ROhAN: Row-Order Agnostic Null Models for Statistically-sound Knowledge Discovery

Maryam Abuissa[1], Alexander Lee[1] and Matteo Riondato[1*]

[1*]Department of Computer Science, Amherst College, Box #2232, Amherst College, Amherst, MA, 01002,USA.

*Corresponding author(s). E-mail(s): mriondato@amherst.edu;
Contributing authors: mabuissa24@amherst.edu;
alexanderwlee@proton.me;

Abstract

We introduce a novel class of null models for the statistical validation of results obtained from binary transactional and sequence datasets. Our null models are *Row-Order Agnostic (ROA)*, i.e., do not consider the order of rows in the observed dataset to be fixed, in stark contrast with previous null models, which are *Row-Order Enforcing (ROE)*. We present ROhAN, an algorithmic framework for efficiently sampling datasets from ROA models according to user-specified distributions, which is a necessary step for the resampling-based statistical hypothesis tests employed to validate the results. ROhAN uses Metropolis-Hastings or rejection sampling to build on top of existing or future ROE sampling procedures. Our experimental evaluation shows that ROA models are very different from ROE ones, impacting the statistical validation, and that ROhAN is efficient, mixes fast, and scales well as the dataset grows.

Keywords: Hypothesis Testing, Pattern Mining, Sequences, Transactions

"Forth Eorlingas!" — King Théoden of Rohan

1 Introduction

Results obtained from a dataset through the Knowledge Discovery from Data (KDD) process should be *statistically validated* to ensure that they are not just

due to the randomness in the Data-Generating Process (DGP) (Hämäläinen and Webb, 2019; Pellegrina et al, 2019; Zimmermann, 2014): the goal of the analysis is to *gain new knowledge about the DGP through the observed dataset*, rather than knowledge about the dataset itself.

A rigorous validation approach subjects the results to *statistical hypothesis tests* (Lehmann and Romano, 2022): results that pass the tests are deemed *statistically significant*,[1] as they appear to give new information on the DGP.

The significance of the results is assessed against a *null model* $(\mathcal{Z}, \pi)$, where the *null set* $\mathcal{Z}$ is the collection of datasets that the DGP *may* generate, which are assumed to share some characteristics with the observed dataset (e.g., size, frequency of items, number of simple patterns, ...), and π is a user-specified probability distribution over $\mathcal{Z}$. The null model captures assumed or existing knowledge about the DGP. Results that are deemed significant under an appropriate null model constitute new knowledge about the DGP.

A null model is partially independent from the task whose results one wants to validate, as it models the generation of datasets, not directly of results, but on the other hand, it is used to evaluate the results of the task, so it needs to be representative of the task. The choice of the null model by the user must therefore be *deliberate and informed*, as the meaning of "significant" depends on it: results deemed significant under one null model cannot in general be compared to those deemed significant under a different null model. "All models are wrong, but some are useful" (George E. P. Box), and some null models may be more appropriate for testing the significance of the results of a task than others, because they *more closely represent the settings of the task*. Many null models should be available, capturing different properties of the observed dataset, and users must be informed of their differences, so they can choose the one most appropriate for their needs (Ferkingstad et al, 2015). In this work, we present null models that, we argue, are more appropriate for many data mining tasks from transactional datasets, thus expanding the "library" of models available to practitioners.

Many hypothesis tests are based on *resampling* (Westfall and Young, 1993; Lehmann and Romano, 2022): they analyze multiple datasets drawn from the null model in order to approximate the distribution of the test statistic, and then compare the observed value of the statistic against such distribution. Thus, *computationally-efficient* procedures to sample from the null model distribution are necessary for statistically validating KDD results.

Contributions

We study the problem of evaluating the significance of results from binary transactional[2] and sequence datasets, using resampling-based hypothesis tests.

- We introduce a *novel class of null models* for these datasets. Our models are *Row-Order Agnostic* (ROA), i.e., do not consider the *order of the rows (i.e., transactions or sequences) in the observed dataset to be fixed*. Previous null

[1] Throughout this work, we use "significant" to mean "statistically significant".
[2] We drop "binary" and just use "transactional" in the rest of this work.

models were instead *Row-Order Enforcing* (ROE). We argue that the order of the rows is not meaningful for many KDD tasks on such datasets (e.g., frequent pattern mining, large tile identification), thus ROA null models more closely represent the settings of such tasks. Apart from this difference, ROA models can preserve the same properties (e.g., number of rows, lengths of the rows, item/itemset frequencies, ...) as ROE models.

- We present ROHAN, a *general algorithmic approach* for the efficient sampling of datasets from ROA models. Our methods can use *existing or future* approaches for sampling from ROE models as subroutines (thus building on top of a vast literature), and rely on the *Metropolis-Hastings (MH)* algorithm when these are based on the Markov-Chain-Monte-Carlo (MCMC) method, and on *rejection sampling* otherwise. Our procedures can be used in resampling-based hypothesis testing for the validation of KDD results.
- The results of our experimental evaluation show that ROE and ROA null models are not equivalent, and this difference affects the validation of results. We evaluated ROHAN on real datasets: it is fast, (empirically) rapidly-mixing, and scalable as the dataset grows.

2 Related work

Transactional and sequence datasets are a natural representation of data from many areas, from logs, to gene mutations, to temporal events, to athletes' vitals (Hrovat et al, 2015), to satellite images (Méger et al, 2015). They are extremely common, and many KDD methods for them are available. We focus here on works related to the validation of results from such datasets.

Null models for transactional datasets

The need to evaluate the statistical significance of results obtained from transactional datasets has long been noted (Megiddo and Srikant, 1998) and remarked by the KDD community (Zimmermann, 2014). A long line of research studied how to discard non-interesting patterns from mined collections, or directly mine patterns w.r.t. different interestingness measures (Vreeken and Tatti, 2014). This direction is orthogonal to assessing the statistical significance of the results, but they may be combined (Dalleiger and Vreeken, 2022).

Many works focused on finding significant patterns, where the meaning of "significance" is varied. Hämäläinen and Webb (2019) and Pellegrina et al (2019) survey this area, so we focus on the contributions most relevant to ours.

Gionis et al (2007) study a ROE null model $(\mathcal{Z}_{\mathrm{M}}, \pi)$ for transactional datasets, where $\mathcal{Z}_{\mathrm{M}}$ is the set of *all* $I \times J$ binary matrices with the same row and column sums (a.k.a., margins) as the observed dataset.[3] The problem of how to generate such matrices has been studied in mathematics (Ryser, 1963, Ch. 6) (e.g., as the problem of generating bipartite graphs with fixed degree sequences) and statistics (e.g., to sample 2-way $I \times J$ binary contingency tables)

[3] When considering the order of transactions as fixed, as ROE models do, there is a 1:1 correspondence between transactional datasets and binary matrices. The row sums correspond to the transaction lengths, and the column sum to the supports of single items.

for a long time (Besag and Clifford, 1989, Sect. 3), as it has applications to, e.g., ecology (Connor and Simberloff, 1979). Gionis et al (2007) use MCMC approaches to sample from $(\mathcal{Z}_{\mathrm{M}}, \pi)$, to assess KDD results. We argue that the output of many KDD tasks (e.g., frequent itemset mining) from transactional datasets is not dependent on the order of the transactions, and null models that do not consider this order fixed, i.e., the ROA models that we introduce, are more representative of the settings of such tasks. Our algorithmic framework ROHAN can use existing and *future* methods to sample from $(\mathcal{Z}_{\mathrm{M}}, \pi)$ as subroutines to sample from ROA models, thus allowing us to build on top of an extensive literature, discussed in depth by Fout (2022). Recently, Preti et al (2022) presented a ROA model for transactional datasets preserving the number of caterpillars in the graph corresponding to a transactional dataset. They leverage our Lemma 3.

De Bie (2010) proposes null models $(\mathcal{Z}_{\mathbb{E}}, \pi_{\mathrm{MaxEnt}})$ that preserve properties of the observed transactional dataset *in expectation* w.r.t. π_{MaxEnt}, rather than *exactly*, as our ROA models and the ROE model studied by Gionis et al (2007). The distribution π_{MaxEnt} over $\mathcal{Z}_{\mathbb{E}}$ is the one with the maximum entropy among with the required expectations.[4] These models are *ROE in expectation*, thus less appropriate, as argued, for many tasks from transactional datasets, than the ROA models we propose. While requiring the distribution to have maximum entropy may be appropriate in some cases, a user-specified π can incorporate additional existing or assumed knowledge about the DGP in the null model. We therefore do not consider preserving properties in expectation in this work, but developing "in-expectation ROA models", and efficient procedures to sample from them, is a possible direction for future work.

Null models for sequence datasets

Jenkins et al (2022, Sect. 2) discuss previous work on assessing results from sequence datasets in depth, so here we only comment on the most relevant.

Tonon and Vandin (2019) introduce two null models for sequence datasets: one that preserves the number of sequences, the number of itemsets participating in each sequence (i.e., the length of the sequence), and the number of times an itemset participates in the sequences (i.e., the multi-support of the itemset), and a more restrictive null model preserving all structure of the observed dataset, except the order of the itemsets participating in each sequence. A more restrictive model is studied by Pinxteren and Calders (2021). All these models are ROE, as is the null model introduced by Jenkins et al (2022, Sect. 4.2.2.), which preserves the item-lengths of the sequences (i.e., the sums of the lengths of the itemsets participating in them), rather than the lengths. As in the case of transactional datasets, we argue that the order of the sequences in the dataset is not relevant for many KDD tasks, thus motivating our work on

[4] Preserving properties exactly can partially be incorporated in these null models, but they usually make it impossible to derive a closed form for π, with relevant computational consequences. The same is also true for many complex in-expectation constraints (Cimini et al, 2019).

