neighbor of v according to ξ_v, and the desired sampling distribution ϕ over V.

IV. A More Descriptive Null Model

As discussed in Sect. III-A, a good null model should preserve important characteristics of the observed dataset $\mathring{\mathcal{D}}$, and we mentioned the two fundamental properties that previous works have focused on [6, 16]. We now introduce a null model that preserves an additional property, and then show efficient methods to sample datasets from it.

A. Datasets, Matrices, and Bipartite Graphs

Before defining the additional characteristic quantity of $\mathring{\mathcal{D}}$ that we want to preserve, we must describe "alternative" representations of a dataset $\mathcal{D}$. The most natural one is a *binary matrix* $M_{\mathcal{D}}$ with $|\mathcal{D}|$ rows and $|\mathcal{I}|$ columns, where the (i, j) entry is 1 iff transaction $i \in \mathcal{D}$ contains item $j \in \mathcal{I}$, and where the order of the transactions (i.e., of the rows) is arbitrary [6, Sect. 4.1]. Since the order is arbitrary, there are *multiple matrices* that correspond to the same dataset, differing by the ordering of the rows. This fact is of key importance for the correctness of methods that sample datasets (and not matrices) from a null model.

Any matrix $M_{\mathcal{D}}$ corresponding to $\mathcal{D}$ can be seen as the *biadjacency matrix* of an *undirected bipartite graph* $G_{\mathcal{D}} = (\mathcal{D} \cup \mathcal{I}, E)$ corresponding to $\mathcal{D}$, where there is an edge[5] $(t, i) \in E$ iff transaction t contains the item i. Different matrices M' and M'' corresponding to $\mathcal{D}$ are the biadjacency

[2] Thus, Π depends on $\mathring{\mathcal{D}}$, but we hide it in the notation to keep it light.

[3] This property implies that the size of the dataset is preserved as well, i.e., $|\mathcal{D}| = |\mathring{\mathcal{D}}|$ for any $\mathcal{D} \in \mathcal{Z}$.

[4] This hypothesis is just one simple example of many possible different hypotheses that could be tested.

[5] We always denote an edge of a bipartite graph corresponding to a dataset as (a, b) with $a \in \mathcal{D}$ and $b \in \mathcal{I}$, i.e., as an element of $\mathcal{D} \times \mathcal{I}$, to make it clear which endpoint is a transaction and which is an item.

matrices of bipartite graphs that are *structurally equivalent*, up to the labeling of the transactions in $\mathcal{D}$. In other words, all graphs corresponding to a dataset share the *same structural properties*, no matter their biadjacency matrices. To define our new null model we use the graph $G_{\mathring{\mathcal{D}}}$.

B. Preserving the BJDM

One of our goals is to define a null model $\Pi = (\mathcal{Z}, \pi)$ such that the datasets in $\mathcal{Z}$ preserve not only the two fundamental properties, but also an additional descriptive property of $\mathring{\mathcal{D}}$: the *Bipartite Joint Degree Matrix (BJDM)* $\mathsf{J}_{G_{\mathring{\mathcal{D}}}}$ of its bipartite graph representation $G_{\mathring{\mathcal{D}}}$.

Definition 1 (BJDM). *Let $G = (L \cup R, E)$ be a bipartite graph, k_L and k_R be the largest degree of a node in L and R, respectively. The* Bipartite Joint Degree Matrix (BJDM) J_G *of G, is a $k_L \times k_R$ matrix whose (i,j)-th entry $\mathsf{J}_G[i,j]$ is the number of edges connecting a node $u \in L$ with degree $\mathsf{deg}(u) = i$ to a node $v \in R$ with degree $\mathsf{deg}(v) = j$, i.e.,*

$$\mathsf{J}_G[i,j] \doteq |\{(u,v) \in E : \mathsf{deg}(u) = i \wedge \mathsf{deg}(v) = j\}| \ .$$

We define $\mathcal{Z}$ as the set of all datasets $\mathcal{D}$ whose transactions are built on $\mathcal{I}$ and whose corresponding bipartite graph $G_{\mathcal{D}}$ has the same BJDM $\mathsf{J}_{G_{\mathring{\mathcal{D}}}}$. We justify this choice by first showing that preserving the BJDM also preserves the two fundamental properties, and then that it preserves additional ones.

Fact 1. *For every $1 \le j \le k_R$, it holds*

$$|\{v \in R : \mathsf{deg}(v) = j\}| = \frac{1}{j} \sum_{i=1}^{k_L} \mathsf{J}_G[i,j], \tag{5}$$

i.e., the BJDM J_G determines, for every $1 \le j \le k_R$, the number of vertices $v \in R$ of degree $\mathsf{deg}(v) = j$.

Similarly, for every $1 \le i \le k_L$, it holds

$$|\{u \in L : \mathsf{deg}(u) = i\}| = \frac{1}{i} \sum_{j=1}^{k_R} \mathsf{J}_G[i,j] \tag{6}$$

i.e., the BJDM J_G determines, for every $1 \le i \le k_L$, the number of vertices $u \in L$ with degree $\mathsf{deg}(u) = i$.

Corollary 1. *For any dataset $\mathcal{D}$, the BJDM $\mathsf{J}_{G_{\mathcal{D}}}$ determines, for every $1 \le j \le |\mathcal{I}|$, the number of transactions in $\mathcal{D}$ with length j. Also, it determines, for every $1 \le i \le |\mathcal{D}|$, the number of items with support i in $\mathcal{D}$.*

Corollary 1 states that preserving the BJDM also preserves the two fundamental properties. We now show an additional property that is preserved, among others.

Let $\mathsf{z}(G_{\mathring{\mathcal{D}}})$ be the number of *simple paths of length three* in $G_{\mathring{\mathcal{D}}}$, which, since $G_{\mathring{\mathcal{D}}}$ is bipartite, is also known as the number of *caterpillars* of $G_{\mathring{\mathcal{D}}}$ [29]. Corollary 2 shows that preserving the BJDM of $G_{\mathring{\mathcal{D}}}$ preserves the number of caterpillars. The numbers of simple paths of length one and two are already preserved by preserving the two fundamental properties, thus preserving also the number of simple paths of length three is a natural step. Our desired result is a corollary of Lemma 1, which shows that $\mathsf{z}(G)$ can be expressed through the BJDM.

