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

Giulia Preti[1*], Gianmarco De Francisci Morales[1] and Matteo Riondato[2]

[1*]CENTAI, Corso Inghilterra 3, Turin, 10138, TO, Italy.
[2]Dept. of Computer Science Amherst College, 25 East Drive
Amherst, 01002 MA, USA.

*Corresponding author(s). E-mail(s): giulia.preti@centai.eu;
Contributing authors: gdfm@acm.org; mriondato@amherst.edu;

Abstract

We introduce novel null models for assessing the results obtained from observed binary transactional and sequence datasets, using statistical hypothesis testing. Our null models maintain more properties of the observed dataset than existing ones. Specifically, they preserve the Bipartite Joint Degree Matrix of the bipartite (multi-)graph corresponding to the dataset, which ensures that the number of caterpillars, i.e., paths of length three, is preserved, in addition to other properties considered by other models. We describe Alice, a suite of Markov-Chain Monte-Carlo algorithms for sampling datasets from our null models, 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.

Keywords: Hypothesis Testing, Markov Chain Monte Carlo Methods, Sequence Datasets, Significant Pattern Mining, Swap Randomization, Transactional Datasets

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

1 Introduction

Binary transactional datasets and sequence datasets are the object of study in several areas, from marketing to network analysis, to finance modeling, processing of satellite images, and more. In genomics, 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/sequence 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 (Ferkingstad et al, 2015; Relator et al, 2018; Sese et al, 2014).

The *statistical hypothesis testing* framework (Lehmann and Romano, 2022) is a very rigorous validation process for the results obtained from an observed dataset. Hypotheses about the results are formulated, and then tested by comparing the results (or appropriate statistics about them) to their distribution over the *null model*, i.e., a set of datasets enriched with a user-specified probability distribution (see Sect. 3.2), which 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* (Westfall and Young, 1993), 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 are deemed to 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 hypotheses 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 binary[1] transactional or sequence dataset by performing statistical hypothesis tests via resampling methods from a descriptive null model. Specifically, our contributions are the following.

- We introduce novel null models (Sect. 4 and Sect. 6.2) that preserve additional properties of the observed dataset than those preserved by existing null models (Gionis et al, 2007; Tonon and Vandin, 2019). Specifically, all datasets in our null models have the same *Bipartite Joint Degree Matrix (BJDM)* of the bipartite (multi-)graph corresponding to the observed dataset (Sect. 4.1 and 4.2). Maintaining the BJDM captures additional "structure" of the observed dataset: e.g., on transactional datasets, in addition to dataset size, transaction lengths, and item or itemset supports, the number of *caterpillars* in the observed dataset is also preserved (Lemma 3). We also explain why more natural properties, such as the supports of itemsets of length two on transactional datasets, are not as informative as one may think.
- We present ALICE,[2] a suite of Markov-Chain-Monte-Carlo algorithms for sampling datasets from our null models according to a user-specified distribution. ALICE-A (Sect. 5.1) 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. A second algorithm, ALICE-B, (Sect. 5.2) adapts the CURVEBALL approach (Verhelst, 2008; Carstens, 2015) to RSOs, to essentially perform multiple RSOs at every step, thus leading to faster mixing. Finally, ALICE-S samples from the null model for sequence datasets, using Metropolis-Hastings and a variant of RSOs, to take into account the fact that the bipartite graph corresponding to a sequence dataset is a *multi-graph*.
- 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.

The present article extends the conference version (Preti et al, 2022) in multiple ways, including:

- The extension to sequence datasets and the development of ALICE-S (Sect. 6) is entirely new. In addition to introducing a novel null model and algorithm, to the best of our knowledge, our work is the first to look at sequence datasets as bipartite multi-graphs, which is a generic representation that can be used in other works.

[1] In the rest of the work, we drop the attribute "binary": all datasets we refer to are binary.

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

- We give an explicit counterexample (Fig. 2) showing that preserving the number of caterpillars and other fundamental properties is not sufficient to preserve the BJDM, while the opposite is true (Sect. 4.2).
- We include a discussion of Gram mates (Kirkland, 2018; Kim and Kirkland, 2022), to explain why a model preserving the supports of itemsets of length two may not be very interesting.
- We add examples and figures to help the understanding of important concepts.

Outline

After discussing related work in Sect. 2, we focus the presentation on binary transactional datasets, with preliminaries (Sect. 1) also covering statistical hypothesis testing. Then we describe the null model for transactional datasets (Sect. 4), and then the two algorithms to sample datasets from this null model (Sect. 5). Covering first only transactional datasets allows us to discuss sequence datasets, the null model, and the specific algorithm for this case in Sect. 6. Our experimental evaluation and its results are presented in Sect. 7.

2 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 (Brin et al, 1997; Megiddo and Srikant, 1998). A long line of works also studies how to filter out uninteresting patterns, or directly mine *interesting* ones (Vreeken and Tatti, 2014). This direction is orthogonal to the study of the *statistical validity* of the results, which is our focus.

Many works concentrate on the case of *labeled* transactional datasets (Terada et al, 2015, 2013a,b; Pellegrina et al, 2019b; Hämäläinen, 2016; Pellegrina and Vandin, 2020; Papaxanthos et al, 2016; Minato et al, 2014; Llinares-López et al, 2015; Komiyama et al, 2017; Wu et al, 2016; Duivesteijn and Knobbe, 2011), where each transaction comes with a binary label. Most of these works use resampling-based approaches, as we do, but the very different nature of the studied tasks and data, as we study the *unlabeled* case, make them inapplicable to our problems. We refer to the tutorial by Pellegrina et al (2019a) for a detailed survey of the work done in *unlabeled datasets*, including resampling methods. The different nature of the data makes these approaches inapplicable to our case.

Most work has been on mining *significant frequent itemsets*, tiles, or association rules (Hämäläinen, 2010; Webb, 2007; Lijffijt et al, 2014). The survey by Hämäläinen and Webb (2019) presents many of these works in depth. The most relevant to ours are those by Gionis et al (2007) and Hanhijärvi (2011), 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 (2010) 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.

Assessing results obtained from sequence datasets has also generated interest (Pinxteren and Calders, 2021; Tonon and Vandin, 2019; Jenkins et al, 2022). We refer the interested reader for an in-depth discussion of related work in this area to (Jenkins et al, 2022, Sect. 2). To the best of our knowledge, we are the first to look at sequence datasets as bipartite *multi*-graphs, and to propose a null model that explicitly preserves properties of such multi-graphs. Our null model for sequence datasets preserves additional properties than the one introduced by Tonon and Vandin (2019), similarly to how our null model for transactional datasets preserves additional properties than the one by Gionis et al (2007), as indeed the Tonon and Vandin's model is essentially an adaptation of the Gionis et al's model to sequence datasets. Tonon and Vandin (2019) and Jenkins et al (2022) present other null models for sequence datasets. Extending these models to preserve the additional properties we consider is an interesting direction for future work.

Beyond binary transactional and sequence datasets, resampling methods for assessing data mining results have been proposed for graphs (Hanhijärvi et al, 2009; Sugiyama et al, 2015; Silva et al, 2017; Günnemann et al, 2012), real-valued and mixed-valued matrices (Ojala, 2010), and database tables (Ojala et al, 2010). 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 (Cimini et al, 2019; Bonifati et al, 2020; Greenhill, 2022; Akoglu and Faloutsos, 2009; Aksoy et al, 2017; Saracco et al, 2015; Karrer and Newman, 2011; Van Koevering et al, 2021; Fischer et al, 2015; Ritchie et al, 2017; Silva et al, 2017; Orsini et al, 2015; Tillman et al, 2019). The surveys by Cimini et al (2019), Bonifati et al (2020), and Greenhill (2022) give 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 (2015) 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. 4); thus our null model preserves more characteristics of the observed dataset. Aksoy et al (2017) proposes a method to generate bipartite networks that preserve also the clustering coefficient, which is not related to the BJDM. Amanatidis et al (2015) 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 (2015), but not for the purpose used in Alice. Boroojeni et al (2017) 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. 3.2), and Alice achieves these goals.

We are interested in sampling graphs (but really, datasets) from a set of graphs that preserve the same properties as some observed graph (i.e., dataset). This task is different from the problem of generating a graph from a random family, such as Erdős-Rényi graphs, stochastic block models, Kronecker graphs, preferential attachment graphs, and others, or fitting the parameters of such a family on the basis of one or more observed graphs.

3 Preliminaries

We now define the key concepts and notation used in this work. Table 1 summarizes the most important notation. Preliminaries for sequence datasets are deferred to Sect. 6.1.

3.1 Transactional Datasets

Let $\mathcal{I} \doteq \{a_1, \ldots, a_{|\mathcal{I}|}\}$ be a finite alphabet of *items*. W.l.o.g., we can assume $\mathcal{I} = \{1, \ldots, |\mathcal{I}|\}$. Any $A \subseteq \mathcal{I}$ is an *itemset*. A *transactional dataset*[3] $\mathcal{D}$ 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. Figure 1 (lower) shows a dataset of shopping baskets with three baskets (transactions) of length 6, 5, and 4, respectively.

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:

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

[3] From here to the end of Sect. 5, we only discuss *transactional* datasets, so we drop the attribute and just refer to them as "datasets".

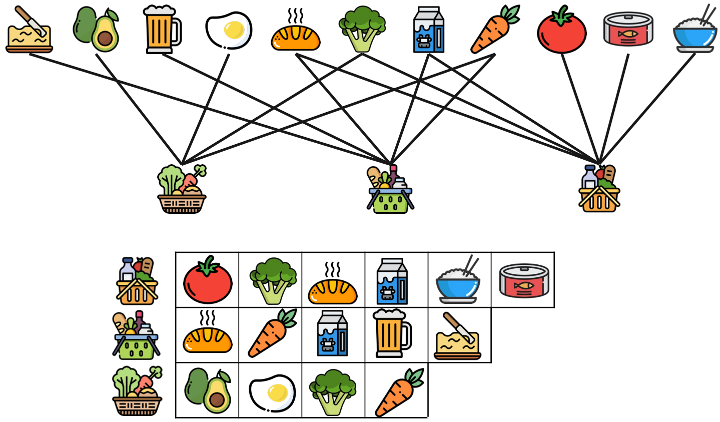


Fig. 1: A dataset of shopping baskets (lower) and the respective bipartite graph (upper).

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: $\mathsf{FI}_\theta(\mathcal{D}) \doteq \{A \subseteq \mathcal{I} : \sigma_{\mathcal{D}}(A) \geq \theta\}$ (Agrawal and Srikant, 1994). Given $\theta = 2$, for $\mathcal{D}$ in Fig. 1 (lower), $\mathsf{FI}_\theta(\mathcal{D})$ contains the itemsets { carrot }, { broccoli }, { bread }, { milk }, and { bread , milk }.

3.2 Null Models and Hypothesis Testing

The statistical hypothesis testing framework (Lehmann and Romano, 2022) 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 $\mathcal{Z}$.[4] Null models in previous works (Gionis et al, 2007; Bie, 2010) 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 $\mathring{\mathcal{D}}$;[5] and

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

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

- 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

$$H_0 \doteq \text{“} \mathop{\mathbb{E}}_{\mathcal{D}\sim\pi}\left[|\mathsf{FI}_\theta(\mathcal{D})|\right] = R_{\mathring{\mathcal{D}}}\text{”}.[6] \tag{1}$$

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.,

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

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 essentially impossible. E.g., for statistically-sound knowledge discovery tasks on sequence datasets, the exact distribution of test statistics is known only in very restricted cases (Pinxteren and Calders, 2021), while all other approaches use resampling (Tonon and Vandin, 2019; Jenkins et al, 2022). 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 (Westfall and Young, 1993). Let $\mathcal{D}_1, \dots, \mathcal{D}_T$ be T datasets *independently sampled* from $\mathcal{Z}$ according to π, then

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

Such *resampling methods*, of which the well-known bootstrap is also an instance, are often to be preferred to the explicit derivation of the statistics for multiple reasons:

- they are, in some sense, independent from the test being conducted, as the test statistic distribution (or better, the p-value) is estimated from the sampled datasets, as in (3);
- they leverage data-dependent distributional characteristics, which tend to result in higher statistical power; and
- they scale to high-dimensional settings.