ROA models for sequence datasets. When sampling from ROA models that preserve the same properties as these ROE models, ROHAN employs the efficient methods by Jenkins et al (2022) in combination with rejection sampling.

Gwadera and Crestani (2010) and Low-Kam et al (2013) present maximum entropy ROE null models for sequence datasets. The comments above about the maximum entropy model by De Bie (2010) apply to these models as well.

Null models for other data

ROE null models have been proposed for database tables (Ojala et al, 2010), and real-valued and mixed-value matrices (Ojala et al, 2008; Ojala, 2010). Developing ROA null models, and efficient algorithms to sample from them, is an interesting direction for future work.

3 Preliminaries

We now first define the types of datasets we study, and then discuss the fundamentals of resampling-based statistical hypothesis testing.

3.1 Transactional and sequence datasets

Let $\mathcal{I} \doteq \{a_1, \ldots, a_n\}$ be a finite *alphabet* of $n \doteq |\mathcal{I}|$ *items*. W.l.o.g., $\mathcal{I} \doteq \{1, \ldots, n\}$. An *itemset* $A \subseteq \mathcal{I}$ is any non-empty subset of $\mathcal{I}$.

A *transactional dataset* $\mathcal{D} \doteq \{t_1, \ldots, t_m\}$ is a finite *bag* of $m \doteq |\mathcal{D}|$ itemsets, which, as elements of $\mathcal{D}$, are known as *transactions*. An itemset A *appears* in transaction t when $A \subseteq t$. The *support* $\sigma_{\mathcal{D}}(A)$ *of itemset* A *in the transactional dataset* $\mathcal{D}$ is the number of transactions of $\mathcal{D}$ in which A appears, i.e.,

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

For example, if we let $\mathcal{D} = \{\{1, 2, 4\}, \{2, 4\}\}$, then $\sigma_{\mathcal{D}}(\{2, 4\}) = 2$ since the itemset $\{2, 4\}$ appears in both transactions in $\mathcal{D}$, while, e.g., $\sigma_{\mathcal{D}}(\{1, 2\}) = 1$.

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. The *itemlength* $\|S\| \doteq \sum_{A_i \in S} |A_i|$ is the total number of items in S. A sequence $S = \langle A_1, \ldots, A_{|S|} \rangle$ is a *subsequence* of a sequence $T = \langle B_1, \ldots, B_{|T|} \rangle$, or $S \sqsubseteq T$, if there exists ordered integers $i_1 < i_2 < \cdots < i_{|S|}$ such that $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \ldots, A_{|S|} \subseteq B_{i_{|S|}}$. Suppose that $A = \{1, 2, 4\}$ and $B = \{2, 4\}$, and let $S = \langle A, B, B \rangle$. Then $A, B \in S$, $|S| = 3$, and $\|S\| = 7$. In addition, if $T = \langle A, A, B, C, B \rangle$, for any itemset C, then $S \sqsubseteq T$: a possible choice of indices is $i_1 = 1, i_2 = 2, i_3 = 5$ (or $i_3 = 3$), as $B \subset A$, but other choices are also possible.

A *sequence dataset* $\mathcal{D}$ is a finite bag of sequences, which, as elements of $\mathcal{D}$, are known as *seq-transactions*. The *support* $\sigma_{\mathcal{D}}(S)$ of a sequence S in $\mathcal{D}$ is the number of seq-transactions in $\mathcal{D}$ of which S is a subsequence. 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, if $\mathcal{D} = \{\langle A, B\rangle, \langle A, C, A\rangle, \langle B, C\rangle\}$, then $\sigma_{\mathcal{D}}(A) = 2$ and $\rho_{\mathcal{D}}(A) = 3$.

In the rest of the work, we use the term "row" to refer to a transaction for transactional datasets, or to a sequence for sequence datasets. We also use the term *pattern* to refer to an itemset or a sequence respectively, and we denote with $\mathcal{L}$ the set of all possible patterns. Doing this allows us to define the generic task of *frequent pattern mining*: given a *minimum support threshold* $\theta \in [1, |\mathcal{D}|]$, the set $\mathsf{FP}_{\mathcal{D}}(\theta)$ of *frequent patterns in* $\mathcal{D}$ *w.r.t.* θ is the set of patterns that have support at least θ in $\mathcal{D}$, i.e.,

$$\mathsf{FP}_{\mathcal{D}}(\theta) \doteq \{P \in \mathcal{L} : \sigma_{\mathcal{D}}(P) \geq \theta\} \quad . \tag{1}$$

Efficient algorithms for finding the frequent patterns exist for both transactional and sequence datasets (Agrawal and Srikant, 1994; Pei et al, 2004).

We define transactional and sequence datasets as *bags*, so the rows in them have no fixed order. Later we discuss ROE models for which the order of the rows in a dataset is considered *fixed*. In this case, datasets are *ordered lists* or *vectors* of rows, and we refer to them as *ordered datasets*.

3.2 Null models and hypothesis testing

We tailor the presentation of hypothesis testing to the task of evaluating the significance of the size $|\mathsf{FP}_{\mathcal{D}}(\theta)|$ of the collection of frequent patterns. We choose this simple statistically-sound KDD task because it allows for a self-contained presentation that is also accessible to non-experts, rather than describing an arguably more interesting, but certainly more convoluted task such as mining statistically-significant frequent patterns. Both the ROE and the ROA models we discuss can be used to validate any kind of results obtained from transactional and sequence datasets, including mining statistically-significant frequent patterns, evaluating the correlations between different items, and more.

Statistical significance is assessed w.r.t. a user-specified *null model*, defined on the basis of an *observed dataset* $\mathring{\mathcal{D}}$, given by the user. A null model is a pair $\Pi \doteq (\mathcal{Z}, \pi)$, where $\mathcal{Z}$ is a set of datasets, known as the *null set*, and π is a user-specified probability distribution over $\mathcal{Z}$. The null set $\mathcal{Z}$ is such that $\mathring{\mathcal{D}} \in \mathcal{Z}$ and $\mathcal{Z}$ contains all and only datasets that share some user-specified characteristic properties with $\mathring{\mathcal{D}}$, i.e., the null model depends on the observed dataset $\mathring{\mathcal{D}}$.[5] For example, the user may want to preserve the number $|\mathring{\mathcal{D}}|$ of rows, and/or the support of single items in $\mathring{\mathcal{D}}$, and much more. The user may specify any distribution π over $\mathcal{Z}$. Choosing which properties of $\mathring{\mathcal{D}}$ to preserve, and which distribution to sample from, allows the user to incorporate in the null model existing or assumed knowledge about the DGP, as $\mathcal{Z}$ is the set of all

[5] We do not indicate this fact in the notation, to keep it light.

the datasets that the DGP may generate, and π is the distribution according to which the DGP generates datasets.

The null model is used to understand whether the observed results represent *new* knowledge about the DGP. Specifically, the goal is understanding how "typical" the results from $\mathring{\mathcal{D}}$ are w.r.t. the distribution of the results from datasets sampled from the null model Π: if they are not "typical", the results are considered *significant* (under Π), i.e., expressing new knowledge about the DGP. For example, if we want to assess whether the number $|\mathsf{FP}_{\mathring{\mathcal{D}}}(\theta)|$ of frequent patterns w.r.t. θ in $\mathring{\mathcal{D}}$ is significant, we could make the *null hypothesis*

$$H_0 \doteq \text{“}|\mathsf{FP}_{\mathring{\mathcal{D}}}(\theta)| = \mathbb{E}_{\mathcal{D}\sim\pi}[|\mathsf{FP}_{\mathcal{D}}(\theta)|]\text{”}, \tag{2}$$

and then perform a *statistical hypothesis test* to assess whether there is sufficient evidence that this null hypothesis may be false. If so, we *reject* the null hypothesis and say that the value $|\mathsf{FP}_{\mathring{\mathcal{D}}}(\theta)|$ appears significant.

One way to perform such a test is to approximate the distribution of the statistic of interest (in this case, the number of frequent patterns) by *sampling datasets from the null model* (Lehmann and Romano, 2022, Ch. 17), and then compare the observed statistic $|\mathsf{FP}_{\mathring{\mathcal{D}}}(\theta)|$ to the obtained empirical distribution, as follows. Assume to sample a collection $\mathcal{T} \doteq \{\mathcal{D}_1, \ldots, \mathcal{D}_\ell\}$ of ℓ datasets independently from $\mathcal{Z}$ according to π. The *(empirical)* p-value $\tilde{\mathsf{p}}(\mathring{\mathcal{D}}, \mathcal{T})$ is defined as the fraction of datasets in $\mathcal{T} \cup \{\mathring{\mathcal{D}}\}$ with a number of frequent itemsets w.r.t. θ that is not smaller than the one observed in $\mathring{\mathcal{D}}$, i.e.,

$$\tilde{\mathsf{p}}(\mathring{\mathcal{D}}, \mathcal{T}) \doteq \frac{1 + |\{1 \leq i \leq \ell : |\mathsf{FP}_{\mathcal{D}_i}(\theta)| \geq |\mathsf{FP}_{\mathring{\mathcal{D}}}(\theta)|\}|}{1 + \ell} \ .$$

Now let $\alpha \in (0, 1)$ be a user-specified *acceptable probability of error*. If $\tilde{\mathsf{p}}(\mathring{\mathcal{D}}, \mathcal{T}) \leq \alpha$, then we say $|\mathsf{FP}_{\mathring{\mathcal{D}}}(\theta)|$ is significant at level α, which can be interpreted as meaning there is evidence that the null hypothesis from (2) is false and should be rejected. The value α is the probability of getting a *false discovery*, i.e., of wrongly declaring the observed results significant.