Lemma 1. *It holds*

$$\mathsf{z}(G) = \sum_{i=2}^{k_L} \sum_{j=2}^{k_R} \mathsf{J}_G[i,j](i-1)(j-1) \ .$$

Proof: Each edge $(u,v) \in E$ is the middle edge of $(\mathsf{deg}(u)-1)(\mathsf{deg}(v)-1)$ caterpillars, so

$$\mathsf{z}(G) = \sum_{(u,v) \in E} (\mathsf{deg}(u)-1)(\mathsf{deg}(v)-1) \ . \tag{7}$$

From here, we can conclude that

$$\sum_{(u,v) \in E} (\mathsf{deg}(u)-1)(\mathsf{deg}(v)-1) = \sum_{i=2}^{k_L} \sum_{j=2}^{k_R} \mathsf{J}_G[i,j](i-1)(j-1)$$

because each edge $(u,v) \in E$ that connects a node $u \in L$ with degree $\mathsf{deg}(u) = i$ to a node $v \in R$ with degree $\mathsf{deg}(v) = j$ contributes $(i-1)(j-1)$ caterpillars to the summation in Eq. (7), and there are $\mathsf{J}_G[i,j]$ such edges. ■

Corollary 2. *For any $\mathcal{D}$, the BJDM $\mathsf{J}_{G_{\mathcal{D}}}$ determines $\mathsf{z}(G_{\mathcal{D}})$.*

We remark that preserving the two fundamental properties and the number of caterpillars does not imply that the BJDM is preserved: it is easy to construct datasets that have the same transaction lengths, same item supports, and same number of caterpillars as an observed dataset $\mathring{\mathcal{D}}$, but whose BJDM is different than $\mathsf{J}_{G_{\mathring{\mathcal{D}}}}$. We show an example in App. A.

We considered preserving more "natural" characteristics than the BJDM, such as the support of each itemset of length two. However, doing so would lead to null sets $\mathcal{Z}$ that contain very few datasets in most cases, and are therefore not very informative about the data generation process, as they are likely overly constrained. Informally, the reason is that the biadjacency matrix $M_{\mathcal{D}}$ of the graph $G_{\mathcal{D}}$ corresponding to any dataset $\mathcal{D}$ in such a $\mathcal{Z}$ must satisfy $M_{\mathcal{D}} M_{\mathcal{D}}^{\intercal} = M_{\mathring{\mathcal{D}}} M_{\mathring{\mathcal{D}}}^{\intercal}$ and have the same row and column sums as $M_{\mathring{\mathcal{D}}}$. There are very few such matrices and the relative size of their set decreases as the number of transactions in $\mathring{\mathcal{D}}$ and/or the number of items in $\mathcal{I}$ grow [35]. We defer an in-depth discussion of this case to the extended version of this work.

V. Sampling from the Null Model

We now present ALICE, a suite of two algorithms for sampling datasets from the null model $\Pi = (\mathcal{Z}, \pi)$.

ALICE takes the MCMC approach with MH (see Sect. III-B). Its set of states is the set $\mathcal{M}$ of matrices defined as follows. Fix $M_{\mathring{\mathcal{D}}}$ to be any of the biadjacency matrices of a bipartite graph corresponding to the observed dataset $\mathring{\mathcal{D}}$. $\mathcal{M}$ contains all and only the matrices M of size $|\mathring{\mathcal{D}}| \times |\mathcal{I}|$ such that, when considering M as the biadjacency matrix of a bipartite graph G_M, it holds $\mathsf{J}_{G_M} = \mathsf{J}_{G_{\mathring{\mathcal{D}}}}$.

$\mathcal{M}$ may contain multiple matrices associated to the same dataset (see Sect. IV-A), and different datasets may have a different number of matrices in $\mathcal{M}$ associated to them. ALICE takes this fact into account to ensure that the sampling of datasets from $\mathcal{Z}$ is done according to π. For $M \in \mathcal{M}$, we use $\mathsf{dat}(M)$ to denote the unique dataset corresponding to M, and for a dataset $\mathcal{D} \in \mathcal{Z}$, we use $\mathsf{mat}(\mathcal{D})$ to denote the *set*

of matrices in $\mathcal{M}$ corresponding to $\mathcal{D}$. The following result, whose combinatorial proof we omit due to space limitations, gives an expression for the size $\mathsf{c}(\mathcal{D}) \doteq |\mathsf{mat}(\mathcal{D})|$.

Lemma 2. *For any dataset $\mathcal{D} \in \mathcal{Z}$, let $\{\ell_1, \ldots, \ell_{z_\mathcal{D}}\}$ be the set of the $z_\mathcal{D}$ distinct lengths of the transactions in $\mathcal{D}$. For each $1 \le i \le z_\mathcal{D}$, let T_i be the* bag *of transactions of length ℓ_i in $\mathcal{D}$. Let $\bar{T}_i = \{\tau_{i,1}, \ldots, \tau_{i,r_i}\}$ be the* set *of transactions of length ℓ_i in $\mathcal{D}$, i.e., without duplicates. For each $1 \le j \le r_i$, let $Q_{i,j} \doteq \{t' \in T_i : t' = \tau_{i,j}\}$ be the* bag *of transactions in T_i equal to $\tau_{i,j}$ (including $\tau_{i,j}$). Then, the number of matrices M in $\mathcal{M}$ such that $\mathsf{dat}(M) = \mathcal{D}$ is*

$$\mathsf{c}(\mathcal{D}) = \prod_{i=1}^{z_\mathcal{D}} \underbrace{\binom{|T_i|}{|Q_{i,1}|, \ldots, |Q_{i,r_i}|}}_{\textit{multinomial coefficient}} = \prod_{i=1}^{z_\mathcal{D}} \frac{|T_i|!}{\prod_{j=1}^{r_i} |Q_{i,j}|!} \quad . \tag{8}$$

ALICE takes as inputs π and the observed dataset $\mathring{\mathcal{D}}$. It uses MH (see Sect. III-B) to sample a matrix $M \in \mathcal{M}$ according to a distribution ϕ (defined below), and returns $\mathcal{D} = \mathsf{dat}(M) \in \mathcal{Z}$ distributed according to π. Both algorithms we present share the same set $\mathcal{M}$ of states, but they have different neighborhood structures (i.e., the graphs used by MH for the two algorithms have different sets of edges), different neighbor distributions ξ_M, $M \in \mathcal{M}$, and different neighbor sampling procedures.

A. ALICE-A: *RSO-based Algorithm*

In our first algorithm, ALICE-A, the neighborhood structure over $\mathcal{M}$ is defined by using *Restricted Swap Operations (RSOs)* [31, Sect. 2].

Definition 2 (RSO)**.** *Let M be the $|L|\times|R|$ bi-adjacency matrix of a bipartite graph $G = (L \cup R, E)$. Let $1 \le a \ne b \le |L|$ and $1 \le c \ne d \le |R|$ be the indices of two rows and columns of M, respectively, such that*

$$M[a,c] = M[b,d] \wedge M[a,d] = M[b,c] \wedge M[a,c] \ne M[a,d]$$

and such that at least one *of the following conditions holds*

$$C_{ab} = \text{“} \sum_{j=1}^{|R|} M[a,j] = \sum_{j=1}^{|R|} M[b,j] \text{”}$$
$$C_{cd} = \text{“} \sum_{i=1}^{|L|} M[i,c] = \sum_{i=1}^{|L|} M[i,d] \text{”} \quad .$$

The Restricted Swap Operation (RSO) $(a,c),(b,d) \to (a,d),(b,c)$ on M *is the operation that obtains the matrix M' which is the same as M but $M'[a,c] = M[a,d]$, $M'[a,d] = M[a,c]$, $M'[b,c] = M[b,d]$, and $M'[b,d] = M[b,c]$.*

Any RSO on $M \in \mathcal{M}$ results in a matrix M' that belongs to $\mathcal{M}$ as well. In the graph $G = (\mathcal{M}, E)$ needed for MH, there is an edge from M to M' if there is an RSO from M to M'. Additionally, there are *self-loops* from any $M \in \mathcal{M}$ to itself. These self-loops do not correspond to RSOs, but they simplify the neighbor sampling procedure (described next). There are zero or one RSOs between any pair of matrices in $\mathcal{M}$, but $\mathcal{M}$ is strongly connected by RSOs [31, Thm. 8].[6]

[6]The proof of [31, Thm. 8] must be adapted, in a straightforward way, to account for the fact that $\mathcal{M}$ contains biadjacency matrices of bipartite graphs.