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

In many knowledge discovery tasks, and in many applications such as during clinical trials for drug approvals (He et al, 2021), or in genomics studies (Goeman and Solari, 2014), one is interested in testing *multiple hypotheses.* For example, *significant itemset mining* (see Sect. 2) requires testing one hypothesis

$$H_0^A \doteq \text{“} \mathop{\mathbb{E}}_{\mathcal{D}\sim\pi}[\sigma_{\mathcal{D}}(A)] = \sigma_{\mathring{\mathcal{D}}}(A)\text{”}$$

for each itemset A.[7] When testing multiple hypotheses, i.e., all hypotheses in a class $\mathcal{H}$, one is interested in ensuring that the *Family-Wise Error Rate*, i.e., the probability of making *any* false discovery, is at most a user-specified acceptable threshold δ. Classic methods for controlling the FWER, such as the Bonferroni correction (Bonferroni, 1936), lack the *statistical power* to be useful in knowledge discovery settings, i.e., the probability that a *true* significant discovery is marked as such is very low, due to the large number $|\mathcal{H}|$ of hypotheses. *Resampling-based methods* (Westfall and Young, 1993) perform better for these tasks because they empirically estimate the distribution of the minimum p-value of the hypotheses in $\mathcal{H}$ by *sampling datasets from* $\mathcal{Z}$, and use this information to compute an *adjusted critical value* $\hat{\alpha}$.

For example, the Westfall-Young approach works as follows. Let $\mathcal{D}'_1, \ldots, \mathcal{D}'_T$ be T datasets *sampled independently* from $\mathcal{Z}$ according to π, and let

$$\check{p}_i \doteq \min_{h\in\mathcal{H}} p_{\mathcal{D}'_i,h} \tag{4}$$

be the minimum p-value, on $\mathcal{D}'$, of any hypothesis $h \in \mathcal{H}$. The *adjusted critical value* $\hat{\alpha}$ to which the p-values of the hypotheses are compared is

$$\hat{\alpha} \doteq \max\left\{\alpha : \frac{|\{\mathcal{D}'_i : \check{p}_i \leq \alpha\}|}{T} \leq \delta\right\} \ .$$

That is, $\hat{\alpha}$ is the largest $\alpha \in [0,1]$ such that the fraction of the T datasets $\mathcal{D}'_i$ whose minimum p-value $\check{p}$ is at most α is not greater than δ. Estimates computed as in (3) are used in place of the exact p-values in the r.h.s. of (4). Comparing the (estimated) p-value of each hypothesis in $\mathcal{H}$ to $\hat{\alpha}$ guarantees that the FWER is at most δ. Thus, efficiently drawing random datasets from $\mathcal{Z}$ according to π plays a key role in statistical hypothesis testing. Our goal in this work is to develop efficient methods to sample a dataset from $\mathcal{Z}$ according to π where $\mathcal{Z}$ is the set of datasets that, in addition to preserving the aforementioned three properties from $\mathring{\mathcal{D}}$, also preserve an additional important characteristic property that we describe in Sect. 4.2.

[7] This hypothesis is one of many kinds of hypotheses that can be tested by using the support as the test statistic.

3.3 Markov Chain Monte Carlo Methods

Alice follows the *Markov chain Monte Carlo (MCMC) method*, and uses the *Metropolis-Hastings (MH) algorithm* (Mitzenmacher and Upfal, 2005, 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 vertices V are known as *states* in this context. W.l.o.g., we can assume $V = \{1, 2, \dots, |V|\}$. For any state v, let $\Gamma(v)$ be the set of (out-)neighbors of v, i.e., the set of states u such that $(v, u) \in E$ (it holds $v \in \Gamma(v)$ if there is a self-loop). For any neighbor $u \in \Gamma(v)$, the weight $\mathsf{w}(v, u)$ of the edge (v, u) is strictly positive, and it holds $\sum_{u \in \Gamma(v)} \mathsf{w}(v, u) = 1$. In other words, there is a probability distribution ξ_v over $\Gamma(v)$ such that $\xi_v(u) = \mathsf{w}(v, u)$. Let W be the $|V| \times |V|$ matrix such that $W[v, u] = \mathsf{w}(v, u)$ if $(v, u) \in E$, and 0 otherwise.[8]

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

$$\min\left\{1, \frac{\phi(u)\xi_u(v)}{\phi(v)\xi_v(u)}\right\}, \tag{5}$$

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.

4 A More Descriptive Null Model

As discussed in Sect. 3.2, a good null model should preserve important characteristics of the observed dataset $\mathring{\mathcal{D}}$, and we mentioned the two fundamental properties that were the focus of previous work (Gionis et al, 2007; Bie, 2010). We now introduce a null model that preserves an additional property, and then show efficient methods to sample datasets from it.

4.1 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

[8] The strong-connectivity and aperiodicity of G, together with having $W[u, v] \geq 0$ iff $(u, v) \in E$, ensure that the Markov chain on V whose matrix of transition probabilities is W has a unique stationary distribution (Mitzenmacher and Upfal, 2005, Thm. 7.7).

Table 1: Table of symbols.

	Symbol	Description
Dataset	$\mathcal{I}$	Set of items
	S	Ordered list of itemsets
	$\mathcal{D}$	Dataset (bag of itemsets in the transactional case) (bag of sequences in the sequence case)
	$M_{\mathcal{D}}$	Binary matrix associated to the transactional dataset $\mathcal{D}$
	$\mathsf{mat}(\mathcal{D})$	Set of binary matrices associated to the transactional dataset $\mathcal{D}$
	$\mathsf{dat}(M)$	Transactional dataset whose binary matrix is M
	$\mathring{\mathcal{D}}$	Observed dataset
Bipartite (multi-)Graph	G	Bipartite (multi-)graph
	$L \cup R$	Set of left (L) and right (R) vertices of G
	E	Set of (multi-)edges of G
	$\mathcal{G}$	Set of bipartite multi-graphs
	$\Gamma(v)$	Set of nodes connected to v in G
	J_G	Bipartite Joint Degree Matrix (BJDM) of G
	$\mathsf{z}(G)$	Number of simple paths of length 3 (caterpillars) in G
	$\mathcal{M}$	Set of binary matrices of graphs with the same BJDM
Null Model	Π	Null model
	$\mathcal{Z}$	Set of datasets sharing some properties of $\mathring{\mathcal{D}}$
	π	Probability distribution over $\mathcal{Z}$
	$p_{\mathring{\mathcal{D}},H_0}$	p-value of a null hypothesis H_0 involving Π and $\mathring{\mathcal{D}}$

order of the transactions (i.e., of the rows) is arbitrary (Gionis et al, 2007, 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, i.e., that are *row-order agnostic* (Abuissa et al, 2023).

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[9] $(t, i) \in E$ iff transaction t contains the item i. Figure 1 (upper) depicts the bipartite graph corresponding to the dataset in the lower part of the figure. The left nodes (bottom nodes) model the three shopping baskets, while the right nodes (top nodes) represent the product bought. Different matrices M' and M'' corresponding to $\mathcal{D}$ are the biadjacency 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 $G_{\mathring{\mathcal{D}}}$.

[9] 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.

4.2 Preserving the Bipartite Joint Degree Matrix

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.,

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

The BJDM of the graph in Fig. 1 (upper) is the following:

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 2 & 2 \\ 2 & 3 \\ 3 & 3 \end{pmatrix}$$

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 $\mathsf{J}_{G_{\mathcal{D}}}$. 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*

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

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*

$$|\{u \in L : \mathsf{deg}(u) = i\}| = \frac{1}{i} \sum_{j=1}^{k_R} \mathsf{J}_G[i,j], \tag{7}$$

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 2 *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 2 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 $G_{\mathring{\mathcal{D}}}$ (Aksoy et al, 2017). Corollary 4 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 3, which shows that $\mathsf{z}(G)$ can be expressed through the BJDM.

Lemma 3 *It holds*

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

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

$$\mathsf{z}(G) = \sum_{(u,v)\in E} (\mathsf{deg}(u)-1)(\mathsf{deg}(v)-1) \enspace . \tag{8}$$

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. (8), and there are $\mathsf{J}_G[i,j]$ such edges. □

Corollary 4 *For any $\mathcal{D}$, the BJDM $\mathsf{J}_{G_{\mathcal{D}}}$ determines $\mathsf{z}(G_{\mathcal{D}})$.*

On the other hand, preserving the two fundamental properties and the number of caterpillars is not sufficient to preserve the BJDM: as we now show, 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 Fig. 2. Both bipartite graphs in Fig. 2 have three connected components each, with a total of 27 left-hand side nodes (light-blue, striped nodes) and 8 right-hand side nodes (yellow, dotted nodes). It is easy to see that the two graphs have the same degree distributions, and the same number of caterpillars (48). 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

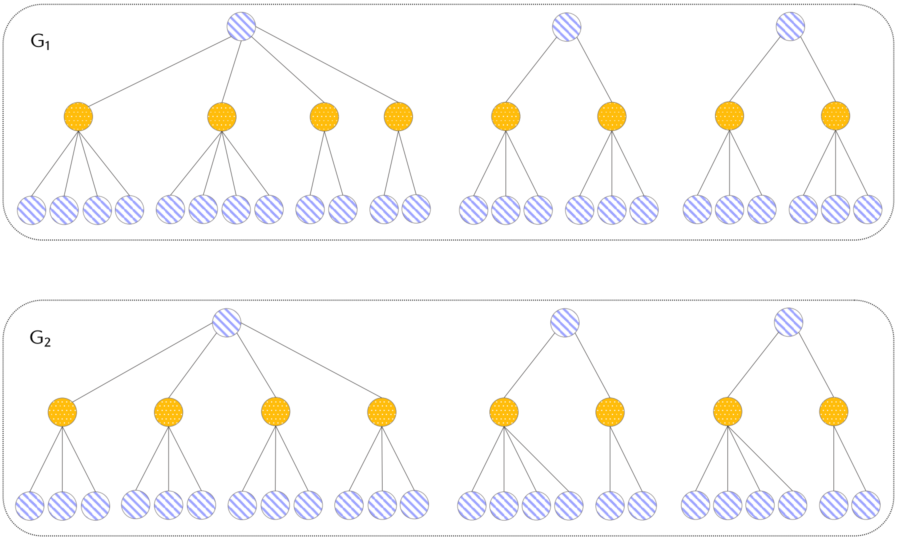
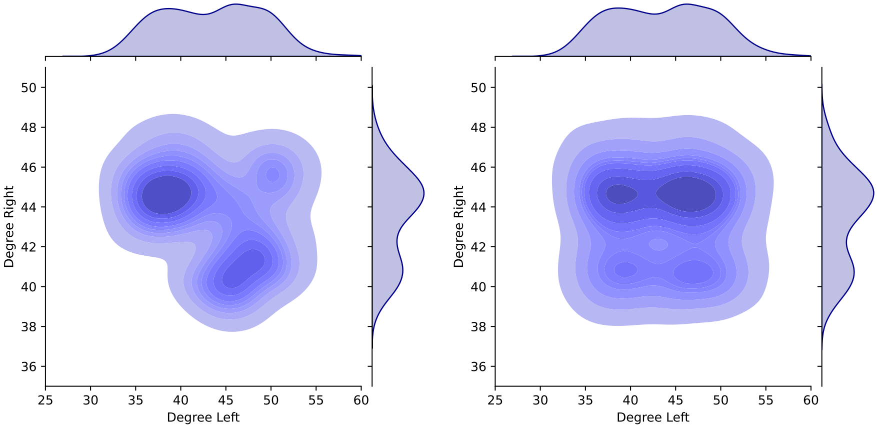


Fig. 2: Two bipartite graphs with the same degree distributions and the same number of caterpillars, but different BJDMs.