In most statistically-sound KDD tasks, *multiple* hypotheses must be tested. For example, in significant pattern mining (Hämäläinen and Webb, 2019; Pellegrina et al, 2019), there is one hypothesis per pattern. One then wants guarantees, e.g., on the *Family-Wise Error Rate* (FWER), i.e., on the probability of making *any* false discovery. To ensure that the FWER is bounded by an user-specified threshold $\delta \in (0, 1)$, the p-value of each hypothesis to be tested is compared to an *adjusted critical value* $\alpha(\Pi, \mathcal{H}, \delta)$, where $\mathcal{H}$ is the set of the null hypotheses of interest. Resampling approaches for multiple hypothesis testing (Westfall and Young, 1993) compute adjusted critical values using *datasets sampled* according to π, and they have been used with success in significant itemset mining (Pellegrina et al, 2019).

This discussion highlights how *efficient* procedures to draw datasets from $\mathcal{Z}$ independently according to π are needed for assessing the statistical validity

of results obtained from these datasets. Our algorithmic framework ROHAN achieves this goal for ROA models.

4 Row-Order-Enforcing null models

We now describe ROE null models, i.e., models that consider the order of rows in a dataset to be *fixed*, thus permuting the order of the rows results, in general, in a different dataset, and we briefly describe the algorithms to sample from them, using existing examples.

4.1 ROE models for transactional datasets

Gionis et al (2007) define a ROE model $(\mathcal{Z}, \pi)$ where, given an observed ordered dataset $\mathring{\mathcal{D}}$, $|\mathring{\mathcal{D}}| = m$, $\mathcal{Z}$ contains all and only the ordered datasets such that:

1. $|\mathcal{D}| = |\mathring{\mathcal{D}}| = m$, i.e., $\mathcal{D}$ has the same *size*, i.e., number m of transactions, as $\mathring{\mathcal{D}}$; and
2. $\sigma_{\mathcal{D}}(\{a\}) = \sigma_{\mathring{\mathcal{D}}}(\{a\})$, for every *item* $a \in \mathcal{I}$, i.e., each item has the same support in $\mathcal{D}$ and $\mathring{\mathcal{D}}$; and
3. for $i = 1, \ldots, m$, $|\mathcal{D}[i]| = |\mathring{\mathcal{D}}[i]|$, i.e., the transaction at index i of $\mathcal{D}$ has the same length as the transaction at index i of $\mathring{\mathcal{D}}$, for every i.

The distribution π can be any distribution over $\mathcal{Z}$.[6] We call ROE models that maintain the three constraints above *"Size, Item-Supports, and Length Preserving" (SISLP)*. All SISLP null models for a given $\mathring{\mathcal{D}}$ have the *same* null set $\mathcal{Z}$, i.e., they differ only in π. De Bie (2010) considers a null model where the SISLP constraints are preserved only in expectation.

The SISLP models are just one example of ROE models for transactional datasets. One can devise others that preserve additional properties of the observed dataset. We take the SISLP models as an example for the whole class, and most of what we say for them can be applied to other ROE models.

Binary matrices and sampling algorithms

ROE models for transactional datasets effectively equate ordered datasets to *binary* matrices: the (i, j) entry of the matrix $M_{\mathcal{D}}$ corresponding to the ordered dataset $\mathcal{D} = [t_1, \ldots, t_{|\mathcal{D}|}]$ is 1 iff item $j \in t_i$. Thus, the null set $\mathcal{Z}$ of SISLP ROE models corresponds to the set $\mathcal{M}$ of $m \times n$ binary matrices with fixed column sums and fixed rows sums, which is a classical object of study in mathematics (Ryser, 1963, Ch. 6) and statistics (Fout, 2022). This identity is extremely convenient, as it allows to reuse existing algorithms that sample from $\mathcal{M}$ to sample from $\mathcal{Z}$. Indeed Gionis et al (2007) describe, among others, also an MCMC algorithm introduced by Besag and Clifford (1989, Sect. 3),[7] but *any* algorithm to sample from $\mathcal{M}$ can be used, and the literature is extensive (Fout,

[6] Gionis et al (2007) focus on the case where π is the uniform distribution, but extending their discussion to a generic π is straightforward.

[7] If not even earlier.

2022), including importance sampling algorithms (Chen et al, 2005) and recent MCMC algorithms (Wang, 2020).

We now briefly describe one of the MCMC algorithms by Gionis et al (2007). For ease of presentation, we assume here that π is the uniform over $\mathcal{Z}$, i.e., over $\mathcal{M}$. In Sect. 5.3 we show how ROHAN can use this algorithm as a subroutine to sample from SISLP-like ROA models. The algorithm, which we call SWAPRAND (for "Swap Randomization"), runs a Markov chain as follows. The state space is $\mathcal{M}$, and there is an edge from matrix M' to matrix M'' if there are two row indices $1 \leq r_1, r_2 \leq m$ and two column indices $1 \leq c_1, c_2 \leq n$ such that $M'(r_1, c_1) = 1$, $M'(r_1, c_2) = 0$, $M'(r_2, c_1) = 0$, $M'(r_2, c_2) = 1$, and M'' can be obtained from M' by setting $M'(r_1, c_1) = 0$, $M'(r_1, c_2) = 1$, $M'(r_2, c_1) = 1$, and $M'(r_2, c_2) = 0$, i.e., by performing a single *swap*. When running the Markov Chain, the algorithm chooses a neighbor M'' of the current state M' uniformly at random from the $\mathsf{nei}(M')$ neighbors of M', and moves to it with probability $\min\{1, \mathsf{nei}(M')/\mathsf{nei}(M'')\}$, otherwise it stays in M' (i.e., it follows a self-loop). Gionis et al (2007, Alg. 2, Thm. 4.3) give procedures to compute $\mathsf{nei}(M)$ for any matrix, and for drawing a neighbor uniformly at random. It is easy to show that the stationary distribution of this Markov chain is uniform over $\mathcal{M}$. Thus, the algorithm runs the chain for a sufficient number τ of steps to ensure that the distribution of the current state is (approximately) the stationary one, and returns the state at time τ as a sample. This algorithm is just one example of MCMC methods to sample from $\mathcal{M}$, and ROHAN is able to use any such algorithm as a subroutine, as we show in Sect. 5.3.

4.2 ROE models for sequence datasets

Sequence data is more complex or richer than transactional data, which makes it possible to define many null models on it, by preserving different properties of the observed dataset $\mathring{\mathcal{D}}$. Tonon and Vandin (2019), Pinxteren and Calders (2021), and Jenkins et al (2022) give ROE models for sequence datasets, and we now describe two of them as examples, but what we say can be applied to others. The first null model $(\mathcal{Z}^{(1)}, \pi^{(1)})$ is essentially a SISLP model adapted to sequence datasets. $\mathcal{Z}^{(1)}$ is the set of all and only the ordered datasets $\mathcal{D}$ such that

1. $|\mathcal{D}| = |\mathring{\mathcal{D}}| = m$, i.e., $\mathcal{D}$ has the same *size*, i.e., number m of seq-transactions, as $\mathring{\mathcal{D}}$; and
2. for every itemset A participating in at least one seq-transaction of $\mathring{\mathcal{D}}$, it holds $\rho_{\mathcal{D}}(A) = \rho_{\mathring{\mathcal{D}}}(A)$, i.e., the *multi-supports* of itemsets participating in the seq-transactions are preserved; and
3. for $i = 1, \ldots, m$, $|\mathcal{D}[i]| = |\mathring{\mathcal{D}}[i]|$, i.e., the seq-transaction at index i of $\mathcal{D}$ has the same length as the seq-transaction at index i of $\mathring{\mathcal{D}}$, for every i.

The second null model $(\mathcal{Z}^{(2)}, \pi^{(2)})$ preserves the same properties as the first, and also the additional property that, for $i = 1, \ldots, m$, $\|\mathcal{D}[i]\| = \|\mathring{\mathcal{D}}[i]\|$, i.e., the seq-transaction at index i of $\mathcal{D}$ has the same *itemlength* as the seq-transaction at index i of $\mathring{\mathcal{D}}$, for every i.

Jenkins et al (2022) give efficient, exact algorithms for sampling from these and other ROE models for sequence datasets when π is the uniform distribution. Tonon and Vandin (2019) give an MCMC algorithm (a variant of the one described for the SISLP model for transactional datasets in Sect. 4.1) for the first null model, which can be modified to handle non-uniform distributions, and a similar one can also be devised for the second null model.

5 Row-Order-Agnostic null models and ROhAN

Here we introduce ROA null models, which consider datasets as bags of rows, i.e., do not fix the order of the rows. We also describe ROhAN, our algorithmic framework for sampling from ROA null models.