RSOs are just one of the many possible operations that make $\mathcal{Z}$ strongly connected. We discuss one such different operation in Sect. V-B. Finding other operations to replace RSOs or to use in addition to RSOs is a interesting research direction.

We now discuss the second ingredient needed to use MH: the distribution ξ_M over the set of neighbors $\Gamma(M)$ of any $M \in \mathcal{M}$. At first, using a distribution ξ_M of the form

$$\xi_M(M') \doteq \begin{cases} \frac{2}{|\mathcal{I}|^2|\mathcal{D}|^2} & M' \in \Gamma(M) \setminus \{M\} \\ 1 - \frac{2(|\Gamma(M)|-1)}{|\mathcal{I}|^2|\mathcal{D}|^2} & M' = M \end{cases}$$

may seem an appealing option, because it could be realized by first drawing a 4-tuple (a,b,c,d) uniformly at random from $\mathcal{D}\times\mathcal{D}\times\mathcal{I}\times\mathcal{I}$, and then verifying whether $(a,c),(b,d) \to (a,d),(b,c)$ is an RSO: if it is, one would set M' to be the matrix resulting from applying the RSO to M, otherwise $M' = M$. The major issue with this approach is that, depending on M, the number of tuples that must be drawn before finding one that is an RSO may be very large, thus slowing down the process of moving on the graph. Conversely, more complex probability distributions that ensure drawing a neighbor different than M are quite easy to define, but come with the serious drawback that they need expensive computation and bookkeeping of quantities such as $|\Gamma(M)|$ and $|\Gamma(M')|$ for $M' \in \Gamma(M)$ (due to Eq. (4)), or the number of pairs of different rows/columns of the same lengths in M and $M' \in \Gamma(M)$. The process of sampling a neighbor would then be much more expensive, thus again slowing down the walk on the graph. We propose a distribution over $\Gamma(M)$ and a procedure to sample from it that strikes a balance between statistical and computational “efficiency”, i.e., the probability of sampling M is smaller than in the naïve case described above, and sampling a neighbor is still quite efficient.

Let $M \in \mathcal{M}$ be the current state. For any $1 \le m \le |\mathcal{I}|$ (resp. $1 \le n \le |\mathcal{D}|$), let R_m be the set of row indices in M whose rows have sum m (resp. let C_n be set of column indices in M whose columns have sum n). To sample a neighbor M' of M, we start by flipping a fair coin. If the outcome is *heads*, we first draw a row sum $1 \le m \le |\mathcal{I}|$ with probability

$$\beta(m) = \binom{|R_m|}{2} \Big/ \sum_{j=1}^{|\mathcal{I}|} \binom{|R_j|}{2}, \tag{9}$$

and then we draw a pair (a,b) of *different* row indices in R_m uniformly at random between such pairs. If the row of index a and the row of index b in M are identical, then we set $M' = M$. Otherwise, consider the set $H_{a,b}$ of column index pairs (p,q) such that $M[a,p] = M[b,q]$, $M[a,q] = M[b,p]$, and $M[a,p] \ne M[a,q]$. We draw a pair (c,d) from $H_{a,b}$ uniformly at random. Clearly $(a,c),(b,d) \to (a,d),(b,c)$ is an RSO by construction, and we set M' to be the matrix obtained by performing this RSO on M. If the outcome of the coin flip is *tails*, we first draw a column sum $1 \le n \le |\mathcal{D}|$

with probability

$$\gamma(n) = \binom{|C_n|}{2} \Big/ \sum_{j=1}^{|\mathcal{D}|} \binom{|C_j|}{2}, \tag{10}$$

and then we draw a pair (c, d) of different column indices in C_n uniformly at random between such pairs. If the column of index c and the column of index d in M are identical, then we set $M' = M$. Otherwise, consider the set $K_{c,d}$ of row index pairs (p, q) such that $M[p, c] = M[q, d]$, $M[p, d] = M[q, c]$, and $M[p, c] \neq M[p, d]$. We draw a pair (a, b) from $K_{c,d}$ uniformly at random. Clearly $(a, c), (b, d) \to (a, d), (b, c)$ is also an RSO by construction, and we set M' to be the matrix obtained by performing this RSO on M.

This procedure induces a probability distribution ξ_M over $\Gamma(M)$. Let us analyze $\xi_M(M')$ for $M' \neq M$. Let $(a, c), (b, d) \to (a, d), (b, c)$ be the sampled RSO, and let M' be the neighbor of M obtained by performing such RSO on M. Recall that the sampled RSO is the only RSO from M to M'. Consider the following events:

$E_{\text{row}} \doteq$ "rows a and b of M have the same row sum m";

$E_{\text{col}} \doteq$ "columns c and d of M have the same column sum n".

There are three possible cases for the probability $\xi_M(M')$ of sampling M':

- if only E_{row} holds, then

$$\xi_M(M') = \frac{1}{2} \frac{1}{\sum_{j=1}^{|\mathcal{I}|} \binom{|R_i|}{2}} \frac{1}{|H_{a,b}|}; \tag{11}$$

- if only E_{col} holds, then

$$\xi_M(M') = \frac{1}{2} \frac{1}{\sum_{j=1}^{|\mathcal{D}|} \binom{|C_j|}{2}} \frac{1}{|K_{a,b}|}; \tag{12}$$

- if both E_{row} and E_{col} hold, then M' (i.e., the RSO) may be sampled regardless of the outcome of the coin flip. Thus, $\xi_M(M')$ is the sum of r.h.s.'s of Eq. (11) and Eq. (12).

We do not need to analyze $\xi_M(M)$ because if M is drawn as the "neighbor", then MH will definitively select M as the next state, thus we do not need to explicitly compute its probability.

It holds that $\xi_M(M') = \xi_{M'}(M)$, which greatly simplifies the use of MH: from Eq. (4), we see that, thanks to the construction of the graph and the definition of the neighbor sampling distribution, we really only need the distribution ϕ over $\mathcal{M}$. We define it as

$$\phi(M) = \frac{\pi(\mathsf{dat}(M))}{\mathsf{c}(\mathsf{dat}(M))}, \tag{13}$$

where $\mathsf{c}(\mathsf{dat}(M))$ is from Eq. (8). The following lemma shows that ALICE-A samples a dataset $\mathcal{D}$ from $\mathcal{Z}$ according to π, i.e., it samples from the null model.