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.

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 $M_{\mathcal{D}} M_{\mathcal{D}}^{\top} = M_{\mathring{\mathcal{D}}} M_{\mathring{\mathcal{D}}}^{\top}$. Binary matrices A and B satisfying $AA^{\top} = BB^{\top}$ are known as *Gram mates* (Kirkland, 2018; Kim and Kirkland, 2022). Kirkland (2018, Corol. 1.1.1) shows an upper bound to the relative size of the set of Gram mates w.r.t. the set of all binary matrices, which decreases as the number of transactions in $\mathring{\mathcal{D}}$ and/or the number of items in $\mathcal{I}$ grow. While Kirkland (2018) and Kim and Kirkland (2022) construct infinite families of Gram mates, they observe that these families "possess a tremendous amount of structure" (Kirkland, 2018, Sect. 4), and it seems unlikely that such a structure would ever occur on matrices corresponding to real datasets, to the point that it is still an open question to determine whether a matrix A even admits *any* Gram mate, which would at least allow us to determine whether or not $|\mathcal{Z}| = 1$. On the other hand, if one can find at least one pair of Gram mates, Kim and Kirkland (2022, Sect. 5) give methods to build others (but possibly not *all*), thus if the open question is settled in a constructive way, one may be able to sample from (a subset of) $\mathcal{Z}$, if so interested.

Finally, we give an intuition about the properties that Alice preserves in addition to the fundamental ones. Preserving the BJDM of a bipartite graph means preserving the number of edges connecting two nodes with given degrees. This property implies, for instance, that the *assortativity* of the graph (Newman, 2002), i.e., the Pearson correlation coefficient of the vectors of degrees of nodes connected by an edge, is also maintained. Figure 3 shows an example of this property. Assume to have a dataset with an empirical joint degree distribution as in Fig. 3a. Alice preserves this joint degree distribution exactly. Conversely, by preserving only the two fundamental properties, we only preserve the marginal distributions as in Fig. 3b. In this latter case, the joint distribution is simply the product of the marginals, i.e., the marginals are assumed independent.



(a) Joint distribution under Alice, which preserves the BJDM and maintains the degree assortativity of the dataset.

(b) Joint degree distribution when preserving the two fundamental properties, where the left and right degree distributions are independent.

Fig. 3: Example of two different joint degree distributions of bipartite graphs with the same marginal degree distributions.

5 Sampling from the Null Model

We now present Alice-A and Alice-B, two algorithms for sampling datasets from the null model $\Pi = (\mathcal{Z}, \pi)$.

These algorithms take the MCMC approach with MH (see Sect. 3.3). Their 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. 4.1), and different datasets may have a different number of matrices in $\mathcal{M}$ associated to them. Alice-A and Alice-B take 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 $\mathcal{D}$. Abuissa et al (2023, Lemma 3) give an expression for the size $\mathsf{c}(\mathcal{D}) \doteq |\mathsf{mat}(\mathcal{D})|$ of $\mathsf{mat}(\mathcal{D})$. The correctness of the two algorithms relies on it so we report it here.

Lemma 5 (Abuissa et al, 2023, Lemma 3) *For any dataset $\mathcal{D} \in \mathcal{Z}$, let $\{\ell_1, \dots, \ell_{z_\mathcal{D}}\}$ 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 $\bar{T}_i = \{\tau_{i,1}, \dots, \tau_{i,r_i}\}$ 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*

$$\mathsf{c}(\mathcal{D}) = \prod_{i=1}^{z_\mathcal{D}} \underbrace{\binom{|T_i|}{|Q_{i,1}|, \dots, |Q_{i,r_i}|}}_{\textit{multinomial coefficient}} = \prod_{i=1}^{z_\mathcal{D}} \frac{|T_i|!}{\prod_{j=1}^{r_i} |Q_{i,j}|!} \quad . \tag{9}$$

Alice-A and Alice-B take as inputs π and the observed dataset $\mathring{\mathcal{D}}$. It uses MH (see Sect. 3.3) 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.

5.1 Alice-A: RSO-based Algorithm

In our first algorithm, Alice-A, the neighborhood structure over $\mathcal{M}$ is defined using *Restricted Swap Operations (RSOs)* (Czabarka et al, 2015, Sect. 2).

Definition 2 (Restricted Swap Operation (RSO)) Let M be the $|L| \times |R|$ biadjacency 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

$$M[a,c] = M[b,d] = 1 \land M[a,d] = M[b,c] = 0$$

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 $M'[b,d] = M[b,c]$.

Figure 4 (left) depicts a bipartite graph, where dotted nodes indicate left nodes, and striped nodes indicate right nodes. For ease of presentation, we use different colors to denote nodes with the same degree. A RSO in this graph is $(A,1),(B,5) \to (A,5),(B,1)$, because A and B satisfy condition C_{ab} and the edges $(A,5)$ and $(B,1)$ are not part of the graph. Figure 4 (right) shows the graph resulting from the application of the RSO. Dashed edges are edges involved in the RSO.

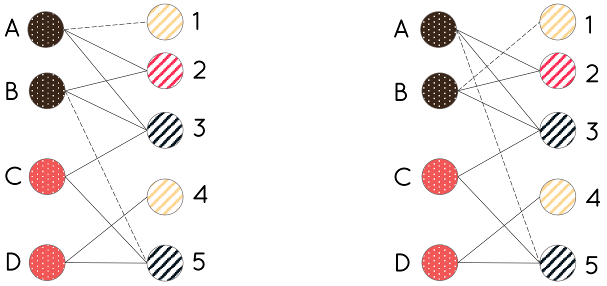


Fig. 4: The RSO denoted with dashed edges transforms the left graph into the right graph. Different patterns denote nodes on different sides of the graph, while different colors denote different degrees.

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 a 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 (Czabarka et al, 2015, Thm. 8).[10]

RSOs are just one of the many possible operations that make $\mathcal{Z}$ strongly connected. We discuss one such different operation in Sect. 5.2. Finding other operations to replace RSOs or to use in addition to RSOs is an 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

[10] The proof of (Czabarka et al, 2015, Thm. 8) must be adapted, in a straightforward way, to account for the fact that $\mathcal{M}$ contains biadjacency matrices of bipartite graphs.

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 a 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 a RSO may be very large, thus slowing down the process of moving on the graph. We briefly touch upon the convergence problem of this approach in Section 7. 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. (5)), or the number of pairs of different rows or 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": 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 $1 \leq m \leq |\mathcal{I}|$ (resp. $1 \leq n \leq |\mathcal{D}|$), let A_m be the set of row indices in M whose rows have sum m (resp. let B_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 $1 \leq m \leq |\mathcal{I}|$ with probability

$$\beta(m) = \binom{|A_m|}{2} \Bigg/ \sum_{j=1}^{|\mathcal{I}|} \binom{|A_j|}{2}, \tag{10}$$

and then we draw a pair (a, b) of *different* row indices in A_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

$$M[a, p] = M[b, q] \land M[a, q] = M[b, p] \land M[a, p] \neq M[a, q] \ .$$

We draw a pair (c, d) from $H_{a,b}$ uniformly at random. Then, either $(a, c), (b, d) \to (a, d), (b, c)$ or $(a, d), (b, c) \to (a, c), (b, d)$ is a 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 $1 \leq n \leq |\mathcal{D}|$

with probability

$$\gamma(n) = \binom{|B_n|}{2} \Big/ \sum_{j=1}^{|\mathcal{D}|} \binom{|B_j|}{2}, \tag{11}$$

and then we draw a pair (c,d) of different column indices in B_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

$$M[p,c] = M[q,d] \wedge M[p,d] = M[q,c] \wedge M[p,c] \neq M[p,d] \ .$$

We draw a pair (a,b) from $K_{c,d}$ uniformly at random. Then, either $(a,c),(b,d) \rightarrow (a,d),(b,c)$ or is also a RSO by construction, and we set M' to be $(a,d),(b,c) \rightarrow (a,c),(b,d)$ is a 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 $M' \neq M$. W.l.o.g., let $(a,c),(b,d) \rightarrow (a,d),(b,c)$ 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:

$$E_{\text{row}} \doteq \text{“rows } a \text{ and } b \text{ of } M \text{ have the same row sum } m\text{”;}$$
$$E_{\text{col}} \doteq \text{“columns } c \text{ and } d \text{ of } M \text{ have the same column sum } n\text{”.}$$

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

- if only E_{row} holds, then

$$\xi_M(M') = \frac{1}{2} \frac{1}{\sum_{i=1}^{|\mathcal{I}|} \binom{|R_i|}{2}} \frac{1}{|H_{a,b}|}; \tag{12}$$

- if only E_{col} holds, then

$$\xi_M(M') = \frac{1}{2} \frac{1}{\sum_{j=1}^{|\mathcal{D}|} \binom{|C_j|}{2}} \frac{1}{|K_{a,b}|}; \tag{13}$$

- 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. (12) and Eq. (13).

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. (5), 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

$$\phi(M) = \frac{\pi(\mathsf{dat}(M))}{\mathsf{c}(\mathsf{dat}(M))}, \tag{14}$$

where $\mathsf{c}(\mathsf{dat}(M))$ is from Eq. (9). 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 6 *Let $\mathcal{D} \in \mathcal{Z}$.* Alice-A *outputs $\mathcal{D}$ with probability $\pi(\mathcal{D})$.*

Proof Let $M \in \mathcal{M}$. From the correctness of MH we have that Alice-A samples M according to ϕ from Eq. (14). 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 $\mathcal{D}$. □

Algorithm 1 illustrates the main steps performed by Alice-A to sample a dataset in $\mathcal{Z}$. The algorithm receives in input a matrix $M \in \mathcal{M}$ and a number of swaps s sufficiently large for convergence. Previous works estimated that a number of steps in order of the number of 1s in M is sufficient. We will discuss this aspect in Section 7.

Algorithm 1 Alice

```
Require: Matrix $M \in \mathcal{M}$, Number of Swaps $s$
Ensure: Dataset $\mathcal{D}$ sampled from $\mathcal{Z}$ with probability $\pi(\mathcal{D})$
$\mathsf{c}(\mathsf{dat}(M)) \leftarrow$Equation (9)
$i \leftarrow 0$
while $i < s$ do
    $i \leftarrow i + 1$
    $\mathsf{out} \leftarrow$ flip a fair coin
    if $\mathsf{out}$ is heads then
        $a, b \leftarrow$ different row indices drawn u.a.r. such that $C_{ab}$ holds
        $c, d \leftarrow$ pair drawn u.a.r. from $H_{ab}$
    else
        $c, d \leftarrow$ different column indices drawn u.a.r. such that $C_{cd}$ holds
        $a, b \leftarrow$ pair drawn u.a.r. from $K_{cd}$
    $M' \leftarrow$ perform $(a,c),(b,d) \rightarrow (a,d),(b,c)$ on $M$
    $\mathsf{c}(\mathsf{dat}(M')) \leftarrow$Equation (9)
    $p \leftarrow$ random real number in $[0,1]$
    $a \leftarrow \min(1, \mathsf{c}(\mathcal{D})/\mathsf{c}(\mathcal{D}'))$
    if $p \le a$ then $M \leftarrow M'$
return $\mathsf{dat}(M)$
```

5.2 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 (Verhelst, 2008), which samples a matrix from the space of binary matrices with fixed row and column sums, to use RSOs. 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 U 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,U) *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

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

and

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

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.