5.1 ROA models for transactional datasets

In ROA models for transactional datasets, the 1:1 mapping between datasets and binary matrices is lost, since this equivalence only holds between *ordered* datasets and binary matrices. We argue that the loss of this elegant identity is completely offset by the advantage of having null models that are more representative of the settings of KDD tasks on these datasets. Consider, for example, the task of mining the frequent patterns $\mathsf{FP}_{\mathcal{D}}(\theta)$ from (1): the definition of this collection does not depend on the order of the transactions in the dataset, and algorithms for finding this collection (e.g., A-Priori, FP-Growth, Eclat) do not rely on the order of the transactions being fixed or being anything but an arbitrary order that the algorithm can choose itself.[8] In general, whenever the KDD task to be performed is insensitive to the order of the rows in the dataset, in the sense that the output of the task is the same for any permutation of the rows, a ROA model is likely more appropriate than a ROE one. The latter could instead be a better choice when the task output includes, even in a potentially implicit way, the identifiers of the rows. The difference could, at times, be subtle: consider for example the task of finding cluster centers for the rows (i.e., finding points in a space), and evaluating the significance of these centers, versus the task of finding a clustering of the rows (i.e., finding a partitioning of the rows) and evaluating the significance of such clustering or, e.g., the significance of groups of rows being in the same cluster. In the first case, a ROA model seems more appropriate than a ROE model. In the second case, it is necessary to know what rows belong to what cluster in order to perform statistical validation, and to analyze how the clusters, which are subsets of rows, differ across different datasets in the null set, thus making a ROE model more appropriate. We stress again that the choice of the null model is crucial, and the user needs to exercise extreme care in this regard. It is therefore hard to give generic advice about which between a ROE and a ROA model is to be preferred.

[8] Some presentations of the algorithms mention a "transaction identifier" associated to each transaction, but this identifier is used only to uniquely label transactions, not for the purpose of ordering the rows, and it is in part a leftover of the idea that a transactional dataset is stored in a table in a relational database.

Properties of the observed dataset $\mathring{\mathcal{D}}$ that can be preserved by ROE models, can also be preserved, with minor modifications in some cases, by ROA models. As an example, we define a ROA SISLP model $(\mathcal{Z}, \pi)$ for $\mathring{\mathcal{D}}$, where $\mathcal{Z}$ contains all and only the unordered datasets such that:

1. $|\mathcal{D}| = |\mathring{\mathcal{D}}| = m$, i.e., $\mathcal{D}$ has the same *size*, i.e., number m of transactions, as $\mathring{\mathcal{D}}$; and
2. $\sigma_{\mathcal{D}}(\{a\}) = \sigma_{\mathring{\mathcal{D}}}(\{a\})$, for every *item* $a \in \mathcal{I}$, i.e., each item has the same support in $\mathcal{D}$ and $\mathring{\mathcal{D}}$; and
3. if we let $\mathcal{D} = \{t_1, \ldots, t_m\}$ and $\mathring{\mathcal{D}} = \{\mathring{t}_1, \ldots, \mathring{t}_m\}$, there is a 1:1 mapping ϕ from $\mathring{\mathcal{D}}$ to $\mathcal{D}$ such that $|\phi(t)| = |t|$ for every transaction $t \in \mathring{\mathcal{D}}$, i.e., $\mathcal{D}$ has the same *distribution of transaction lengths* as $\mathring{\mathcal{D}}$;

The first two properties are the same as the first two in the ROE SISLP model from Sect. 4.1, and the third is a straightforward adaptation of the third one. The distribution π can be any distribution over $\mathcal{Z}$. In Sect. 5.3 we show how to use ROHAN to sample from this model.

We now comment on the differences between ROE and ROA SISLP models. Let $\mathring{\mathcal{D}}$ be an observed dataset, and let $\mathsf{ord}(\mathring{\mathcal{D}})$ be an *ordered* dataset obtained by fixing an arbitrary order of the transactions of $\mathring{\mathcal{D}}$. Consider the null set $\mathcal{Z}_{\mathrm{A}}$ of a ROA SISLP model for $\mathring{\mathcal{D}}$ and the null set $\mathcal{Z}_{\mathrm{E}}$ of a ROE SISLP model for $\mathsf{ord}(\mathring{\mathcal{D}})$. There is a surjective function $\mathsf{un}()$ from $\mathcal{Z}_{\mathrm{E}}$ to $\mathcal{Z}_{\mathrm{A}}$ which maps an ordered dataset to the corresponding unordered one (e.g., $\mathsf{un}(\mathsf{ord}(\mathring{\mathcal{D}})) = \mathring{\mathcal{D}}$). For any $\mathcal{D} \in \mathcal{Z}_{\mathrm{A}}$ let $\mathsf{c}(\mathcal{D})$ be the number of ordered datasets in $\mathcal{Z}_{\mathrm{E}}$ that $\mathsf{un}()$ maps to $\mathcal{D}$ (it holds $\mathsf{c}(\mathcal{D}) \geq 1$). The following lemma shows that the ROE SISLP model $(\mathcal{Z}_{\mathrm{E}}, \pi)$ and the ROA SISLP model $(\mathcal{Z}_{\mathrm{A}}, \pi)$ are not equivalent, in the sense that one cannot sample an ordered dataset $\mathcal{D}$ from $\mathcal{Z}_{\mathrm{E}}$ w.r.t. π, and consider the *unordered* dataset $\mathsf{un}(\mathcal{D})$ as a sample from $\mathcal{Z}_{\mathrm{A}}$ w.r.t. π.

Lemma 1 *There exists an observed dataset $\mathring{\mathcal{D}}$ such that, if we let $\mathcal{D}$ be an ordered dataset drawn uniformly at random from $\mathcal{Z}_{\mathrm{E}}$, then $\mathsf{un}(\mathcal{D})$ is not chosen uniformly at random from $\mathcal{Z}_{\mathrm{A}}$.*

Proof Let $\mathring{\mathcal{D}} = \{\{1,2\},\{1,3\},\{3\}\}$, and assume, w.l.o.g., that $\mathsf{ord}(\mathring{\mathcal{D}}) = [\{1,2\},\{1,3\},\{3\}]$, to which corresponds the binary matrix

$$M = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} .$$

The matrix

$$M' = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

can be obtained from M with a single swap, and it corresponds to the ordered dataset $\mathcal{D}' = [\{1,3\},\{1,2\},\{3\}]$, which, being ordered, is *different* from $\mathsf{ord}(\mathring{\mathcal{D}})$, but it holds $\mathsf{un}(\mathcal{D}') = \mathring{\mathcal{D}} = \mathsf{un}(\mathsf{ord}(\mathring{\mathcal{D}}))$. $\mathcal{Z}_{\mathrm{E}}$ thus contains *at least two* ordered datasets corresponding to the unordered dataset $\mathring{\mathcal{D}}$. From the definition of $\mathcal{Z}_{\mathrm{E}}$,

it holds that it also contains the ordered dataset $\mathcal{D}'' = [\{1,3\},\{1,3\},\{2\}]$, with $\mathsf{un}(\mathcal{D}'') = \{\{1,3\},\{1,3\},\{2\}\}$. It is easy to see that there is no other ordered dataset $\mathcal{D}''' \in \mathcal{Z}_{\mathrm{E}}$ such that $\mathsf{un}(\mathcal{D}''') = \mathsf{un}(\mathcal{D}'')$. Thus, if we sample an ordered dataset $\mathcal{D}$ uniformly at random from $\mathcal{Z}_{\mathrm{E}}$, then there is a higher probability that $\mathsf{un}(\mathcal{D}) = \mathring{\mathcal{D}}$ than $\mathsf{un}(\mathcal{D}) = \mathsf{un}(\mathcal{D}'')$, and our proof is complete. □

Our algorithmic framework ROhAN (Sect. 5.3) returns samples from $\mathcal{Z}_{\mathrm{A}}$ according to π_{A}.

5.2 ROA models for sequence datasets

The reasons for considering ROA models for sequence datasets are similar to those we discussed for transactional datasets, i.e., the order of the seq-transactions is not relevant for many KDD tasks on such data. Results similar to Lemma 1 can be obtained for sequential datasets.

The ROE models from Sect. 4.2 can be "converted" in ROA models in a way similar to what we discussed above for SISLP models for transactional datasets. The consequence of this "conversion" is deep: the correctness of the exact sampling algorithms by Jenkins et al (2022) for these null models depend on their ROE nature, thus they cannot be easily adapted to the ROA models. For example, the algorithm for the first null model considers the observed sequence dataset as a single long vector of itemsets, and samples from the null model by applying to this vector a permutation chosen uniformly at random using the Fisher-Yates algorithm. The key ingredient for the correctness is that the number of permutations resulting in an ordered dataset $\mathcal{D} \in \mathcal{Z}$ is a constant for all datasets. This property is lost in ROA models, thus new algorithms are needed. In Sect. 5.3 we show that ROhAN is able to build on top of efficient algorithms for ROE models, such as those by Jenkins et al (2022).

5.3 ROhAN: sampling from ROA models

We now describe ROhAN, our algorithmic framework for sampling from ROA models. ROhAN uses, as subroutines, algorithms to sample from ROE models, thus allowing us not only to to build on the extensive library of such methods, but also to show that it will be possible to adapt to ROA models any algorithm that may be developed *in the future* for (possibly not-yet-defined) ROE models.

5.3.1 ROhAN-m: using MCMC algorithms for ROE models

We first show ROhAN-m, which essentially "converts" an MCMC algorithm $\mathcal{A}_{\mathrm{E}}$ for a ROE model $(\mathcal{Z}_{\mathrm{E}}, \pi_{\mathrm{E}})$ to an MCMC algorithm $\mathcal{A}_{\mathrm{A}}$ for a ROA model $(\mathcal{Z}_{\mathrm{A}}, \pi_{\mathrm{A}})$ which preserve the same properties, up to the distinction about the *sequence* of row lengths vs. the *distribution* of row lengths, as in the ROE vs. ROA SISLP models from Sect. 4.1 and 5.1 respectively, or similarly for the ROA versions of the null models for sequence datasets from Sect. 4.2. We impose no assumption on the distributions π_{E} and π_{A} nor on their relationship (e.g., they do not need to be both uniform).