Lemma 3. *Let $\mathcal{D} \in \mathcal{Z}$. ALICE-A outputs $\mathcal{D}$ with prob. $\pi(\mathcal{D})$.*

Proof: Let $M \in \mathcal{M}$. From the correctness of MH we have that ALICE-A samples M according to ϕ from Eq. (13). The thesis then follows from noticing that $\mathcal{D}$ is returned in output whenever ALICE-A samples one of the $\mathsf{c}(\mathcal{D})$ matrices in $\mathcal{M}$ corresponding to $\mathcal{D}$. ■

B. ALICE-B: *Adapting Curveball*

We now introduce a second algorithm, ALICE-B, that can essentially perform multiple RSOs at each step of the Markov chain, thus leading to a faster mixing of the chain, i.e., to fewer steps needed to sample a dataset from Π. Our approach adapts the CURVEBALL algorithm [7] to use RSOs. Due to space limitations, we do not discuss the original CURVEBALL algorithm, introduced for sampling a matrix from the space of binary matrices with fixed row and column sums. ALICE-B is also an MCMC algorithm that uses MH. The vertex set of the graph $G = (\mathcal{M}, E)$ is still the set $\mathcal{M}$ previously defined, but ALICE-B uses a different set of edges than ALICE-A: there is an edge $(M, M') \in E$ from a matrix $M \in \mathcal{M}$ to $M' \in \mathcal{M}$ iff $M' = M$ or there is a *Restricted Binomial Swap Operation (RBSO)* on M that results in M'. RBSOs are defined as follows.

Definition 3 (Restricted Binomial Swap Operation (RBSO))**.** *Given a matrix $M \in \mathcal{M}$, let a and b be the indices of two* distinct and different *rows of M with the same row sum. Let $Z_a(M, b)$ be the set of column-indices q such that $M[a, q] = 1$ and $M[b, q] = 0$, and define $Z_b(M, a)$ similarly (it holds $Z_a(M, b) \cap Z_b(M, a) = \varnothing$ and $|Z_a(M, b)| = |Z_b(M, a)|$). Let S be any subset of $Z_a(M, b) \cup Z_b(M, a)$ of size $|Z_a(M, b)|$. The* row Restricted Binomial Swap Operation (rRBSO) *(a, b, S) on M is the operation that obtains a matrix M' such that $M'[i, j] = M[i, j]$ except for $i \in \{a, b\}$, and such that the rows of index a and b of M' are*

$$M'[a, q] \doteq \begin{cases} M[a, q] & q \notin Z_a(M, b) \cup Z_b(M, a) \\ 1 & q \in S \\ 0 & q \in (Z_a(M, b) \cup Z_b(M, a)) \setminus S \end{cases}$$

and

$$M'[b, q] \doteq \begin{cases} M[b, q] & q \notin Z_a(M, b) \cup Z_b(M, a) \\ 0 & q \in S \\ 1 & q \in (Z_a(M, b) \cup Z_b(M, a)) \setminus S \end{cases}$$

A corresponding definition for a column RBSO (cRBSO) *can be given for a and b being the indices of two distinct and different columns with the same column sum.*

We use "RBSO" to refer to either a rRBSO or a cRBSO, and the set of RBSOs is composed by all rRBSOs and cRBSOs.

Any RBSO on a matrix M preserves J_M. Any RBSO can be seen as a sequence of RSOs. For any RSO $(a, c), (b, d) \to (a, d), (b, c)$ on M there is an equivalent RBSO $(a, b, (Z_a(M, b) \setminus \{c\}) \cup \{d\})$ from M, and thus the graph $G = (\mathcal{M}, E)$ is also strongly connected, as it has all the edges which are created by RSOs, plus potentially others.

Fact 2. *Let (a, b, S) be a cRBSO (resp. rRBSO) from M to $M' \in \Gamma(M)$ with $M' \neq M$. Then $(a, b, Z_a(M, b))$ is a cRBSO (resp. rRBSO) from M' to M.*

Lemma 4. *There are either one or two RBSOs from $M \in \mathcal{M}$ to $M' \in \Gamma(M)$ with $M' \neq M$. When there are* two *RBSOs, one is a cRBSO and the other is a rRBSO.*

Proof: Let us start from the second part of the thesis. If $(a, b, \{c\})$ is a cRBSO (resp. rRBSO) from M to M', then

$$(c, (Z_a(M, b) \cup Z_b(M, a)) \setminus \{c\}, \{a\})$$

is a rRBSO (resp. cRBSO) from M to M'.

The fact that there can only be one or two RBSOs is a consequence of Fact 2. ∎

In order for two RBSOs from M to M' to exist, it is necessary that $|Z_a(M, b)| = |Z_b(M, a)| = 1$, the columns at indices a and b have the same sum, and the rows at indices c and $(Z_a(M, b) \cup Z_b(M, a)) \setminus \{c\}$ have the same sum.

Corollary 3. *For any two M and M', there is the same number of RBSOs from M to M' as from M' to M.*

Let us now give the procedure to sample a neighbor $M' \in \Gamma(M)$ of M. The procedure is similar to the one for ALICE-A. First, we flip a fair coin. If the outcome is *heads*, we draw a row sum $1 \leq m \leq |\mathcal{I}|$ with probability as per Eq. (9), and then we draw a pair (a, b) of different row indices in R_m uniformly at random between such pairs. If the row of index a and the row of index b in M are identical, then we set $M' = M$. Otherwise, we compute the set $Z_a(M, b) \cup Z_b(M, a)$ defined in Def. 3 and the cardinality $|Z_a(M, b)|$ with a linear scan of the rows a and b. By using reservoir sampling, we obtain S through a linear scan of $Z_a(M, b) \cup Z_b(M, a)$. If the outcome of the coin flip is *tails*, we first draw a column sum $1 \leq n \leq |\mathcal{D}|$ with probability as per Eq. (10), then we draw a pair (a, b) of different column indices in C_n uniformly at random between such pairs. We then proceed in a fashion similar as for the row case. The purpose of flipping the coin at the start is to ensure that we can sample both rRBSOs (when the outcome is heads), and cRBSOs (otherwise).

The probability $\xi_M(M')$ of sampling a RBSO (a, b, S) on M that results in M', is not uniform. Rather than giving the expression for it, we use the fact that, in order to use MH, we really only need the distribution ϕ over $\mathcal{M}$, and the *ratio* $\xi_{M'}(M)/\xi_M(M')$ (see Eq. (4)), and we now show that $\xi_M(M') = \xi_{M'}(M)$, i.e., the ratio is always 1.

Lemma 5. *Let $M \in \mathcal{M}$ and $M' \in \Gamma(M)$. Then $\xi_M(M') = \xi_{M'}(M)$.*

Proof: We assume that $M' \neq M$, otherwise the thesis is obviously true. For ease of presentation, we focus on the case where there is only a cRBSO (a, b, S) from M to M'. The analysis for the case when there is only a rRBSO follows the same steps, and the one for the case when there is both a cRBSO and a rRBSO follows by combining the two cases.