Figure 5 (left) depicts a bipartite graph using the same style used in Fig. 4. Let $a = 1$ and $b = 2$, which are two right nodes with the same degree but different sets of neighbors. Then, $Z_a(M,b) = \{A, D\}$ and $Z_b(M,a) = \{B, G\}$. For $U = \{B, G\}$, the RBSO (a,b,U) generates the graph in Fig. 5 (right). Dashed edges are edges involved in the RBSO.

Any RBSO on a matrix M preserves J_M, and 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.

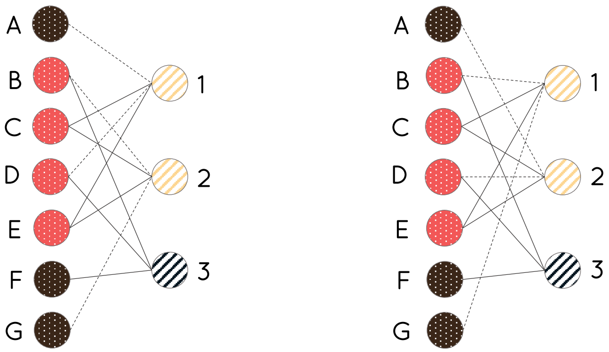


Fig. 5: The RBSO denoted with dashed edges transforms the left graph into the right graph. Different patterns denote nodes on different sides of the graph, while different colors denote different degrees.

Fact 7 *Let (a, b, U) 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 8 *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 7. □

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 9 *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. (10), 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 (Vitter, 1985), we obtain U 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. (11), 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,U) 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. (5)), and we now show that $\xi_M(M') = \xi_{M'}(M)$, i.e., the ratio is always 1.

Lemma 10 *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,U) 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 7, 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 $|U| = |Z_a(M,b)|$, 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 U 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,U) 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. 5.1, 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. (14). We thus have all the ingredients to use MH, and our description of Alice-B is complete. Note that Alice-B follows the same structure presented in Algorithm 1 but samples a rRBSO (a,b,U) at line 8:

```
$U \subset Z_a(M,b) \cup Z_b(M,a)$ s.t. $|U| = |Z_a(M,b)|$ obtained via reservoir sampling
```

and a cRBSO (c,d,U) at line 11:

```
$U \subset Z_c(M,d) \cup Z_d(M,c)$ s.t. $|U| = |Z_c(M,d)|$ obtained via reservoir sampling
```

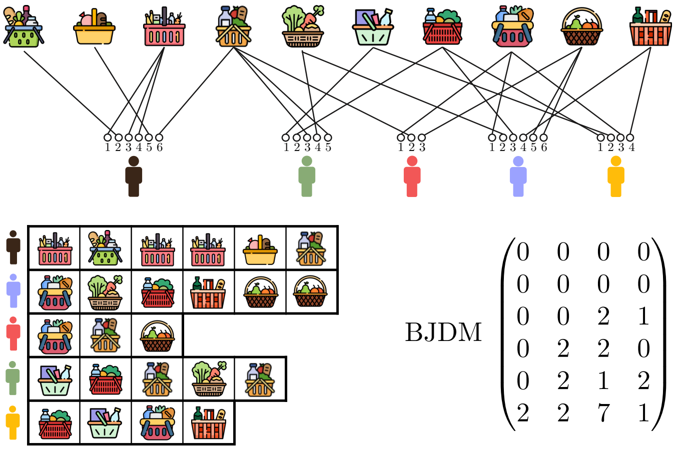


Fig. 6: Example of sequence dataset (lower left), corresponding multi-graph (top), and BJDM of the multi-graph (lower right).

6 Sequence Datasets

Previous work studied null models for testing the statistical significance of results obtained from other kinds of datasets, such as sequence datasets (Tonon and Vandin, 2019; Pinxteren and Calders, 2021; Jenkins et al, 2022; Low-Kam et al, 2013). We now define a new null model for sequence datasets to also preserve the BJDM, and we introduce a new algorithm Alice-S to sample from this null model.

6.1 Preliminaries on sequence datasets and multi-graphs

Let us start with a brief description of sequence datasets and related concepts. A *sequence* is a finite *ordered list* (or a *vector*) of not-necessarily-distinct itemsets, i.e., $S = \langle A_1, \ldots, A_\ell \rangle$ for some $\ell \geq 1$, with $A_i \subseteq \mathcal{I}$, $1 \leq i \leq \ell$. Itemsets A_i *participate* in S, and we denote this fact with $A_i \in S$, $1 \leq i \leq \ell$. The *length* $|S|$ of a sequence is the number of itemsets participating in it. A *sequence dataset* $\mathcal{D}$ is a finite bag of sequences, which, as elements of $\mathcal{D}$, are known as *seq-transactions*. The *support* $\sigma_{\mathcal{D}}(A)$ of an itemset A in $\mathcal{D}$ is the number of seq-transactions of $\mathcal{D}$ in which A participates. The *multi-support* $\rho_{\mathcal{D}}(A)$ of A in $\mathcal{D}$ is the number of times that A participates *in total* in the seq-transactions of $\mathcal{D}$. For example, in the dataset $\mathcal{D} = \{\langle A, B\rangle, \langle A, C, A\rangle, \langle B, C\rangle\}$, it holds that $\sigma_{\mathcal{D}}(A) = 2$ and $\rho_{\mathcal{D}}(A) = 3$.

A sequence dataset $\mathcal{D}$ can be represented as a bipartite *multi*-graph $G_{\mathcal{D}} = (L \cup R, E)$, where L are the seq-transactions of $\mathcal{D}$, and R is the *set* of all and only the itemsets with support at least 1 in $\mathcal{D}$, i.e., participating in at least one seq-transaction of $\mathcal{D}$. Each vertex $v \in L$ has degree[11] equal to the length of the

[11] In multi-graphs, the degree of a vertex v is still the number of edges incident to it, so each edge is counted, even if multiple edges connect v to the same vertex.

corresponding seq-transaction S_v of $\mathcal{D}$, i.e., $\mathsf{deg}(v) = |S_v|$. Each vertex $v \in L$ has $\mathsf{deg}(v)$ *ports*, which can be thought as the "locations" where the edges "connect" to v. The ports are arbitrarily labeled from 1 to $\mathsf{deg}(v)$. This labeling is needed to define the edge *multi*-set E as follows: there is an edge between $v \in L$ and $w \in R$ using port k of v iff the itemset B_w corresponding to the vertex w appears in position k of S_v, i.e., iff $S_v = \langle A_1, \ldots, A_{k-1}, B_w, A_{k+1}, \ldots, A_{|S_v|} \rangle$. We denote this edge as (v, k, w), thus E can also be thought as a set of such tuples. To the best of our knowledge, the one we just gave is the first description of sequence datasets as bipartite multi-graphs, which is somewhat surprising because representing transactional datasets as bipartite graphs has been a standard practice for a long time.

The definition of BJDM from Def. 1 is also valid for multi-graphs. Figure 6 shows an example of a sequence dataset (lower left), the corresponding multi-graph (top), and its BJDM (lower right).

6.2 BJDM-preserving null model for sequence datasets

Tonon and Vandin (2019) introduce a null model $\Pi = (\mathcal{Z}, \pi)$ for sequence datasets that can be seen as an adaptation of Gionis et al (2007)'s null model for transactional datasets. It preserves the following two properties of an observed dataset $\mathring{\mathcal{D}}$:

- the distribution of the seq-transaction lengths, i.e., for any seq-transaction length $\ell \in [1, \max_{S \in \mathring{\mathcal{D}}} |S|]$, any $\mathcal{D} \in \mathcal{Z}$ contains the same number of transactions of length ℓ as $\mathring{\mathcal{D}}$; and
- the multi-support of the itemsets participating in the seq-transactions of $\mathring{\mathcal{D}}$, i.e., for any $A \subseteq \mathcal{I}$ and $\mathcal{D} \in \mathcal{Z}$, $\rho_{\mathring{\mathcal{D}}}(A) = \rho_{\mathcal{D}}(A)$.

It should be evident how these two properties can be mapped to the two fundamental properties defined in Sect. 3.2 for transactional datasets, with the difference that itemsets participating in seq-transactions play the role that was of items in transactional datasets. Tonon and Vandin (2019) gave a MCMC algorithm to sample from this null model, while Jenkins et al (2022) gave an exact sampling algorithm.

The null model we define for sequence datasets preserves the BJDM of the multi-graph corresponding to the observed dataset. The following property can be derived in a way similar to that from Corol. 2, and confirms that preserving the BJDM also preserves the two above properties.

Corollary 11 *For any sequence dataset $\mathcal{D}$, the BJDM $\mathsf{J}_{G_{\mathcal{D}}}$ determines, for every $1 \leq j \leq \max_{S \in \mathcal{D}} |S|$, the number of seq-transactions in $\mathcal{D}$ with length j. Also, it determines, for every $1 \leq i \leq |\mathcal{D}|$, the number of itemsets with multi-support i in $\mathcal{D}$.*

On the other hand, it is not true that preserving the BJDM also preserves the number of caterpillars on multi-graphs, i.e., there is no equivalent of Lemma 3 and Corol. 4. The reason is that the BJDM does not encode information that allows the distinction between simple and multiple edges, i.e., the fact that a vertex with degree x may have any number of neighbors between 1 and x. It is also easy to come up with examples showing that it is not true that preserving the BJDM on multi-graphs preserves the number of *not-necessarily simple* paths of length three composed of three distinct edges. For instance, the multi-graph in Fig. 7 (left) includes the following 10 paths of length three: $(\beta,3,D)-(\beta,1,B)-(\alpha,3,B)$, $(\beta,3,D)-(\beta,1,B)-(\alpha,2,B)$, $(\beta,2,C)-(\beta,1,B)-(\alpha,3,B)$, $(\beta,2,C)-(\beta,1,B)-(\alpha,2,B)$, $(\beta,1,B)-(\alpha,3,B)-(\alpha,2,B)$, $(\beta,1,B)-(\alpha,2,B)-(\alpha,3,B)$, $(\beta,1,B)-(\alpha,3,B)-(\alpha,1,A)$, $(\beta,1,B)-(\alpha,2,B)-(\alpha,1,A)$, $(\alpha,3,B)-(\alpha,2,B)-(\alpha,1,A)$, and $(\alpha,2,B)-(\alpha,3,B)-(\alpha,1,A)$. The multi-graph on the right, which can be obtained by applying the mRSO $(\alpha,1,A),(\beta,1,B)\to(\alpha,1,B),(\beta,1,A)$ has the same BJDM but only six paths of length three: $(\alpha,1,B)-(\alpha,2,B)-(\alpha,3,B)$, $(\alpha,1,B)-(\alpha,3,B)-(\alpha,2,B)$, $(\alpha,2,B)-(\alpha,1,B)-(\alpha,3,B)$, $(\alpha,2,B)-(\alpha,3,B)-(\alpha,1,B)$, $(\alpha,3,B)-(\alpha,2,B)-(\alpha,1,B)$, and $(\alpha,3,B)-(\alpha,1,B)-(\alpha,2,B)$.

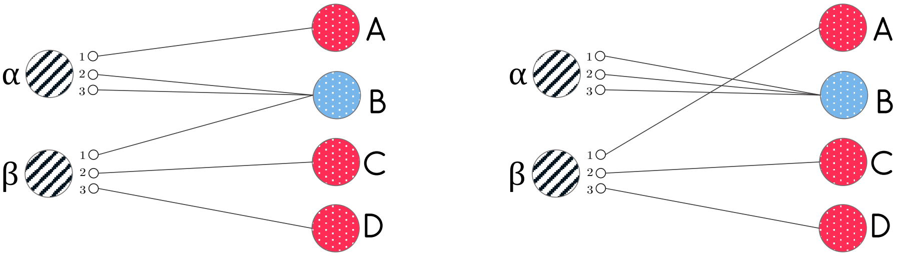


Fig. 7: Two bipartite multi-graphs with the same BJDM but different numbers of paths of length three. Different patterns denote nodes on different sides of the multi-graph, while different colors denote different degrees.