The intuition behind ROHAN-M is that given a Markov chain on $\mathcal{Z}_E$ with stationary distribution π_E, we can use the *Metropolis-Hasting (MH)* approach (Mitzenmacher and Upfal, 2005, Ch. 10) to convert it to a Markov chain still defined on $\mathcal{Z}_E$ but with stationary distribution $\zeta = \zeta(\pi_A)$ so that, if we sample an ordered dataset $\mathcal{D}$ from $\mathcal{Z}_E$ w.r.t. ζ, then $\mathsf{un}(\mathcal{D})$ is a sample from $\mathcal{Z}_A$ w.r.t. π_A. We later derive the appropriate ζ to use.

ROHAN-M uses $\mathcal{A}_E$ as a subroutine as follows. Let $\mathcal{D}$ be the ordered dataset that is the current state of the Markov chain on $\mathcal{Z}_E$ used by algorithm $\mathcal{A}_E$, and let $\mathcal{D}'$ be an ordered dataset obtained by *simulating* a step of the Markov chain of $\mathcal{A}_E$ and $\eta_{\mathcal{D}}(\mathcal{D}')$ be the transition probability from $\mathcal{D}$ to $\mathcal{D}'$. The chain used by ROHAN-M will then move to $\mathcal{D}'$ with probability

$$\min\left(\frac{\zeta(\mathcal{D}')\eta_{\mathcal{D}'}(\mathcal{D})}{\zeta(\mathcal{D})\eta_{\mathcal{D}}(\mathcal{D}')}, 1\right), \tag{3}$$

and otherwise stays in $\mathcal{D}$ (i.e., follows a self-loop). The resulting Markov chain has stationary distribution ζ (Mitzenmacher and Upfal, 2005, Ex. 10.12). ROHAN-M runs this Markov chain starting from $\mathsf{ord}(\mathring{\mathcal{D}})$. Once the chain has mixed, the algorithm returns $\mathsf{un}(\mathcal{D})$, where $\mathcal{D}$ is the ordered dataset corresponding to the final state of the chain. We remark that the Markov chain run by ROHAN-M is still defined on $\mathcal{Z}_E$, not on $\mathcal{Z}_A$.

We now move to derive ζ, and then show the correctness of ROHAN-M. The intuition is that the desired probability π_A to sample $\mathcal{D}$ from $\mathcal{Z}_A$ should be "spread" among the $\mathsf{c}(\mathcal{D})$ ordered datasets in $\mathcal{Z}_E$ that $\mathsf{un}()$ maps to $\mathcal{D}$. The stationary distribution used by ROHAN-M is then

$$\zeta(\mathcal{D}) \doteq \frac{\pi_A(\mathsf{un}(\mathcal{D}))}{\mathsf{c}(\mathsf{un}(\mathcal{D}))}, \text{ for } \mathcal{D} \in \mathcal{Z}_E \ . \tag{4}$$

Theorem 2 ROHAN-M *outputs a sample from* $\mathcal{Z}_A$ *with distribution* π_A.

Proof A unordered dataset $\mathcal{D}' \in \mathcal{Z}_A$ is output by ROHAN-M iff the algorithm samples an ordered dataset $\mathcal{D}$ such that $\mathsf{un}(\mathcal{D}) = \mathcal{D}'$. There are $\mathsf{c}(\mathcal{D}')$ such ordered datasets in $\mathcal{Z}_E$, each sampled with probability $\zeta(\mathcal{D})$ as in (4). Thus, the probability of returning $\mathcal{D}'$ is exactly $\pi_A(\mathcal{D}')$. □

The only missing ingredient is an expression for $\mathsf{c}(\mathcal{D})$, which will depend on the type of the data (sequence vs. transactional), and on the null model, but it does not depend on the fact that we are considering MCMC algorithms in this section: the same expressions we present in this section, can be used also when using rejection sampling, as we discuss in Sect. 5.3.2. For transactional datasets, we give an expression valid for essentially any null model, under a weak general assumption. For sequence datasets, the richer nature of the data, and therefore of the null models, makes deriving such a generic expression impossible, so we show it for the two null models from Sect. 4.2. Obtaining

such an expression is really the only necessary additional step needed to use ROHAN-M for other null models.

$\mathsf{c}(\mathcal{D})$ *for transactional datasets*

We now discuss the computation of $\mathsf{c}(\mathcal{D})$ for transactional datasets. The following result gives an expression for this quantity. It is valid as long as the ROE null set $\mathcal{Z}_{\mathrm{E}}$ contains *all* possible ordered datasets corresponding to an unordered dataset $\mathcal{D} \in \mathcal{Z}_{\mathrm{A}}$, which is a very weak assumption, as if that was not the case, it would mean that preserving the ordering of the transactions is important, i.e., a ROE model is appropriate, and a corresponding ROA model would likely not be. The following result has recently been used by Preti et al (2022) for the same purpose.

Lemma 3 *For any dataset $\mathcal{D} \in \mathcal{Z}_{\mathrm{A}}$, let $z_{\mathcal{D}}$ be the maximum length of any transaction in $\mathcal{D}$. For each $1 \leq i \leq z_{\mathcal{D}}$, let T_i be the* bag *of transactions of length i in $\mathcal{D}$. Let $\bar{T}_i = \{\tau_{i,1}, \ldots, \tau_{i,h_i}\}$ be the* set *of transactions of length i in $\mathcal{D}$, i.e., without duplicates. For each $1 \leq j \leq h_i$, let $W_{i,j} \doteq \{t' \in T_i : t' = \tau_{i,j}\}$ be the* bag *of transactions in T_i (including $\tau_{i,j}$) identical to $\tau_{i,j} \in \bar{T}_i$. Then, the number of ordered datasets in $\mathcal{Z}_{\mathrm{E}}$ that are mapped to $\mathcal{D}$ by $\mathsf{un}()$ is*

$$\mathsf{c}(\mathcal{D}) \doteq \prod_{i=1}^{z_{\mathcal{D}}} \underbrace{\binom{|T_i|}{|W_{i,1}|, \ldots, |W_{i,h_i}|}}_{\textit{multinomial coefficient}} = \prod_{i=1}^{z_{\mathcal{D}}} \frac{|T_i|!}{\prod_{j=1}^{h_i} |W_{i,j}|!} \ . \tag{5}$$

Proof Recall that $\mathcal{Z}_{\mathrm{E}}$ depends on the observed dataset $\mathring{\mathcal{D}}$ and on the arbitrary ordering of its transactions in $\mathsf{ord}(\mathring{\mathcal{D}})$, as the ordering fixes the row-sums r_x, $1 \leq x \leq m$. In other words, it fixes the row indices of rows corresponding to transactions of length i, $1 \leq i \leq z_{\mathcal{D}}$, of $\mathcal{D}$. Thus, the number of different ways in which the transactions of $\mathcal{D}$ can be assigned as the transactions of an ordered dataset in $\mathcal{Z}_{\mathrm{E}}$ is the product, over the transaction lengths, of the number q_i of different ways in which the transactions in T_i can be assigned, i.e.,

$$\mathsf{c}(\mathcal{D}) = \prod_{i=1}^{z_{\mathcal{D}}} q_i \ .$$

Thus, we only have to argue that

$$q_i = \binom{|T_i|}{|W_{i,1}|, \ldots, |W_{i,h_i}|},$$

which is true because the multinomial coefficient $\binom{n}{k_1, \ldots, k_h}$ is the number of different permutations of a bag containing n objects such that k_1 objects are indistinguishable among themselves and of type 1, k_2 objects are indistinguishable among themselves and of type 2, and so on (Stanley, 2011, Eq. 1.22).[9] □

Assume now that ROHAN-M is in state $\mathcal{D}$, and that $\mathcal{D}'$ is the proposed state, which is a neighbor of $\mathcal{D}$. The only use of $\mathsf{c}(\mathsf{un}(\mathcal{D}))$ and $\mathsf{c}(\mathsf{un}(\mathcal{D}'))$ by

[9] We assume $\binom{0}{0, \ldots, 0} = 1$.

ROHAN-M is in the computation of the acceptance probability from (3), as $\mathsf{c}(\mathsf{un}(\cdot))$ appears in the definition of ζ from (4). Plugging the r.h.s. of (4) into (3), we obtain

$$\min\left(\frac{\mathsf{c}(\mathsf{un}(\mathcal{D}))}{\mathsf{c}(\mathsf{un}(\mathcal{D}'))}\frac{\pi_{\mathrm{A}}(\mathsf{un}(\mathcal{D}'))}{\pi_{\mathrm{A}}(\mathsf{un}(\mathcal{D}))}\frac{\eta_{\mathcal{D}'}(\mathcal{D})}{\eta_{\mathcal{D}}(\mathcal{D}')}, 1\right) .$$

The distribution π_{A} is given in input, and both $\eta_{\mathcal{D}'}(\mathcal{D})$ and $\eta_{\mathcal{D}}(\mathcal{D}')$ can be obtained from the $\mathcal{A}_{\mathrm{E}}$ MCMC algorithm used to simulate a step of the underlying Markov chain, so we only need to discuss the computation of the ratio $\mathsf{c}(\mathsf{un}(\mathcal{D}))/\mathsf{c}(\mathsf{un}(\mathcal{D}'))$. We now show that obtaining this ratio can be done without having access to either quantity, not even for the first state $\mathcal{D} = \mathsf{ord}(\mathring{\mathcal{D}})$.