From Fact 2, the cRBSO $(a, b, (Z_a(M, b))$ goes from M' to M. The probability that the coin flip is tails is the same no matter whether the current state is M or if it is M, as is the probability, given that the outcome was tails, of sampling the columns indices a and b. By definition, it holds that $|S| = |Z_a(M, b)|$, and it is easy to see that $Z_a(M, b) \cup Z_b(M, a) = Z_a(M', b) \cup Z_b(M', a)$, thus the probability of sampling S when the current state is M and we have sampled a and b, and the probability of sampling $Z_a(M, b)$ when the current state is M' and we have sampled a and b are the same. Thus, the probability of sampling (a, b, S) when the current state is M is the same as the probability of sampling $(a, b, Z_a(M, b))$ when the current state is M', and the proof is complete. ∎

Thus, to use MH, we really only need the distribution ϕ over $\mathcal{M}$. As in Sect. V-A, in order to sample a dataset $D \in \mathcal{Z}$ according to π, we want to sample a matrix $M \in \mathcal{M}$ with the probability given in Eq. (13). We thus have all the ingredients to use MH, and our description of ALICE-B is complete.

TABLE I
DATASETS STATISTICS: NUM. OF TRANSACTIONS, NUM. OF ITEMS, SUM OF TRANSACTION LENGTHS, AVG. TRANSACTION LENGTH, AND DENSITY.

Dataset	Trans. Num	Item Num	Sum Trans. Lengths	AVG Trans. Length	Density
iewiki	137	558	651	4.752	0.0085
kosarak	3000	5767	23664	7.888	0.0014
chess	3196	75	118252	37.000	0.4933
foodmart	4141	1559	18319	4.424	0.0028
db-occ	10000	8984	19729	1.973	0.0002
BMS1	59602	497	149639	2.511	0.0051
BMS2	77512	3340	358278	4.622	0.0014
retail	88162	16470	908576	10.306	0.0006

VI. EXPERIMENTAL EVALUATION

Our evaluation pursues three goals: empirically study the mixing time of the sampling algorithms, evaluate their scalability as the number of transactions increases, and show that the null model we introduce differs from that which only preserves the two fundamental properties, by showing that it leads to marking different hypotheses as significant.

Datasets. We use eight real-world datasets,[7] listed in Table I. Density is the ratio between the average transaction length and the number of items. **iewiki** is a user-edit dataset, where each transaction is a set of Wikibooks pages edited by the same user; **kosarak**, **BMS1**, and **BMS2** are click-stream datasets; **chess** is a board-description datasets adapted from the UCI Chess (King-Rook vs King-Pawn) dataset; **foodmart** and **retail** are retail transaction datasets; and **db-occ** includes user occupations taken from dbpedia.

Experimental Environment. We run our experiments on a 40-Core (2.40 GHz) Intel® Xeon® Silver 4210R machine, with 384GB of RAM, and running FreeBSD 14.0. Results are compared against GMMT [6], which is a swap randomization algorithm that samples from the null model that only maintains the two fundamental properties. All the samplers are implemented in Java 1.8, and the code is available at https://github.com/acdmammoths/alice.

Convergence. To study the convergence of our samplers, we follow a procedure similar to the one proposed by Gionis et al. [6]. The mixing time, i.e., the number of steps needed for the

[7]From www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php and http://konect.cc/networks.

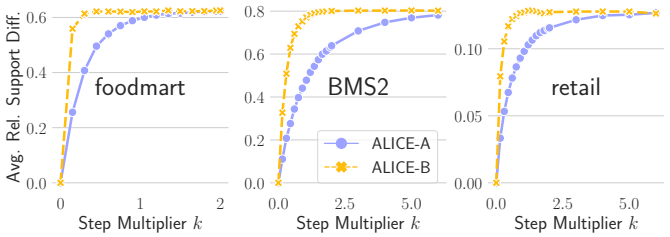

Fig. 1. Convergence of the samplers increasing the step number multiplier k, for foodmart (left), BMS2 (middle), and retail (right).

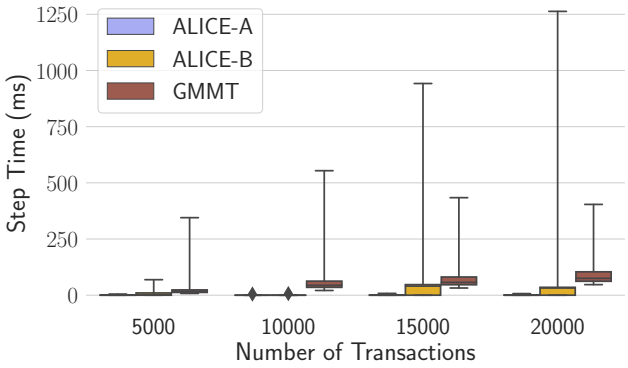

Fig. 2. Step times of the samplers in the synthetic datasets.

state of the chain to be distributed according to π, is estimated by looking at the convergence of the *Average Relative Support Difference (ARSD)*, defined as

$$ARSD(\mathcal{D}^s) = \frac{1}{|\mathsf{FI}_\theta(\mathring{\mathcal{D}})|} \sum_{A \in \mathsf{FI}_\theta(\mathring{\mathcal{D}})} \frac{|\sigma_{\mathring{\mathcal{D}}}(A) - \sigma_{\mathcal{D}^s}(A)|}{|\sigma_{\mathring{\mathcal{D}}}(A)|},$$

where $\mathcal{D}^s$ is the dataset obtained by the sampler after s steps. Figure 1 reports this quantity for foodmart (left), BMS2 (middle), and retail (right), for $s = \lfloor k \cdot w \rfloor$ with $k \in \{0, 0.15, 0.3, \ldots, 2, 3, \ldots, 6\}$ and $w = \sum_{t \in \mathring{\mathcal{D}}} |t|$. Results for other datasets were qualitatively similar. ALICE-B needs $1/3$ or even fewer steps than ALICE-A, thanks to to the fact that it essentially performs multiple RSOs at each step (as each RBSO corresponds to one or more RSOs).

Despite the fewer number of *steps* needed, the *(wall clock) time* to convergence of ALICE-B (not reported in figures), however, is higher than that of ALICE-A. This difference is due to the fact that performing an RBSO, which is a more complex operation than an RSO, requires additional bookkeeping for each element in the set S (see Def. 3). In the worst cases (BMS1, and chess), ALICE-B takes almost 10x the time of ALICE-A to reach convergence. An interesting direction for future work is to study how to avoid this additional bookkeeping in ALICE-B to obtain the same advantage over ALICE-A observed for the number of steps to convergence also for the wall clock time.

Scalability. To study the scalability of ALICE, we create four synthetic datasets with increasing number of transactions ($\{5k, 10k, 15k, 20k\}$), 100 items, and average transaction length 25, by using the IBM Quest generator [33]. For each

TABLE II
STEP TIME (MS): MINIMUM, 1ST QUARTILE, MEDIAN, 3RD QUARTILE, AND MAXIMUM OVER 10K STEPS.