Nevertheless, since the multi-graph corresponding to a sequence dataset may actually be a simple graph, preserving the BJDM preserves more structure of the observed dataset than just the two fundamental properties, as we discussed for the counterexample from Fig. 2.

6.3 Alice-S: Alice for sequence datasets

We now discuss Alice-S, our algorithm for sampling from the BJDM-preserving null model for sequence dataset, which was defined in the previous section. Like the other members of the Alice family, Alice-S also takes the MCMC approach with MH. Its set of states though, is no longer the set $\mathcal{M}$ of biadjacency matrices, but a set $\mathcal{G}$ of bipartite multi-graphs defined as follows. Given the observed sequence dataset $\mathring{\mathcal{D}}$, let $G_{\mathring{\mathcal{D}}}=(L\cup R,E)$ be the multi-graph corresponding to it. $\mathcal{G}$ contains all and only the bipartite multi-graphs with node sets L and R, and with the same BJDM as $G_{\mathring{\mathcal{D}}}$. We remark that

$\mathcal{G}$ therefore includes also bipartite multi-graphs that are isomorphic to each other but differ for the ports to which the edges are connected, as such graphs represent different sequence datasets where the order of the itemsets in (some of) the sequences is shuffled.

The reason for not using the set $\mathcal{M}$ of biadjacency matrices as the state space of Alice-S is that a biadjacency matrix does not capture the entirety of the structure of a multi-graph corresponding to a sequence dataset, as it does not encode the information about the ports. It is important to understand that Alice-A could have been easily presented in Sect. 5.1 with a state space composed of graphs, rather than biadjacency matrices. We chose not to do that because the presentation of Alice-B greatly benefits from using matrices, although even in this case we could have used graphs, given that in the simple graph case, there is a bijection between bipartite graphs and biadjacency matrices. The flow and the notation in the following presentation of Alice-S are similar to the one for Alice-A, to highlight the many similarities between the two algorithms, but there are also many crucial differences.

We now define the concept of *multi-graph Restricted Swap Operation (mRSO)* as an operation that is applied to a multi-graph G to obtain another multi-graph G'.

Definition 4 (multi-graph Restricted Swap Operation (mRSO)) Let $G = (L \cup R, E)$ be a multi-graph, a and b be two non-necessarily distinct vertices in L, and c and d be two distinct vertices in R, such that there exist a port x of a and a port y of b such that

$$\{(a,x,c),(b,y,d)\} \subseteq E \wedge (\mathsf{deg}(a) = \mathsf{deg}(b) \vee \mathsf{deg}(c) = \mathsf{deg}(d)) \ .$$

The mRSO $(a,x,c),(b,y,d) \to (a,x,d),(b,y,c)$ is an operation that transforms G into the multi-graph $G' = (L \cup R, E')$ such that $E' = (E \setminus \{(a,x,c),(b,y,d)\}) \cup \{(a,x,d),(b,y,c)\}$.

It is easy to see that the multi-graph G' obtained by applying an mRSO to G is such that $\mathsf{J}_{G'} = \mathsf{J}_G$. There are zero or one mRSO between any two multi-graphs in $\mathcal{G}$. As an example, the mRSO $(\alpha,1,C),(\beta,4,E) \to (\alpha,1,E),(\beta,4,C)$ transforms the graph in Fig. 8 (left) to the graph on the right of such figure. Here, patterns denote the side of nodes on the graph, and colors denote different degrees. Dotted edges are the ones involved in the mRSO.

The neighborhood structure of the state space $\mathcal{G}$ is such that there is an edge from a multi-graph G to a multi-graph G' iff there is an mRSO transforming G into G'. In addition to these edges, there is a self-loop from each state to itself. This structure results in a strongly connected space, as can be seen by straightforwardly adapting (Czabarka et al, 2015, Thm. 8) in a way similar to what was done also for the bipartite simple graph case discussed in Sect. 5.1.

We now move to defining the neighbor sampling distribution ξ_G that is used to propose the next state $G' \in \Gamma(G)$ when the chain is at state $G \in \mathcal{G}$. As in Sect. 5.1, we first describe how to sample a neighbor of G, and then analyze the resulting distribution.

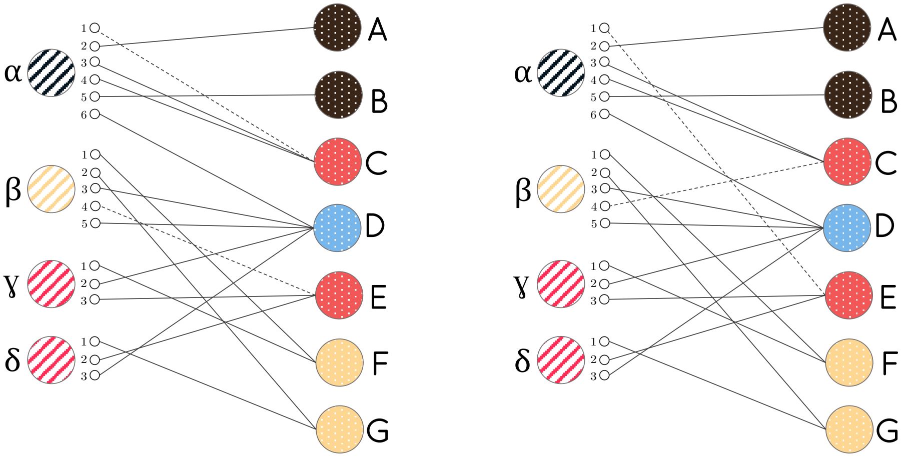


Fig. 8: Example of an mRSO. Dotted edges are edges involved in the mRSO, different patterns denote nodes on different sides of the graph, and different colors denote different degrees.

For any $1 \le m \le |R|$ (resp. $1 \le n \le |\mathring{\mathcal{D}}|$), let A_m (resp. B_n) be the subset of L (resp. of R) containing all and only the vertices with degree m in G (but really, in any $G' \in \mathcal{G}$). The first operation to sample a neighbor of G, is flipping a fair coin. If the outcome is *heads*, then we sample a degree m proportional to the number of pairs of *not-necessarily-distinct* vertices in L with degree m, i.e., we draw $1 \le m \le |R|$ with probability

$$\beta(m) = \binom{|A_m| + 1}{2} \Bigg/ \sum_{j=1}^{|R|} \binom{|A_j| + 1}{2}$$

and then we draw two vertices a and b by sampling uniformly at random, with replacement, from A_m. By sampling with replacement, we ensure that a and b may be the same vertex. Consider now the set

$$H_{a,b} \doteq \{((a,x,f),(b,y,g)) : (a,x,f) \in E \wedge (b,y,g) \in E \wedge f \neq g\}$$

of pairs of edges one incident to a and one incident to b and with different endpoints in R, and sample a pair $((a,x,c),(b,y,d))$ uniformly at random from this set.

If the outcome of the fair coin flip is *tails*, we first sample a degree $1 \le n \le |L|$ proportional to the number of pairs of *distinct* vertices in R with degree n, i.e., we draw $1 \le n \le |L|$ with probability

$$\gamma(n) = \binom{|B_n|}{2} \Bigg/ \sum_{j=1}^{|\mathring{\mathcal{D}}|} \binom{|B_j|}{2},$$

and then we sample two *distinct* vertices c and d from B_n uniformly at random without replacement. Let now (a, x, c) (resp. (b, y, d)) be an edge sampled uniformly at random from those incident to c (resp. to d).

The mRSO $(a, x, c), (b, y, d) \to (a, x, d), (b, y, c)$, when applied to G, gives the neighbor G' which is the proposed next state for the Markov chain.

We now analyze the distribution ξ_G over $\Gamma(G)$ induced by this procedure. Let $(a, x, c), (b, y, d) \to (a, x, d), (b, y, c)$ be the sampled mRSO, and let $G' \in \Gamma(G)$ be the multi-graph obtained by applying this mRSO to G. It must be $G' \neq G$. Recall that this mRSO is the only one leading from G to G'. Consider the following events:

$$E_\ell \doteq \text{“}\mathsf{deg}(a) = \mathsf{deg}(b) = m\text{”};$$
$$E_\mathrm{r} \doteq \text{“}\mathsf{deg}(c) = \mathsf{deg}(d) = n\text{”} \ .$$

There are three possible cases for $\xi_G(G')$:

- If only E_ℓ holds, then

$$\xi_G(G') = \frac{1}{2} \frac{1}{\sum_{j=1}^{|R|} \binom{|A_j|+1}{2}} \frac{1}{H_{a,b}} \ . \tag{15}$$

- If only E_r holds, then

$$\xi_G(G') = \frac{1}{2} \frac{1}{\sum_{j=1}^{|\mathring{\mathcal{D}}|} \binom{|B_j|}{2}} \frac{1}{n^2} \ . \tag{16}$$

- If both E_ℓ and E_r hold, then $\xi_G(G')$ is the sum of the r.h.s.'s of Eqs. (15) and (16).

It is easy to see that $\xi_G(G') = \xi_{G'}(G)$, which, like for ALICE-A, greatly simplifies the use of MH. As in that case, we define ϕ over $\mathcal{G}$ as

$$\phi(G) \doteq \frac{\pi(\mathsf{dat}(G))}{\mathsf{c}(\mathsf{dat}(G))},$$

where $\mathsf{c}(\mathsf{dat}(G))$ is still as in Eq. (9) because the same result also holds for sequence datasets under the null model we are considering (Abuissa et al, 2023, Lemma 4). We can then conclude on the correctness as follows, with the proof that is the same as that of Lemma 6.

Lemma 12 *Let $\mathcal{D} \in \mathcal{Z}$.* ALICE-S *outputs $\mathcal{D}$ with probability $\pi(\mathcal{D})$.*

Algorithm 2 reports the operations performed by ALICE-S to sample a sequence dataset in $\mathcal{Z}$. The algorithm receives in input the bipartite multi-graph $G \in \mathcal{G}$ corresponding to the observed sequence dataset $\mathring{\mathcal{D}}$ and a number of swaps s sufficiently large for convergence.

Algorithm 2 Alice-S

```
Require: Multi-Graph $G \in \mathcal{G}$, Number of Swaps $s$
Ensure: Sequence Dataset $\mathcal{D}$ sampled from $\mathcal{Z}$ with probability $\pi(\mathcal{D})$
c(dat($G$)) ←Equation (9)
$i \leftarrow 0$
while $i < s$ do
    $i \leftarrow i + 1$
    out ← flip a fair coin
    if out is heads then
        $a$, $b$ ← vertices in $L$ drawn u.a.r. such that deg($a$) = deg($b$)
        $(a, x, c)$, $(b, y, d)$ ← pair drawn u.a.r. from $H_{ab}$
    else
        $c$, $d$ ← different vertices in $R$ drawn u.a.r. such that deg($c$) = deg($d$)
        $(a, x, c)$, $(b, y, d)$ ← edges drawn u.a.r. from those incident to $c$,$d$
    $G'$ ← perform $(a, x, c), (b, y, d) \rightarrow (a, x, d), (b, y, c)$ on $G$
    c(dat($G'$)) ←Equation (9)
    $p$ ← random real number in $[0, 1]$
    $a \leftarrow \min(1, \mathsf{c}(\mathcal{D})/\mathsf{c}(\mathcal{D}'))$
    if $p \le a$ then $G \leftarrow G'$
return dat($G$)
```

We leave for future work the development of a Curveball-like approach for sampling sequence datasets from the null model. Jenkins et al (2022) propose other two null models for sequence datasets. Extending these null models to also preserve the BJDM is an interesting direction for future work.