Using the notation from the statement of Lemma 3, given a transaction $t \in \mathsf{un}(\mathcal{D})$, suppose $t \in T_i$ for length $1 \leq i \leq z_{\mathsf{un}(\mathcal{D})}$. Further suppose $t = \tau_{i,j} \in \bar{T}_i$, where $1 \leq j \leq h_i$. Let net be a dictionary that maps each different transaction $t \in \mathsf{un}(\mathcal{D})$ to $|W_{i,j}|$, i.e., the size of the bag of transactions equal to t (including t). This data structure is easy to initialize at the start of ROHAN-M and to keep up to date as the chain evolves. We can then obtain $\mathsf{c}(\mathsf{un}(\mathcal{D}))/\mathsf{c}(\mathsf{un}(\mathcal{D}'))$ as shown in Alg. 1, which leverages the fact that $\mathsf{c}(\mathsf{un}(\mathcal{D})) = \mathsf{c}(\mathsf{un}(\mathcal{D}'))$ if $\mathsf{un}(\mathcal{D}) = \mathsf{un}(\mathcal{D}')$ (line 1), and the definition of the multinomial coefficient, to greatly simplify the computation (lines 4–7).

Algorithm 1 Computing $\mathsf{c}(\mathsf{un}(\mathcal{D}))/\mathsf{c}(\mathsf{un}(\mathcal{D}'))$

Input: ordered dataset $\mathcal{D}$, ordered dataset $\mathcal{D}'$ dictionary net
Output: $\frac{\mathsf{c}(\mathsf{un}(\mathcal{D}))}{\mathsf{c}(\mathsf{un}(\mathcal{D}'))}$

1: **if** $\mathsf{un}(\mathcal{D}) = \mathsf{un}(\mathcal{D}')$ **then return** 1
2: $\{t_a, t_b\} \leftarrow \mathsf{un}(\mathcal{D}) \setminus \mathsf{un}(\mathcal{D}')$
3: $\{\bar{t}_a, \bar{t}_b\} \leftarrow \mathsf{un}(\mathcal{D}') \setminus \mathsf{un}(\mathcal{D})$, s.t. $\exists\{a, b\} \subseteq \mathcal{I}$ s.t. $\bar{t}_a = (t_a \setminus \{a\}) \cup \{b\}$ and $\bar{t}_b = (t_b \setminus \{b\}) \cup \{a\}$
4: **for each** $i \in \{a, b\}$ **do**
5: **if** net has key $\bar{t}_i$ **then** $\beta_i \leftarrow \mathsf{net}[\bar{t}_i]$
6: **else** $\beta_i \leftarrow 0$
7: **return** $\frac{(\beta_a+1)(\beta_b+1)}{\mathsf{net}[t_a]\mathsf{net}[t_b]}$

$\mathsf{c}(\mathcal{D})$ *for sequence datasets*

We now show two results on $\mathsf{c}(\mathcal{D})$ for the two null models for sequence datasets from Sect. 5.2: Lemma 4 for the first null model, and Lemma 5 for the second. Algorithms similar to Alg. 1 can be devised for these cases. The ideas presented here should be useful to derive similar ones for other null models (Tonon and Vandin, 2019; Pinxteren and Calders, 2021; Jenkins et al, 2022).

Lemma 4 *For any sequence dataset $\mathcal{D} \in \mathcal{Z}_{\mathrm{A}}$, let $z_{\mathcal{D}}$, T_i, $\bar{T}_i$, and $W_{i,j}$ be defined as in Lemma 3 (with "seq-transaction" in place of "transaction"). Then,*

$$\mathsf{c}(\mathcal{D}) = \prod_{i=1}^{z_{\mathcal{D}}} \binom{|T_i|}{|W_{i,1}|, \ldots, |W_{i,h_i}|} = \prod_{i=1}^{z_{\mathcal{D}}} \frac{|T_i|!}{\prod_{j=1}^{h_i} |W_{i,j}|!} \ . \tag{6}$$

The fact that the expression is the same as the one in (5) should not be surprising, as the first null model is essentially a SILSP null model for sequence datasets. The proof is the same as Lemma 3, so we do not repeat it.

For the second null model, the following result holds.

Lemma 5 *For any dataset $\mathcal{D} \in \mathcal{Z}_{\mathrm{A}}$, let $z_{\mathcal{D}}$ be as in Lemma 4, and let $y_{\mathcal{D}}$ be the maximum* itemlength *of any seq-transaction in $\mathcal{D}$. For each $1 \leq i \leq z_{\mathcal{D}}$, $1 \leq j \leq y_{\mathcal{D}}$ let $T_{i,j}$ be the* bag *of seq transactions of length i and itemlength j in $\mathcal{D}$. Let $\bar{T}_{i,j} = \{\tau_{i,j,1}, \ldots, \tau_{i,j,h_{i,j}}\}$ be the* set *of seq-transactions of length i and itemlength j in $\mathcal{D}$, i.e., without duplicates. For each $1 \leq k \leq h_{i,j}$, let $W_{i,j,k} \doteq \{t' \in T_{i,j} : t' = \tau_{i,j,k}\}$ be the* bag *of transactions in $T_{i,j}$ (including $\tau_{i,j,k}$) identical to $\tau_{i,j,k} \in \bar{T}_{i,j}$. Then,*

$$\mathsf{c}(\mathcal{D}) = \prod_{i=1}^{z_{\mathcal{D}}} \prod_{j=1}^{y_{\mathcal{D}}} \frac{|T_{i,j}|!}{\prod_{k=1}^{h_{i,j}} |W_{i,j,k}|!}$$

The proof is similar to those for Lemmas 3 and 4, with the necessary adaptation for the fact that we are considering sets/bags of seq-transactions that depend on both length and itemlength.

5.3.2 ROhAN-r: using rejection sampling

Not all algorithms for sampling from a ROE null model $(\mathcal{Z}_{\mathrm{E}}, \pi_{\mathrm{E}})$ are based on MCMC. E.g., Jenkins et al (2022) show non-MCMC algorithms to sample from the first and the second null models for sequence datasets from Sect. 4.2 when π_{E} is uniform. We now describe ROHAN-R, which uses *rejection sampling* (Casella et al, 2004) and such an algorithm $\mathcal{A}$, to sample from a ROA null model $(\mathcal{Z}_{\mathrm{A}}, \pi_{\mathrm{A}})$ which preserves the same properties of the observed dataset as $(\mathcal{Z}_{\mathrm{E}}, \pi_{\mathrm{E}})$, up to the difference between preserving the *sequence* of row lengths vs. the *distribution* of row lengths. $\mathcal{A}$ could even be an MCMC algorithm, but we saw in Sect. 5.3.1 how to directly "upcycle" such methods with ROHAN-M.

For any *unordered* dataset $\mathcal{D} \in \mathcal{Z}_{\mathrm{A}}$, let

$$\rho(\mathcal{D}) \doteq \sum_{\substack{\mathcal{D}' \in \mathcal{Z}_{\mathrm{E}} \text{ s.t.} \\ \mathsf{un}(\mathcal{D}') = \mathcal{D}}} \pi_{\mathrm{E}}(\mathcal{D}') \tag{7}$$

be the probability that $\mathcal{A}$ returns an *ordered* dataset $\mathcal{D}'$ such that $\mathsf{un}(\mathcal{D}') = \mathcal{D}$. Let $Q \in \mathbb{R}$ be a constant such that

$$Q\rho(\mathcal{D}) \geq \pi_{\mathrm{A}}(\mathcal{D}), \text{ for any } \mathcal{D} \in \mathcal{Z}_{\mathrm{A}} \ . \tag{8}$$

ROHAN-R applies rejection sampling by first generating an ordered $\mathcal{D}' \in \mathcal{Z}_E$ using $\mathcal{A}$, and then generating $u \sim \mathcal{U}(0,1)$. If

$$u \leq \frac{\pi_A(\mathsf{un}(\mathcal{D}'))}{Q\rho(\mathsf{un}(\mathcal{D}'))} \tag{9}$$

then $\mathsf{un}(\mathcal{D}')$ is returned as a sample from $\mathcal{Z}_A$ distributed according to π_A. Otherwise, a new $\mathcal{D}' \in \mathcal{Z}_E$ is generated using $\mathcal{A}$, and the process continues. The correctness of ROHAN-R follows from the properties of rejection sampling and of the algorithm $\mathcal{A}$.

The derivation of an expression for the constant Q, which depends on the ROA and ROE null models, but not on the algorithm $\mathcal{A}$, is the only missing ingredient needed to apply ROHAN-R, thus it is left to the user or to the ROE/ROA algorithm designer.

There are even cases when the actual value of Q is not needed, as it partially cancels out in the ratio on the r.h.s. of (9). We now show how that is the case for the two null models for sequence datasets from Sect. 4.2 when π_E and π_A are the uniform distribution and the algorithms to sample from the ROE models are those by Jenkins et al (2022).

Indeed, in these cases we have that $\rho(\mathcal{D}) = {}^{\mathsf{c}(\mathcal{D})}/_{|\mathcal{Z}_E|}$, where $\mathsf{c}(\mathcal{D})$ is either from Lemma 4 or Lemma 5 depending on the null model we are considering. It also holds $\pi_A = {}^{1}/_{|\mathcal{Z}_A|}$. We define $Q \doteq {}^{|\mathcal{Z}_E|}/_{|\mathcal{Z}_A|}$,[10] which clearly is such that the requirement from (8) is satisfied. Then, we have that the condition from (9) can be rewritten as

$$u \leq \frac{1}{\mathsf{c}(\mathsf{un}(\mathcal{D}'))},$$

which is readily computable from Lemma 4 or Lemma 5.