Dataset	Sampler	min	Q1	med.	Q3	max
iewiki	ALICE-A	< 1	< 1	< 1	< 1	3
	ALICE-B	< 1	< 1	< 1	< 1	7
	GMMT	< 1	< 1	< 1	< 1	1
kosarak	ALICE-A	< 1	< 1	< 1	< 1	4
	ALICE-B	< 1	< 1	< 1	< 1	7
	GMMT	1	2	2	3	19
chess	ALICE-A	< 1	< 1	< 1	1	5
	ALICE-B	< 1	< 1	3	4	59
	GMMT	7	16	25	38	357
foodmart	ALICE-A	< 1	< 1	< 1	< 1	4
	ALICE-B	< 1	< 1	< 1	< 1	6
	GMMT	1	2	2	3	17
db-occ	ALICE-A	< 1	< 1	< 1	< 1	5
	ALICE-B	< 1	< 1	< 1	< 1	6
	GMMT	1	3	3	3	24
BMS1	ALICE-A	< 1	< 1	< 1	< 1	5
	ALICE-B	< 1	< 1	< 1	1	6
	GMMT	21	45	48	50	537
BMS2	ALICE-A	< 1	< 1	< 1	< 1	4
	ALICE-B	< 1	< 1	< 1	1	8
	GMMT	62	84	89	94	126
retail	ALICE-A	< 1	< 1	< 1	< 1	4
	ALICE-B	< 1	< 1	< 1	< 1	9
	GMMT	111	167	179	189	296

sampler, we perform 10k steps and compute the distribution of step times, reported in Fig. 2. For completeness, we include the step times of GMMT, although they are not really comparable to those of our algorithms, because GMMT samples from a different null set $\mathcal{Z}$ which includes datasets with different BJDMs. The median step time scales linearly with the size of the dataset. ALICE-A is the fastest sampler, requiring less than 8ms to perform a step in the largest dataset, and less than 1ms in most of the cases. In contrast, the step times of ALICE-B are characterized by more variability, as they depend on *(i)* whether the performed RBSO is an rRBSO or a cRBSO, and *(ii)* the size of the set S: the time required to compute $\mathsf{c}(\mathcal{D})$ is larger for cRBSO, and it grows with the size of S.

Table II reports the min, Q1, median, Q3, and max time required to perform a step, for each sampler. The step time of ALICE-B tends to be larger in chess, despite it not being the largest dataset. This fact is due to the high density of this dataset, and its large transaction length (37). Hence, the size of S is usually high. In foodmart, on the other hand, the average transaction length is 4.42 and the average item support is 5.6, so the size of S is often 1. An algorithmic improvement in the bookkeeping due to the size of S would results in better performance of ALICE-B, as mentioned above.

Significance of the Number of Frequent Itemsets. To show that the null model we introduce is different than the one that only preserves the two fundamental properties, We test the null hypothesis H_0 from Eq. (1), and estimate the p-value as in Eq. (3) with $T = 4352$ samples from the null model, for each sampler.[8] We remark that this kind of hypothesis is just a simple but clear example of the tasks that can (and should) be formed to assess the statistical validity of results

[8]The number of steps is empirically fixed according to the results obtained in the convergence experiment.

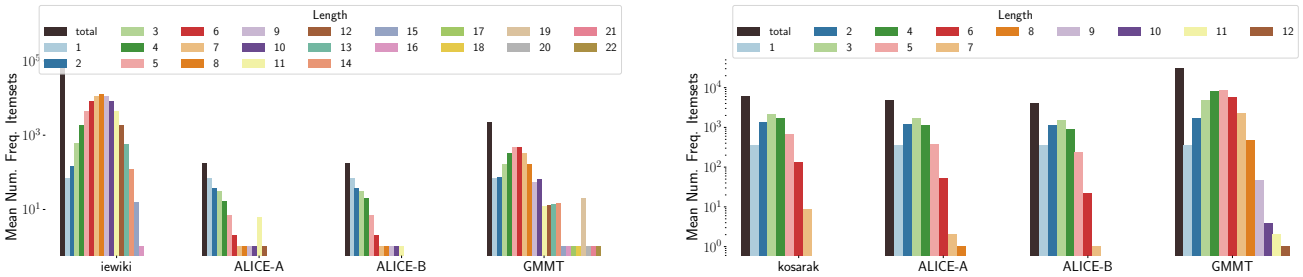

Fig. 3. Mean number of frequent itemsets per length for ALICE-A, ALICE-B, and GMMT, in iewiki (left) and kosarak (right).

TABLE III
NO. OF FIS IN THE ORIGINAL DATASET $\mathring{\mathcal{D}}$, AVG. NO. OF FIS IN THE SAMPLE $\mathcal{D}_i$, ESTIMATED P-VALUE $\tilde{p}_{\mathring{\mathcal{D}},H_0}$ FOR H_0 FROM EQ. (1).

Dataset	$\lvert\mathsf{FI}_\theta(\mathring{\mathcal{D}})\rvert$	Sampler	$\frac{\sum_1^T \lvert\mathsf{FI}_\theta(\mathcal{D}_i)\rvert}{T}$	$\tilde{p}_{\mathring{\mathcal{D}},H_0}$
iewiki θ = 1.4E-2	65665	ALICE-A	173	2.3E-4
		ALICE-B	171	2.3E-4
		GMMT	2257	1.8E-2
kosarak θ = 3.0E-3	6277	ALICE-A	4865	2.3E-4
		ALICE-B	4130	2.3E-4
		GMMT	31774	1.0E-0
chess[9] θ = 0.8	8227	ALICE-A	6183	4.6E-4
		ALICE-B	6182	4.6E-4
		GMMT	6179	4.6E-4
foodmart θ = 3.0E-4	4247	ALICE-A	2229	2.3E-4
		ALICE-B	2228	2.3E-4
		GMMT	2226	2.3E-4
db-occ θ = 5.0E-4	834	ALICE-A	702	2.3E-4
		ALICE-B	703	2.3E-4
		GMMT	598	2.3E-4

obtained from transactional datasets. Other tasks include, for example, mining the statistically-significant frequent itemsets. We limit ourselves to this task because it is straightforward to present and it is sufficient to show the significant (pun intended) difference between preserving the BJDM, as our null model does, and not preserving it.

Table III reports the number of FIs in the observed dataset, the average number of FIs in the sampled datasets, and the empirical p-value, for datasets where GMMT terminated within two days. The fact that (very) different p-values can be obtained with ALICE and with GMMT, which sample from a different null model, highlights the striking impact of preserving the BJDM. As an example, for any critical value in $(0.00023, 0.01815)$, in iewiki H_0 would be rejected under the null model we introduce, but not under the null model that only preserves the two fundamental properties. Figure 3 shows the distribution of the number of FIs of different lengths in the original dataset, and the average of the same quantity over the datasets sampled by the different samplers. Since they sample from the same null model, ALICE-A and ALICE-B obtain the same distribution (up to sampling noise), which is quite different than the one obtained by GMMT. Note that whether the sampled datasets have more or less FIs than the observed dataset depends both on the null model and on the dataset. For instance, in iewiki (Fig. 3, left) datasets sampled from all null models have fewer FIs than the observed one. Conversely, in kosarak (Fig. 3, right) the BJDM-preserving null model produces samples with a similar number of FIs, while the datasets sampled from the null model that preserves the two fundamental properties have a larger number of FIs. In addition, in iewiki, the samples from this latter model usually contain FIs of length larger than any FIs in the observed dataset: the max length of a FI in iewiki is 16, whereas it grows to 22 in the datasets sampled by GMMT. In kosarak, the datasets sampled by GMMT contain both a larger number of FIs per length and FIs of larger length (12 vs. 7). The increase in the number of FIs of length 3, leads to a substantial difference in the number of FIs of length in the range $[4, 7]$: we observe up to 246x more FIs in the sampled datasets. In contrast, since all the transactions in chess have the same length, we observe (not shown in figure) similar average numbers of FIs across the samplers. In this dataset, any swap operation performed by GMMT is actually a RBSO, and hence also the datasets sampled by GMMT preserve the BJDM. Similarly, the fact that the nodes in the graph representation of foodmart display high assortativity indicates that most of the swap operations of GMMT are RBSO.