7 Experimental Evaluation

We now report on the results of our experimental evaluation of Alice-A, Alice-B, and Alice-S. 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 transactional datasets and six real-world sequence datasets,[12] listed in Table 2. 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**, **BMS2**, and **FIFA** 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; **db-occ** includes user occupations taken from dbpedia; **SIGN** is a dataset of sign language utterance; **LEVIATHAN** and **BIBLE** are sentence datasets created

[12] From www.philippe-fournier-viger.com/spmf/index.php and http://konect.cc/networks.

Table 2: Datasets statistics: num. of transactions, num. of items, sum of transaction lengths, avg. transaction length, density, and number of caterpillars.

Dataset	Trans. Num	Item Num	Sum Trans. Lengths	AVG Trans. Length	Density	Num. Cater.
iewiki	137	558	651	4.752	0.0085	10K
kosarak	3000	5767	23664	7.888	0.0014	88M
chess	3196	75	118252	37.000	0.4933	9.93B
foodmart	4141	1559	18319	4.424	0.0028	954K
db-occ	10000	8984	19729	1.973	0.0002	7.5M
BMS1	59602	497	149639	2.511	0.0051	1.13B
BMS2	77512	3340	358278	4.622	0.0014	1.96B
retail	88162	16470	908576	10.306	0.0006	60B
SIGN	730	269	76646	104.994	0.3903	696M
LEVIATHAN	5834	9027	400336	68.621	0.0076	22B
FIFA	20450	2992	1502634	73.478	0.0246	159B
BIKE	21078	69	327844	15.554	0.2254	5.88B
BIBLE	36369	13907	1610501	44.282	0.0032	259B
BMS1	59601	499	358877	6.021	0.0121	1.13B

from the novel Leviathan by Thomas Hobbes (1651) and the Bible, respectively; and in **BIKE** each sequence indicate the bike sharing stations where a bike was parked in Los Angeles over time.

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 (Gionis et al, 2007), which is a swap randomization algorithm that samples from the null model that only maintains the two fundamental properties. The sampler GMMT-S is a variant of the SelfLoop version of GMMT that preserves the left and right degree sequences of the bipartite multi-graph representation of the observed sequence dataset. 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 (2007). The mixing time, i.e., the number of steps needed for the 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 9 reports this quantity for chess (upper left), foodmart (upper right), BMS2 (lower left), and retail (lower 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

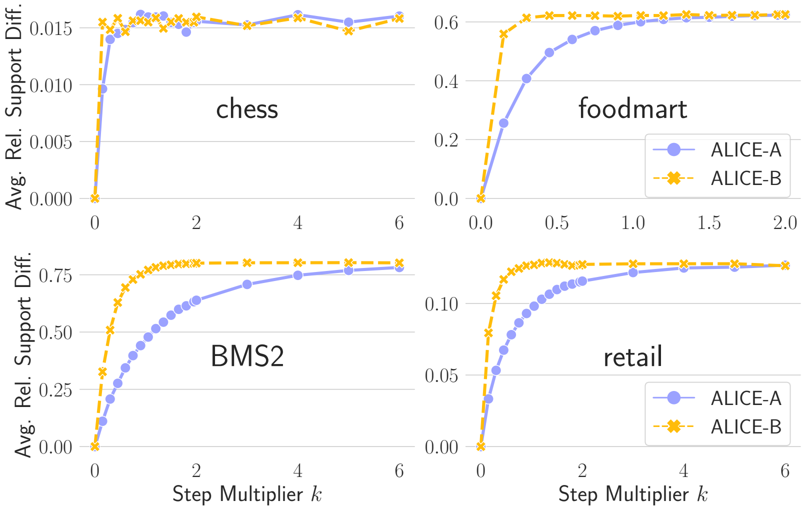


Fig. 9: Convergence of the samplers increasing the step number multiplier k, for chess (upper left), foodmart (upper right), BMS2 (lower left), and retail (lower right).

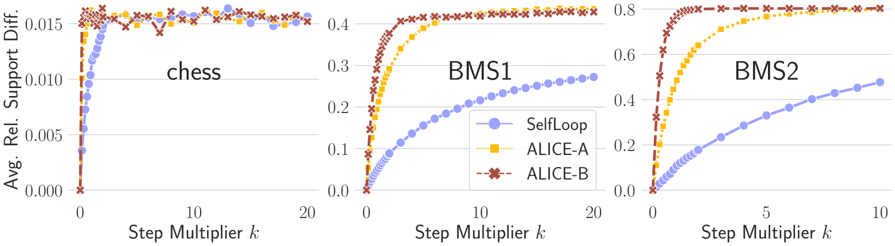


Fig. 10: Convergence of Alice-A and Alice-B vs SelfLoop increasing the step number multiplier k, for chess (left), BMS1 (middle), and BMS2 (right).

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 U (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.

Figure 10 compares the ARSD values obtained by Alice with those measured in the states of the chain traversed by the naïve approach introduced in Section 5.1 (called SelfLoop in the figure). Recall that, at each step, this approach draws two pairs (a, c) and (b, d) of row-column indices uniformly at random, and moves to the next state if $(a, c), (b, d) \rightarrow (a, d), (b, c)$ is a RSO. Especially for larger datasets, we observe that SelfLoop moves slowly in the state space, which prevents the ARSD from stabilizing even after $10w$ steps. As a result, a large number of steps is required to increase the likelihood of convergence, thus rendering SelfLoop impractical for use. In fact, the running time increases with the number of steps. In BMS1, for example, the ARSD for Alice-B stabilizes around $k = 4$, with Alice-B taking roughly 17s to perform the $4w$ steps. In contrast, SelfLoop takes 397s to perform the $10w$ steps.

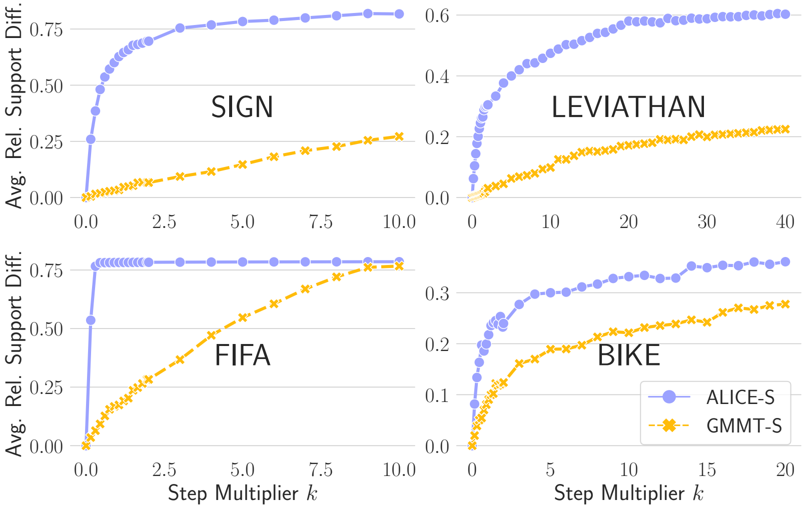


Fig. 11: Convergence of Alice-S and GMMT-S increasing the step number multiplier k, for SIGN (upper left), LEVIATHAN (upper right), FIFA (lower left), and BIKE (lower right).

We notice a similar behavior in Figure 11, which illustrates the convergence of Alice-S and GMMT-S for the sequence datasets SIGN (upper left), LEVIATHAN (upper right), FIFA (lower left), and BIKE (lower right). In this case, $w = \mathsf{E}$, i.e. the number of edges in the multi-graph corresponding to the dataset. In SIGN and FIFA the ARSD stabilizes before $k = 3$ for Alice-S, whereas for GMMT-S it stabilizes only in FIFA. In BIKE and LEVIATHAN both samplers move slowly, and thus convergence is reached after almost $20w$ and $30w$ steps, respectively.

Scalability. To study the scalability of Alice, we create synthetic datasets with increasing number of transactions and average transaction length 25, by using the IBM Quest generator (Agrawal and Srikant, 1994): five datasets with

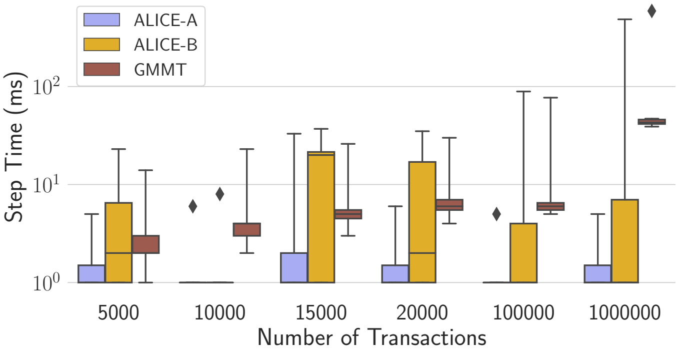


Fig. 12: Step times of the samplers in the synthetic datasets.

100 items and 5k, 10k, 15k, 20k, and 100k transactions, and one dataset with 10k items and 1M transactions. For each sampler for transactional datasets, we perform 10k steps and compute the distribution of step times, reported in Fig. 12 (log values). 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 U: the time required to compute $\mathsf{c}(\mathcal{D})$ is larger for cRBSO, and it grows with the size of U.

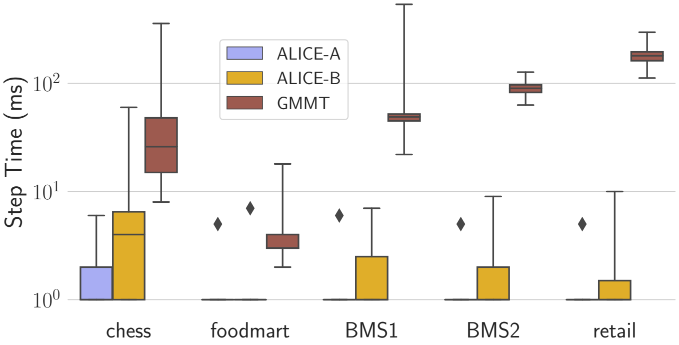


Fig. 13: Step times of the samplers in the real datasets (log times).

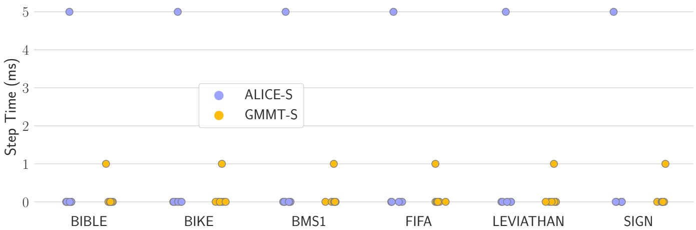


Fig. 14: Step times of the samplers in the real sequence datasets.

Figure 13 reports the distribution of the time required to perform a step for each sampler in each transactional dataset. 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 U 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 U is often 1. An algorithmic improvement in the bookkeeping due to the size of U would results in better performance of Alice-B, as mentioned above.

Figure 14 shows the distribution of step times for Alice-S and GMMT-S in the sequence datasets. The performance of Alice-S is comparable with that of Alice-A, as they follow a similar approach to sample the swap operations to perform. The median step time is always < 1, and the algorithm takes at most 5ms to perform a step. The step times of GMMT-S are far lower than its counterpart for transactional datasets, because this algorithm does not require bookkeeping to compute the transition acceptance probability. we recall that also in this case the running time of GMMT-S is not really comparable with that of our algorithm because they sample from different null models.

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.[13] 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 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 3 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

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

[14] For chess and BMS1, $T = 2176$, due to the prohibitive running time of GMMT.