5.4 Discussion

One may wonder whether "wrapping" existing algorithms for ROE models (whether MCMC or not) to obtain algorithms for ROA models, like ROHAN does, is the correct approach, versus creating methods that *directly* sample from a set of *unordered* datasets. We already argued that one of the advantages, and not a small one, of taking the approach we followed, is that one can reuse the large variety of algorithms available (e.g., for ROE SISLP models, i.e., for sampling binary matrices with fixed row- and column-sums, the literature is extensive (Fout, 2022)), and even ones that will be developed in the future. Here we want to briefly discuss, through an example, a non-immediately-apparent drawback of "direct sampling" methods for ROA models. Suppose that we want to develop an MCMC algorithm DIRECTROA for ROA SISLP models, using a Markov chain whose states are the unordered datasets in $\mathcal{Z}_A$ (and not the ordered datasets in $\mathcal{Z}_E$, as in ROHAN-M). We can define the neighborhood structure of the Markov chain by introducing a ROA variant

[10] Other definitions of Q are possible. Deriving, for example, a tight lower bound $b \leq \min_{\mathcal{D} \in \mathcal{Z}_A} \mathsf{c}(\mathcal{D})$ can be used to define $Q \doteq {}^{|\mathcal{Z}_E|}/_{(|\mathcal{Z}_A| b)}$, which would lead to more samples being accepted. We leave this derivation to future work.

of the swap operation used by the MCMC algorithm for sampling from ROE SISLP models (described in Sect. 4.1): there is an edge from $\mathcal{D}'$ to $\mathcal{D}''$ if the latter can be obtained from the former by swapping a pair of items between two transactions that are not one a subset (proper or improper) of the other, and each of which contains only one of the two items. The fact that such operation can be easily defined and implemented, and that it should be easy to draw one such swap uniformly at random to choose a neighbor of the current state to propose as the next step, may lead us to believe that we are on the right track. Additionally, it would seem that a smaller, well-connected, state space could lead to a faster mixing time of the chain. The issue is that, differently from what happens for ROE swaps, there may be *multiple* ROA swaps from $\mathcal{D}'$ to $\mathcal{D}''$, and that the number of ROA swaps from a dataset to any different dataset (i.e., the ROA swaps that would lead from $\mathcal{D}'$ to a $\mathcal{D}'' \neq \mathcal{D}'$) may also be different for different unordered datasets, as it depends on quantities such as the number of identical transactions in $\mathcal{D}'$. The algorithm would need to compute these two quantities at every step, for both the current state and the proposed next state, as their ratio is the neighbor sampling probability $\eta_{\mathcal{D}'}(\mathcal{D}'')$, which is needed to obtain the acceptance probability as in (3). While computing these quantities is possible, it requires maintaining additional data structures and additional computational time at every step, for no clear advantage. We implemented such algorithm DIRECTROA, and we compare ROHAN-M to it in Sect. 6, showing how ROHAN-M performs better in practice.

Extending our approach to null models that preserve constraints (including the row order) in *expectation* (De Bie, 2010), whether using maximum entropy or not, seems challenging, as it requires to derive the probability from (7), which does not seem straightforward in many cases. This is a very interesting direction for future work.

One limitation of this work is that we do not show an upper bound to the *mixing time* of the Markov chain run by ROHAN-M, i.e., the number of steps needed for the distribution of the current state to be (approximately) the stationary distribution (Mitzenmacher and Upfal, 2005, Ch. 10). Using the MH approach makes such a derivation particularly challenging (e.g., is not available for SWAPRAND either), and in any case it would depend on the nature of the Markov chain used by the ROE sampling algorithm that ROHAN-M uses as a subroutine. We measure the mixing time empirically in Sect. 6.

6 Experimental evaluation

Our experimental evaluation focuses on three aspects. First, assessing the difference between ROE and ROA models, showing also how it can impact the validation of results from datasets. Second, measuring the *speed and scalability* of ROHAN-M by measuring its *step time*, i.e., the time to take a step on the Markov chain, and how it changes as the number $|\hat{\mathcal{D}}|$ of transactions in the datasets grows. Third, empirically estimating the *mixing time* of ROHAN-M, i.e., the number of swaps for the distribution of the chain state to be close to

Table 1 Dataset statistics: number of transactions $|\mathring{\mathcal{D}}|$, number of items $|\mathcal{I}|$, density $^{\text{avg}|t|}/_{|\mathcal{I}|}$, where avg$|t|$ is the average transaction length, sum $w \doteq \sum_{i=1}^{m} |t_i|$ of transaction lengths, support threshold θ used in some experiments, and number of frequent itemsets w.r.t. θ.

Dataset $\mathring{\mathcal{D}}$	$\lvert\mathring{\mathcal{D}}\rvert$	$\lvert\mathcal{I}\rvert$	$\frac{\text{avg}\lvert t\rvert}{\lvert\mathcal{I}\rvert}$	w	θ	$\lvert\mathsf{FP}^{\mathring{\mathcal{D}}}(\theta)\rvert$
Foodmart	4,141	1,559	0.0028	18,319	2	4,247
Chess	3,196	75	0.4933	118,252	2,557	8,227
Mushroom	8,416	119	0.1933	193,568	2,525	2,587
BMS 1	59,602	497	0.0051	149,639	60	3,991
BMS 2	77,512	3,340	0.0014	358,278	156	3,683

the stationary distribution. We do not report on the empirical performance of ROhAN-r because it would mostly be an assessment of that of the underlying algorithm used before the rejection sampling step.

Implementation, environment, datasets

All the algorithms and experiments are implemented in Java 8, and available from https://github.com/acdmammoths/ROhAN-code, together with instructions and a script to reproduce all our results and figures. We run our experiments on an x86–64 AWS EC2 instance with the Amazon Linux 2 OS, 128GB of RAM, and 32 vCPUs. We use the following five publicly available[11] binary transactional datasets, whose relevant statistics are in Table 1:

- Foodmart: customer transactions from a retail store.
- Chess: a conversion of the UCI chess (King-Rook vs. King-Pawn) dataset, whose transactions represent chess board configurations.
- Mushroom: a conversion of the UCI mushroom dataset, whose transactions describe different mushrooms using binary features.
- BMS WebView 1 (BMS 1): click-stream data from a webstore used in KDD-Cup 2000, which has been prepared for itemset mining.
- BMS WebView 2 (BMS 2): click-stream data from a webstore used in KDD-Cup 2000, which has been prepared for itemset mining.

Difference between ROE and ROA null models

Consider the ROE SISLP null model $(\mathcal{Z}_{\text{E}}, \pi_{\text{E}})$ for transactional datasets from Sect. 4.1, with π_{E} being the uniform distribution over $\mathcal{Z}_{\text{E}}$, and consider the ROA SISLP model $(\mathcal{Z}_{\text{A}}, \pi_{\text{A}})$ from Sect. 5.1, with π_{A} being the uniform over $\mathcal{Z}_{\text{A}}$. In Lemma 1 we showed an example of an observed dataset $\mathring{\mathcal{D}}$ for which sampling a dataset $\mathcal{D}$ from $\mathcal{Z}_{\text{E}}$ uniformly at random does not imply that $\mathsf{un}(\mathcal{D})$ is a uniform sample from $\mathcal{Z}_{\text{A}}$. The example was artificial, so we want to evaluate the situation on real datasets. Indeed, if there was a constant C such that $\mathsf{c}(\mathsf{un}(\mathcal{D})) = C$ for every $\mathcal{D} \in \mathcal{Z}_{\text{E}}$, then sampling $\mathcal{D}$ from $(\mathcal{Z}_{\text{E}}, \pi_{\text{E}})$ and then considering the unordered dataset $\mathsf{un}(\mathcal{D})$ would be equivalent to sampling

[11] https://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php

Table 2 Difference between models: Minimum, 1st quartile, median, 3rd quartile, and maximum of $\ln(\mathsf{c}(\mathsf{un}(\mathcal{D})))$ across 10,000 states $\mathcal{D} \in \mathcal{Z}_{\mathrm{E}}$.

	Distribution of $\ln(\mathsf{c}(\mathsf{un}(\mathcal{D})))$				
Dataset	min.	Q1	med.	Q3	max.
FOODMART	21,848.016	21,851.48	21,852.99	21,855.47	21,861.94
CHESS	22,589.17	22,596.11	22,598.19	22,598.88	22,599.57
MUSHROOM	67,449.92	67,580.11	67,628.06	67,639.15	67,649.55
BMS 1	343,570.54	345,695.65	347,551.60	349,260.07	350,721.50
BMS 2	541,598.73	542,301.19	542,926.81	543,515.51	544,058.76

from $(\mathcal{Z}_{\mathrm{A}}, \pi)$, implying that the two null models are effectively the same, and perhaps suggesting that the definition of ROA models is not very interesting. The results of our experimental evaluation show instead that, even in this very simple case, ROE and ROA models are very different.