Thanks to these results, we conclude that the BJDM captures important additional information about the data generation process. Therefore, using a null model that preserves it may lead to very different conclusions about the data generation process compared to one that does not. These results highlight, once more, how the choice of the null model by the user must be extremely deliberate.

VII. CONCLUSION

We introduce a novel null model for statistically assessing the results obtained from an observed transactional dataset, which preserves its Bipartite Joint Degree Matrix (BJDM). This property enforces, in addition to the dataset size, transaction lengths, and item supports, also the number of *caterpillars* of the bipartite graph corresponding to the observed dataset, which is a natural and important property that captures additional structure. We describe ALICE, a suite of two Markov-Chain-Monte-Carlo algorithms for sampling datasets from the null model. The results of our experimental evaluation show that ALICE scales well and that our null model allows to find different significant results than those from existing models.

Directions for future work include the development of even more descriptive null models (e.g., by preserving the number

[9]In this case, T = 2176, due to the prohibitive running time of GMMT.

of *butterflies* [36]), and of efficient procedures to sample from them, which is usually the challenging aspect.

Acknowledgment

This work is sponsored in part by NSF award IIS-2006765.

References

[1] E. Ferkingstad, L. Holden, and G. K. Sandve, "Monte Carlo null models for genomic data," *Statistical Science*, vol. 30, no. 1, pp. 59–71, 2015.

[2] R. T. Relator, A. Terada, and J. Sese, "Identifying statistically significant combinatorial markers for survival analysis," *BMC medical genomics*, vol. 11, no. 2, p. 31, 2018.

[3] J. Sese, A. Terada, Y. Saito, and K. Tsuda, "Statistically significant subgraphs for genome-wide association study," in *Statistically Sound Data Mining*, 2014, pp. 29–36.

[4] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2005.

[5] P. H. Westfall and S. S. Young, *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. Wiley-Interscience, 1993.

[6] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas, "Assessing data mining results via swap randomization," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 3, p. 14, 2007.

[7] N. D. Verhelst, "An efficient MCMC algorithm to sample binary matrices with fixed marginals," *Psychometrika*, vol. 73, no. 4, pp. 705–728, 2008.

[8] S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: Generalizing association rules to correlations," in *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '97, 1997, pp. 265–276.

[9] N. Megiddo and R. Srikant, "Discovering predictive association rules," in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, ser. KDD '98, 1998, pp. 274–278.

[10] J. Vreeken and N. Tatti, "Interesting patterns," in *Frequent pattern mining*. Springer, 2014, pp. 105–134.

[11] W. Hämäläinen, "StatApriori: an efficient algorithm for searching statistically significant association rules," *Knowledge and Information Systems*, vol. 23, no. 3, pp. 373–399, 2010.

[12] G. I. Webb, "Discovering significant patterns," *Machine Learning*, vol. 68, no. 1, pp. 1–33, 2007.

[13] J. Lijffijt, P. Papapetrou, and K. Puolamäki, "A statistical significance testing approach to mining the most informative set of patterns," *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 238–263, 2014.

[14] W. Hämäläinen and G. I. Webb, "A tutorial on statistically sound pattern discovery," *Data Mining and Knowledge Discovery*, vol. 33, no. 2, pp. 325–377, 2019.

[15] S. Hanhijärvi, "Multiple hypothesis testing in pattern discovery," in *International Conference on Discovery Science*. Springer, 2011, pp. 122–134.

[16] T. D. Bie, "Maximum entropy models and subjective interestingness: an application to tiles in binary databases," *Data Mining and Knowledge Discovery*, vol. 23, no. 3, pp. 407–446, dec 2010. [Online]. Available: https://doi.org/10.1007%2Fs10618-010-0209-3

[17] S. Pinxteren and T. Calders, "Efficient permutation testing for significant sequential patterns," in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 2021, pp. 19–27.

[18] A. Tonon and F. Vandin, "Permutation strategies for mining significant sequential patterns," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 1330–1335.

[19] S. Jenkins, S. Walzer-Goldfeld, and M. Riondato, "SPEck: Mining statistically-significant sequential patterns efficiently with exact sampling," *Data Min. Knowl. Discov.*, vol. 36, no. 4, 2022.

[20] S. Hanhijärvi, G. C. Garriga, and K. Puolamäki, "Randomization techniques for graphs," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, ser. SDM '09, 2009, pp. 780–791.

[21] M. Sugiyama, F. Llinares-López, N. Kasenburg, and K. M. Borgwardt, "Significant subgraph mining with multiple testing correction," in *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 2015, pp. 37–45.

[22] M. E. Silva, P. Paredes, and P. Ribeiro, "Network motifs detection using random networks with prescribed subgraph frequencies," in *International Workshop on Complex Networks*. Springer, 2017, pp. 17–29.

[23] S. Günnemann, P. Dao, M. Jamali, and M. Ester, "Assessing the significance of data mining results on graphs with feature vectors," in *2012 IEEE 12th International Conference on Data Mining*, 2012, pp. 270–279.

[24] M. Ojala, "Assessing data mining results on matrices with randomization," in *2010 IEEE International Conference on Data Mining*, 2010, pp. 959–964.

[25] M. Ojala, G. C. Garriga, A. Gionis, and H. Mannila, "Evaluating query result significance in databases via randomizations," in *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM)*, 2010, pp. 906–917.

[26] G. Cimini, T. Squartini, F. Saracco, D. Garlaschelli, A. Gabrielli, and G. Caldarelli, "The statistical physics of real-world networks," *Nature Reviews Physics*, vol. 1, no. 1, pp. 58–71, 2019.

[27] C. Greenhill, "Generating graphs randomly," *arXiv preprint arXiv:2201.04888*, 2022.

[28] F. Saracco, R. Di Clemente, A. Gabrielli, and T. Squartini, "Randomizing bipartite networks: the case of the world trade web," *Scientific reports*, vol. 5, no. 1, pp. 1–18, 2015.

[29] S. G. Aksoy, T. G. Kolda, and A. Pinar, "Measuring and modeling bipartite graphs with community structure," *Journal of Complex Networks*, vol. 5, no. 4, pp. 581–603, 2017.
[30] G. Amanatidis, B. Green, and M. Mihail, "Graphic realizations of joint-degree matrices," *arXiv preprint arXiv:1509.07076*, 2015.
[31] É. Czabarka, A. Dutle, P. L. Erdős, and I. Miklós, "On realizations of a joint degree matrix," *Discrete Applied Mathematics*, vol. 181, pp. 283–288, 2015.
[32] A. A. Boroojeni, J. Dewar, T. Wu, and J. M. Hyman, "Generating bipartite networks with a prescribed joint degree distribution," *Journal of complex networks*, vol. 5, no. 6, pp. 839–857, 2017.
[33] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. 20th Int. Conf. Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 487–499.
[34] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.
[35] S. Kim and S. Kirkland, "Gram mates, sign changes in singular values, and isomorphism," *Linear Algebra and its Applications*, vol. 644, pp. 108–148, 2022.
[36] S.-V. Sanei-Mehri, A. E. Sariyuce, and S. Tirthapura, "Butterfly counting in bipartite networks," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2150–2159.
[37] C. Low-Kam, C. Raïssi, M. Kaytoue, and J. Pei, "Mining statistically significant sequential patterns," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 488–497.