Table 3: 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[14] 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
BMS1 θ = 0.001	3991	Alice-A	1998	4.6E-4
		Alice-B	1609	4.6E-4
		GMMT	1800	4.6E-4

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 15 and Figure 16 show the distribution of the number of FIs of different lengths in the original datasets, and the average of the same quantity over the datasets sampled by the different samplers. For BMS2 and retail we do not report results for GMMT, due to its prohibitive running time. 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. 15, i) datasets sampled from all null models have fewer FIs than the observed one. Conversely, in kosarak (Fig. 15, ii) 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 three,

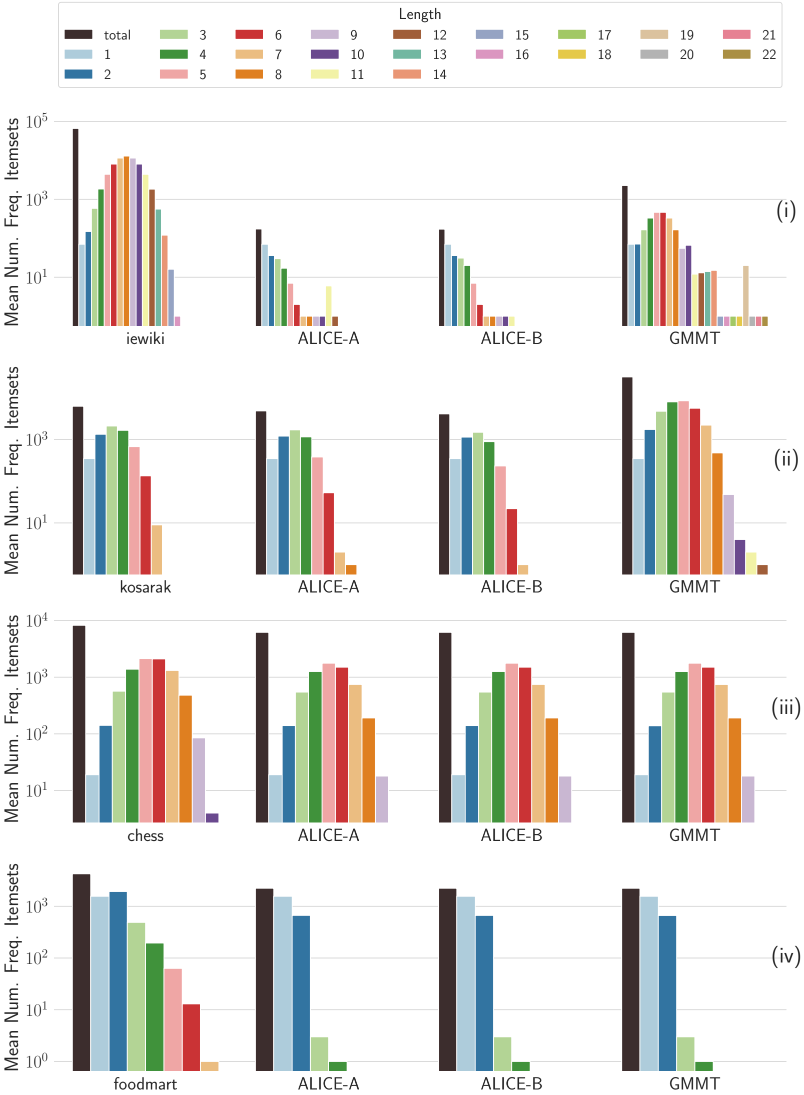


Fig. 15: Mean number of frequent itemsets per length for ALICE-A, ALICE-B, and GMMT, in iewiki (i), kosarak (ii), chess (iii), and foodmart (iv).

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 (Fig. 15, iii) 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 (Fig. 15, iv) display high

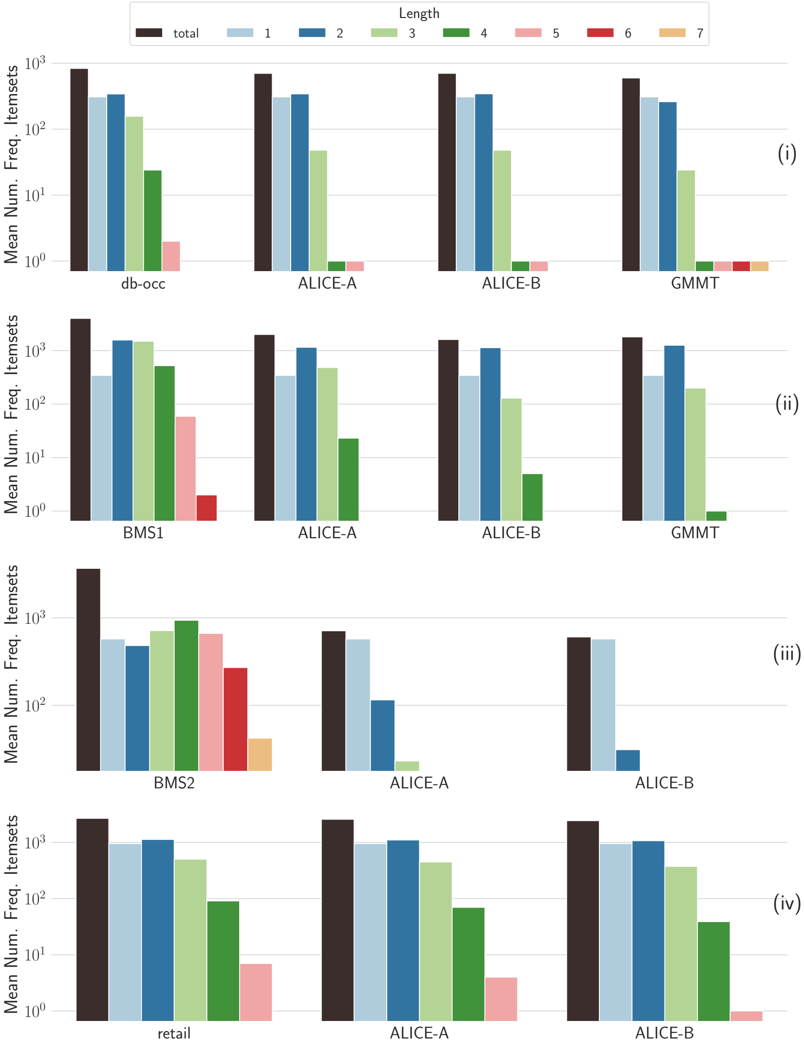


Fig. 16: Mean number of frequent itemsets per length for Alice-A, Alice-B, and GMMT (when available), in db-occ (i), BMS1 (ii), BMS2 (iii), and retail (iv).

assortativity indicates that most of the swap operations of GMMT are RBSO. In fact, when the product between the two marginals is close to the BJDM in terms of Frobenius norm, 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.

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

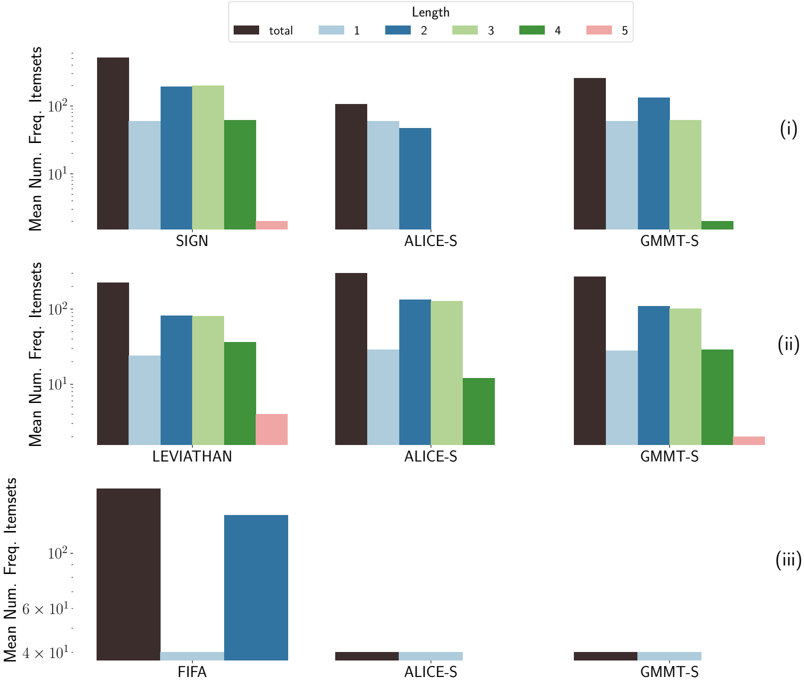


Fig. 17: Mean number of frequent itemsets per length for Alice-S and GMMT-S, in SIGN (i), LEVIATHAN (ii), and FIFA (iii).

datasets than in the original dataset. As an example, BMS2 (Fig. 16, iii) contains many FIs of length larger than three (roughly 52% of the FIs), while most of the FIs in the datasets sampled by Alice have length one.

Figure 17 and Figure 18 present the distribution of frequent sequential itemsets of different lengths in the original sequence datasets, and the average of the same quantity over the datasets sampled by Alice-S and GMMT-S. The frequency thresholds used are taken from Tonon and Vandin (2019): 0.4 for SIGN, 0.15 for LEVIATHAN, 0.275 for FIFA, 0.025 for BIKE, 0.1 for BIBLE, and 0.002 for BMS1. The number of samples extracted is always 4352 and the number of steps performed by Alice-S is $10w$, while it is $50w$ for GMMT-S. Also in this case, w is the number of edges in the multi-graph corresponding to the dataset. Similarly to the transactional dataset case, we tend to observe frequent itemsets of larger size in the datasets sampled by GMMT-S, except in the case of few frequent itemsets in the original dataset (e.g. FIFA and BIKE). In such cases, only trivial itemsets are frequent, and their frequencies tend to be preserved by preserving the two fundamental properties.

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,

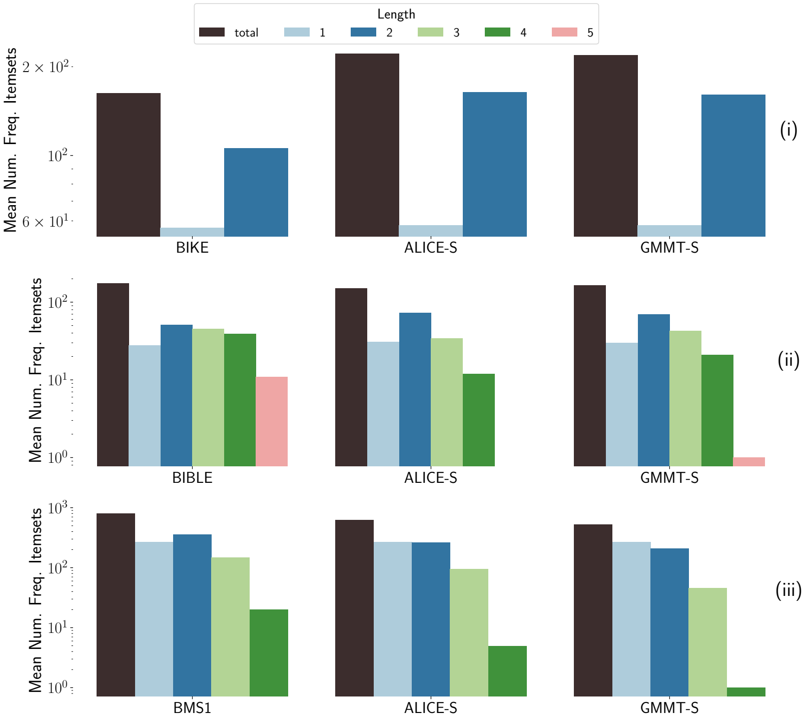


Fig. 18: Mean number of frequent itemsets per length for Alice-S and GMMT-S, in BIKE (i), BIBLE (ii), and BMS1 (iii).

once more, how the choice of the null model by the user must be extremely deliberate.

8 Conclusion

We introduced a novel null model for statistically assessing the results obtained from an observed transactional or sequence dataset, preserving its Bipartite Joint Degree Matrix (BJDM). On transactional datasets, maintaining this property enforces, in addition to the dataset size, transaction lengths, and item supports, also the preservation of 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 Markov-Chain-Monte-Carlo algorithms for sampling datasets from the null models. The results of our experimental evaluation show that Alice scales well and that, when testing results w.r.t. our null models, different results are marked as significant than when using existing null models.

A good direction for future work includes a rigorous theoretical analysis and/or experimental evaluation of the trade-offs between the time taken to perform a single step and the mixing time of the Markov chain when using different neighbor sampling distribution. Towards making statistically-sound knowledge discovery a reality, we also suggest the development of even more descriptive null models (e.g., by preserving the number of *butterflies* (Sanei-Mehri et al, 2018)), and of efficient procedures to sample from them, which is usually the challenging aspect. Another interesting direction is proposing null models for real-valued transactional datasets, such as those used for high-utility itemsets mining.

Acknowledgments

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

References

Abuissa M, Lee A, Riondato M (2023) ROhAN: Row-order agnostic null models for statistically-sound knowledge discovery. Data Mining and Knowledge Discovery https://doi.org/10.1007/s10618-023-00938-4

Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. In: Proc. 20th Int. Conf. Very Large Data Bases. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, VLDB '94, pp 487–499

Akoglu L, Faloutsos C (2009) Rtg: A recursive realistic graph generator using random typing. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, pp 13–28

Aksoy SG, Kolda TG, Pinar A (2017) Measuring and modeling bipartite graphs with community structure. Journal of Complex Networks 5(4):581–603

Amanatidis G, Green B, Mihail M (2015) Graphic realizations of joint-degree matrices. arXiv preprint arXiv:150907076

Bie TD (2010) Maximum entropy models and subjective interestingness: an application to tiles in binary databases. Data Mining and Knowledge Discovery 23(3):407–446. https://doi.org/10.1007/s10618-010-0209-3

Bonferroni CE (1936) Teoria statistica delle classi e calcolo delle probabilità. Pubblicazioni del Regio Istituto Superiore di Scienze Economiche e Commerciali di Firenze 8:3–62

Bonifati A, Holubová I, Prat-Pérez A, et al (2020) Graph generators: State of the art and open challenges. ACM Computing Surveys (CSUR) 53(2):1–30

Boroojeni AA, Dewar J, Wu T, et al (2017) Generating bipartite networks with a prescribed joint degree distribution. Journal of complex networks 5(6):839–857

Brin S, Motwani R, Silverstein C (1997) Beyond market baskets: Generalizing association rules to correlations. In: Proceedings of the 1997 ACM SIGMOD international conference on Management of data, SIGMOD '97, pp 265–276

Carstens CJ (2015) Proof of uniform sampling of binary matrices with fixed row sums and column sums for the fast curveball algorithm. Physical Review E 91(4):042,812

Cimini G, Squartini T, Saracco F, et al (2019) The statistical physics of real-world networks. Nature Reviews Physics 1(1):58–71

Czabarka É, Dutle A, Erdős PL, et al (2015) On realizations of a joint degree matrix. Discrete Applied Mathematics 181:283–288

Duivesteijn W, Knobbe A (2011) Exploiting false discoveries–statistical validation of patterns and quality measures in subgroup discovery. In: 2011 IEEE 11th International Conference on Data Mining, IEEE, pp 151–160

Ferkingstad E, Holden L, Sandve GK (2015) Monte Carlo null models for genomic data. Statistical Science 30(1):59–71

Fischer R, Leitao JC, Peixoto TP, et al (2015) Sampling motif-constrained ensembles of networks. Physical review letters 115(18):188,701

Gionis A, Mannila H, Mielikäinen T, et al (2007) Assessing data mining results via swap randomization. ACM Transactions on Knowledge Discovery from Data (TKDD) 1(3):14

Goeman JJ, Solari A (2014) Multiple hypothesis testing in genomics. Statistics in medicine 33(11):1946–1978

Greenhill C (2022) Generating graphs randomly. arXiv preprint arXiv:220104888

Günnemann S, Dao P, Jamali M, et al (2012) Assessing the significance of data mining results on graphs with feature vectors. In: 2012 IEEE 12th International Conference on Data Mining, pp 270–279, https://doi.org/10.1109/ICDM.2012.70

Hämäläinen W (2010) StatApriori: an efficient algorithm for searching statistically significant association rules. Knowledge and Information Systems 23(3):373–399. https://doi.org/10.1007/s10115-009-0229-8

Hämäläinen W (2016) New upper bounds for tight and fast approximation of Fisher's exact test in dependency rule mining. Computational Statistics & Data Analysis 93:469–482

Hämäläinen W, Webb GI (2019) A tutorial on statistically sound pattern discovery. Data Mining and Knowledge Discovery 33(2):325–377

Hanhijärvi S (2011) Multiple hypothesis testing in pattern discovery. In: International Conference on Discovery Science, Springer, pp 122–134

Hanhijärvi S, Garriga GC, Puolamäki K (2009) Randomization techniques for graphs. In: Proceedings of the 2009 SIAM International Conference on Data Mining, SDM '09, pp 780–791, https://doi.org/10.1137/1.9781611972795.67, https://epubs.siam.org/doi/pdf/10.1137/1.9781611972795.67

He J, Li F, Gao Y, et al (2021) Resampling-based stepwise multiple testing procedures with applications to clinical trial data. Pharmaceutical Statistics 20(2):297–313

Jenkins S, Walzer-Goldfeld S, Riondato M (2022) SPEck: Mining statistically-significant sequential patterns efficiently with exact sampling. Data Min Knowl Discov 36(4)

Karrer B, Newman ME (2011) Stochastic blockmodels and community structure in networks. Physical review E 83(1):016,107

Kim S, Kirkland S (2022) Gram mates, sign changes in singular values, and isomorphism. Linear Algebra and its Applications 644:108–148

Kirkland S (2018) Two-mode networks exhibiting data loss. Journal of Complex Networks 6(2):297–316

Komiyama J, Ishihata M, Arimura H, et al (2017) Statistical emerging pattern mining with multiple testing correction. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 897–906

Lehmann EL, Romano JP (2022) Testing Statistical Hypotheses, 4th edn. Springer

Lijffijt J, Papapetrou P, Puolamäki K (2014) A statistical significance testing approach to mining the most informative set of patterns. Data Mining and Knowledge Discovery 28(1):238–263. https://doi.org/10.1007/s10618-012-0298-2

Llinares-López F, Sugiyama M, Papaxanthos L, et al (2015) Fast and memory-efficient significant pattern mining via permutation testing. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 725–734

Low-Kam C, Raïssi C, Kaytoue M, et al (2013) Mining statistically significant sequential patterns. In: 2013 IEEE 13th International Conference on Data Mining, IEEE, pp 488–497

Megiddo N, Srikant R (1998) Discovering predictive association rules. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, KDD '98, pp 274–278

Minato Si, Uno T, Tsuda K, et al (2014) A fast method of statistical assessment for combinatorial hypotheses based on frequent itemset enumeration. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, pp 422–436

Mitzenmacher M, Upfal E (2005) Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press

Newman MEJ (2002) Assortative Mixing in Networks. Physical Review Letters 89(20):208,701. https://doi.org/10.1103/PhysRevLett.89.208701, URL https://link.aps.org/doi/10.1103/PhysRevLett.89.208701

Ojala M (2010) Assessing data mining results on matrices with randomization. In: 2010 IEEE International Conference on Data Mining, pp 959–964, https://doi.org/10.1109/ICDM.2010.20

Ojala M, Garriga GC, Gionis A, et al (2010) Evaluating query result significance in databases via randomizations. In: Proceedings of the 2010 SIAM International Conference on Data Mining (SDM), pp 906–917, https://doi.org/10.1137/1.9781611972801.79

Orsini C, Dankulov MM, Colomer-de Simón P, et al (2015) Quantifying randomness in real networks. Nature communications 6(1):1–10

Papaxanthos L, Llinares-López F, Bodenham D, et al (2016) Finding significant combinations of features in the presence of categorical covariates. In: Advances in Neural Information Processing Systems, pp 2279–2287

Pellegrina L, Vandin F (2020) Efficient mining of the most significant patterns with permutation testing. Data Mining and Knowledge Discovery 34:1201–1234

Pellegrina L, Riondato M, Vandin F (2019a) Hypothesis testing and statistically-sound pattern mining. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, KDD '19, pp 3215–3216, https://doi.org/10.1145/3292500.3332286

Pellegrina L, Riondato M, Vandin F (2019b) SPuManTE: Significant pattern mining with unconditional testing. In: Proceedings of the 25th ACM

SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, KDD ’19, pp 1528–1538, https://doi.org/10.1145/3292500.3330978

Pinxteren S, Calders T (2021) Efficient permutation testing for significant sequential patterns. In: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM), SIAM, pp 19–27

Preti G, De Francisci Morales G, Riondato M (2022) ALICE and the caterpillar: A more descriptive null models for assessing data mining results. In: Proceedings of the 22nd IEEE International Conference on Data Mining, pp 418–427

Relator RT, Terada A, Sese J (2018) Identifying statistically significant combinatorial markers for survival analysis. BMC medical genomics 11(2):31

Ritchie M, Berthouze L, Kiss IZ (2017) Generation and analysis of networks with a prescribed degree sequence and subgraph family: higher-order structure matters. Journal of complex networks 5(1):1–31

Sanei-Mehri SV, Sariyuce AE, Tirthapura S (2018) Butterfly counting in bipartite networks. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 2150–2159

Saracco F, Di Clemente R, Gabrielli A, et al (2015) Randomizing bipartite networks: the case of the world trade web. Scientific reports 5(1):1–18

Sese J, Terada A, Saito Y, et al (2014) Statistically significant subgraphs for genome-wide association study. In: Statistically Sound Data Mining, pp 29–36

Silva ME, Paredes P, Ribeiro P (2017) Network motifs detection using random networks with prescribed subgraph frequencies. In: International Workshop on Complex Networks, Springer, pp 17–29

Sugiyama M, Llinares-López F, Kasenburg N, et al (2015) Significant subgraph mining with multiple testing correction. In: Proceedings of the 2015 SIAM International Conference on Data Mining, SIAM, pp 37–45

Terada A, Okada-Hatakeyama M, Tsuda K, et al (2013a) Statistical significance of combinatorial regulations. Proceedings of the National Academy of Sciences 110(32):12,996–13,001

Terada A, Tsuda K, Sese J (2013b) Fast Westfall-Young permutation procedure for combinatorial regulation discovery. In: 2013 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, pp 153–158

Terada A, Kim H, Sese J (2015) High-speed Westfall-Young permutation procedure for genome-wide association studies. In: Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics, ACM, pp 17–26

Tillman B, Markopoulou A, Gjoka M, et al (2019) 2k+ graph construction framework: Targeting joint degree matrix and beyond. IEEE/ACM Transactions on Networking 27(2):591–606

Tonon A, Vandin F (2019) Permutation strategies for mining significant sequential patterns. In: 2019 IEEE International Conference on Data Mining (ICDM), IEEE, pp 1330–1335

Van Koevering K, Benson A, Kleinberg J (2021) Random graphs with prescribed k-core sequences: A new null model for network analysis. In: Proceedings of the Web Conference 2021, pp 367–378

Verhelst ND (2008) An efficient MCMC algorithm to sample binary matrices with fixed marginals. Psychometrika 73(4):705–728

Vitter JS (1985) Random sampling with a reservoir. ACM Transactions on Mathematical Software 11(1):37–57

Vreeken J, Tatti N (2014) Interesting patterns. In: Frequent pattern mining. Springer, p 105–134

Webb GI (2007) Discovering significant patterns. Machine Learning 68(1):1–33

Westfall PH, Young SS (1993) Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment. Wiley-Interscience

Wu J, He Z, Gu F, et al (2016) Computing exact permutation p-values for association rules. Information Sciences 346:146–162