APPENDIX

A. Counterexample

We now show that preserving the two fundamental properties and the number of caterpillars does not imply that the BJDM is preserved.

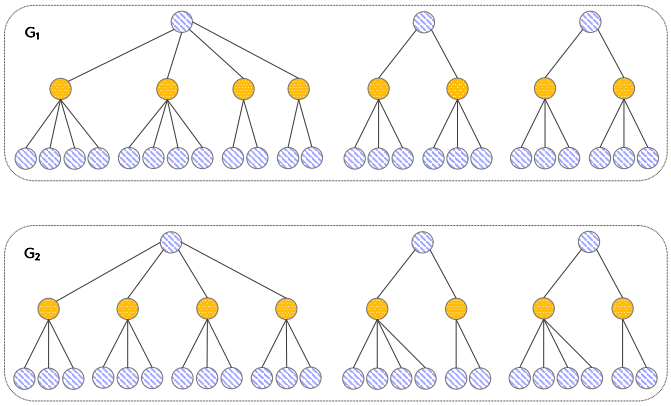

Fig. 4. Two bipartite graphs with the same degree distributions and the same number of caterpillars, but different BJDMs.

Each bipartite graph in Fig. 4 has three connected components, with a total of 27 left-hand side nodes (light-blue nodes) and 8 right-hand side nodes (yellow nodes). It is easy to see that the two graphs have the same degree distributions, and the same number of caterpillars (48). Indeed in the upper graph, the leftmost component contains 36 caterpillars, while each of the other two components contains 6 caterpillars, for a total of 48 caterpillars. Similarly, in the lower graph, the leftmost component contains 36 caterpillars, and the other two 6 caterpillars each. The two graphs have, nevertheless, different BJDMs: in the upper graph there are edges connecting nodes with degree 4 to nodes with degree 5 (top left), but the lower graph has no such edge.

B. Other Results

Figure 5 reports the mean number of FIs per length for ALICE-A, ALICE-B, and GMMT, in chess, foodmart, db-occ, BMS1, and BMS2. For the latter two datasets, we do not report results for GMMT, due to its prohibitive running time.

In chess, all the rows in the biadjacency matrix have the same sum, and thus, any swap operation performed by GMMT is a RSO, which, in turn, preserves the BJDM. As a result, the average numbers of FIs in the datasets generated by GMMT and ALICE are almost identical.

In foodmart, we observe that the product between the two marginals is close to the BJDM in terms of Frobenius norm, meaning that preserving the marginals *almost* preserves the BJDM. As a consequence, also in this case, the distribution of the numbers of FIs for GMMT is similar to that for ALICE.

In all datasets, we can see that the distribution of the number of FIs in the observed dataset is different from those obtained from the sampled datasets. In particular, the longer itemsets are, in general, less frequent in the sampled datasets than in

the original dataset. As an example, BMS2 contains many FIs of length larger than 3 (roughly 52% of the FIs), while most of the FIs in the datasets sampled by ALICE have length 1.

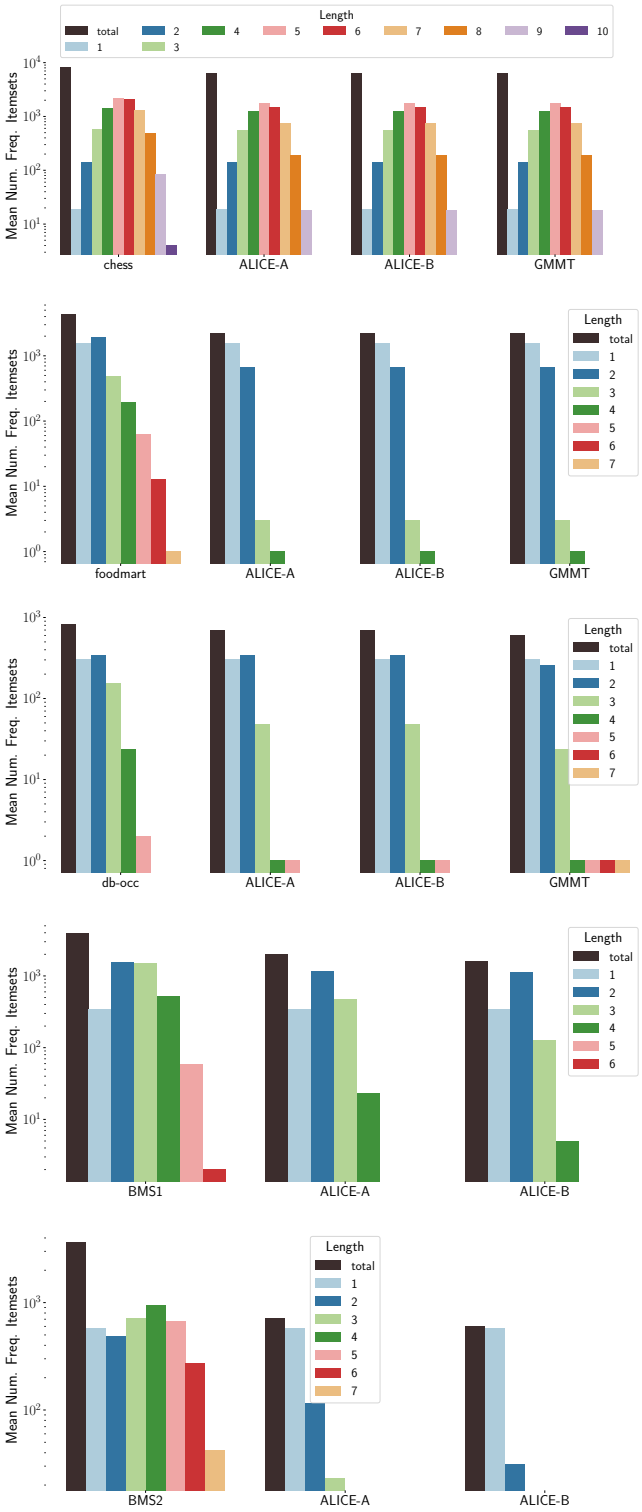

Fig. 5. Mean number of frequent itemsets per length for ALICE-A, ALICE-B, and GMMT, in chess, foodmart, db-occ, BMS1, and BMS2.

C. Extension to other settings

Previous work studied null models for testing the statistical significance of results obtained from other kinds of datasets, such as sequential datasets [17, 18, 19, 37]. We now show how to define new null models for sequential datasets to also preserve the BJDM.

Jenkins et al. [19] propose other two null models for sequential datasets. Most of what we just discussed can be applied, with minor modifications, to these null models.