Our experiment performs a (non-covering) random walk over $\mathcal{Z}_{\mathrm{E}}$, and computes the value $\mathsf{c}(\mathsf{un}(\mathcal{D}))$ for each visited state $\mathcal{D}$. While a random walk may visit a state more than once, it never happened in our experiments. The random walk bias towards higher-degree states has no impact on whether $\mathsf{c}(\mathsf{un}(\mathcal{D}))$ is a constant. We report in Table 2 the distribution over 10,000 steps of $\ln(\mathsf{c}(\mathsf{un}(\mathcal{D})))$ (we report the logarithms because the raw quantities are truly "astronomical"). Clearly, $\mathsf{c}(\mathsf{un}(\mathcal{D}))$ is all but a constant: there are datasets in $\mathcal{Z}_{\mathrm{A}}$ which have $\approx e^{5000} \approx 10^{2470}$ times more equivalent ordered datasets in $\mathcal{Z}_{\mathrm{E}}$ than other datasets in $\mathcal{Z}_{\mathrm{A}}$, as can be seen by considering the difference between the maximum and minimum entries for BMS 1 or BMS2, and noting that this difference is the *natural logarithm* of the ratio between the minimum and maximum raw values. Even in the smallest case (Chess), the raw ratio between the minimum and maximum is more than e^{10}. Thus ROE and ROA null models are quite different, i.e., ROA models are a new addition to the library of available null models for statistically-sound KDD.

Impact of null model choice on statistical validation of results

Using a ROA vs. a ROE model may lead to different outcomes in the validation of results obtained from a transactional dataset. We used ROHAN-M (with SWAPRAND as subroutine) and SWAPRAND to respectively compute the significant frequent itemsets (Hämäläinen and Webb, 2019; Pellegrina et al, 2019) under a ROA and a ROE model. The two returned sets of significant patterns in CHESS, with FWER $\delta = 0.05$, were extremely different, with a Jaccard index of 0.12. This fact should not be surprising, as from the difference highlighted in the previous experiment, one should expect that the (empirical) distributions of the test statistics under the two null models would be very different, and therefore so would be the empirical p-values which are used for the tests. Once more, this result is evidence that the user must be extremely cautious in choosing the assumed null model: the meaning of significance depends

on the null model, and it is not meaningful to compare results obtained under different null models (e.g., to compare the statistical power of two procedures).

Table 3 Step time (in ms): minimum, 1st quartile, median, 3rd quartile, and maximum over 10,000 steps.

		step time (ms)				
Dataset	algorithm	min.	Q1	med.	Q3	max.
FOODMART	ROHAN-M	< 1	1	2	2	16
	DIRECTROA	1	2	2	2	32
	SWAPRAND	1	1	2	2	24
CHESS	ROHAN-M	4	5	5	5	24
	DIRECTROA	11	13	14	14	49
	SWAPRAND	3	5	5	5	16
MUSHROOM	ROHAN-M	8	12	13	13	56
	DIRECTROA	22	28	30	33	82
	SWAPRAND	7	9	9	10	47
BMS 1	ROHAN-M	19	25	27	29	63
	DIRECTROA	27	31	32	38	73
	SWAPRAND	19	24	25	27	63
BMS 2	ROHAN-M	33	44	47	50	98
	DIRECTROA	50	55	56	57	103
	SWAPRAND	38	47	49	51	97

Step times

The *step time* is the time needed to obtain a valid swap, compute the MH acceptance probability, and transition to the next state if it is accepted. In Table 3 we report the distribution, over 10,000 steps, of this quantity for three algorithms: ROHAN-M, SWAPRAND, and the "direct" sampling algorithm DIRECTROA described in Sect. 5.4. We show the results for SWAPRAND *only* for comparison purposes: SWAPRAND is not to be preferred just because it appears faster, as it samples from a ROE model while the other two algorithms sample from a ROA model.

The distribution for ROHAN-M is comparable to that of SWAPRAND, while DIRECTROA is slightly slower. This is expected since the execution of SWAPRAND and ROHAN-M are very similar, where the only additional work for ROHAN-M is to compute the ratio of $\mathsf{c}(\mathsf{un}(\mathcal{D}))$ to $\mathsf{c}(\mathsf{un}(\mathcal{D}'))$ using Alg. 1. DIRECTROA is slower, which may seem a bit surprising because one may think that sampling "directly" from the desired space of non-ordered datasets may be more efficient. On the contrary, as discussed in Sect. 5.4, "moving" over

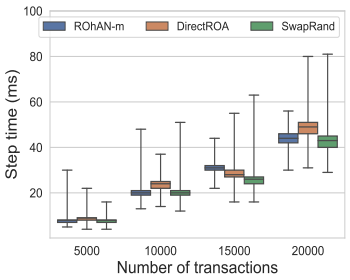


Fig. 1 Scalability results: The step time distribution (in milliseconds) over 10,000 swaps for increasing values of $|\mathring{\mathcal{D}}|$. The line in each box corresponds to the median, the bottom and top of each box correspond to the first and third quartiles, and the lower and upper whiskers correspond to the minimum and maximum.

this space, as the Markov chain of DirectROA does, requires additional computation, which becomes relatively expensive when many transactions have the same length, as in Chess and Mushroom. We find this fact to be an non-intuitive algorithmic observation, which reinforces the appropriateness of the approach taken by ROhAN-m, i.e., reusing existing algorithms for ROE models.

Scalability

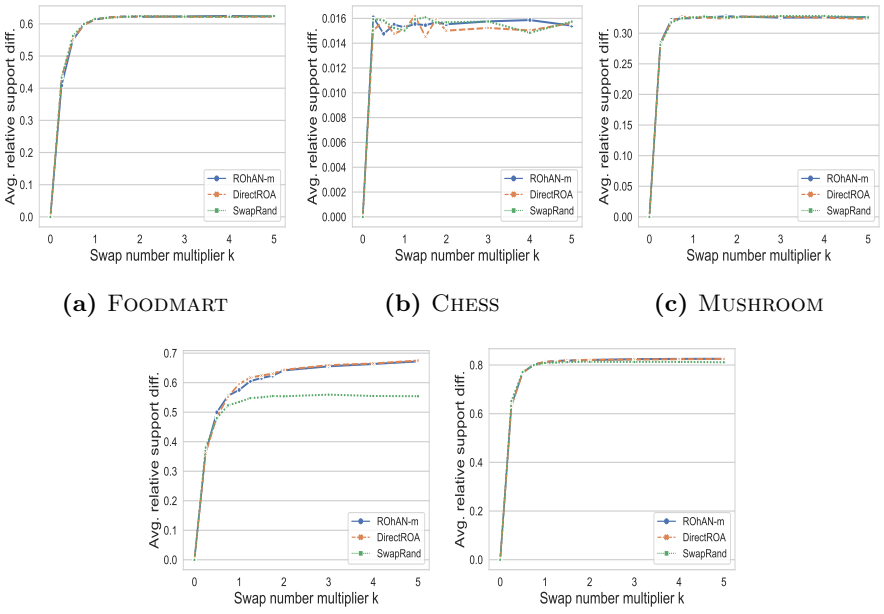
We use the IBM Quest generator to create synthetic datasets with $|\mathring{\mathcal{D}}| \in \{5{,}000, 10{,}000, 15{,}000, 20{,}000\}$, on $|\mathcal{I}| = 100$ and average transaction length $|t| = 25$.[12] We run all algorithms for 10,000 swaps on each dataset, and report the results in Fig. 1. There is a linear relationship between the distribution of step times and the number of transactions, as all algorithms need to compute the number of neighbors for the proposed next state, which takes time linear in $|\mathring{\mathcal{D}}|$. The interquartile range ($Q3 - Q1$) grows in absolute terms because the individual step times grow, but it is essentially constant in relative terms.

Convergence to the stationary distribution

Since we cannot prove an upper bound to the mixing time of the Markov chain used by ROhAN-m (see Sect. 5.4), we empirically estimate it. Following other works (Tonon and Vandin, 2019), we track the *Average Relative Support Difference (ARSD)*, defined as follows, as a proxy for the mixing time: it is assumed that when this quantity stabilizes, the chain has mixed. Given the observed dataset $\mathring{\mathcal{D}}$, let $\theta \in [1, |\mathring{\mathcal{D}}|]$ be a minimum support threshold, and $\mathcal{D}_s$ be the dataset corresponding to the state of the chain after $s \in \mathbb{N}$ swaps. Then,

$$\mathsf{ARSD}(\mathcal{D}_s) \doteq \frac{1}{|\mathsf{FP}_{\mathring{\mathcal{D}}}(\theta)|} \sum_{A \in \mathsf{FP}_{\mathring{\mathcal{D}}}(\theta)} \frac{|\sigma_{\mathring{\mathcal{D}}}(A) - \sigma_{\mathcal{D}_s}(A)|}{\sigma_{\mathring{\mathcal{D}}}(A)} \ .$$

[12] The other parameters of the generator were left to their default values.



(a) FOODMART **(b)** CHESS **(c)** MUSHROOM

(d) BMS 1 **(e)** BMS 2

Fig. 2 Convergence results: $\mathsf{ARSD}(\mathring{\mathcal{D}})$ as the swap number multiplier k grows, where k is s.t. the number of swaps is $s \doteq \lfloor k \sum_{i=1}^{m} |t_i| \rfloor$.

Figure 2 shows $\mathsf{ARSD}(\mathcal{D}_s)$ for $s \doteq \lfloor kw \rfloor$ swaps, where $k \in \{0, 0.25, 0.50, \ldots, 2, 3, 4, 5\}$ and $w \doteq \sum_{i=1}^{m} |t_i|$, for $t_i \in \mathring{\mathcal{D}}$. We use the values of θ from Table 1: the qualitative results do not change with other values.

We remark that comparing the mixing times of Markov chains with different stationary distributions (as SWAPRAND and ROHAN-M) is *meaningless*, as they allow to sample different objects from different sets according to different distributions. Neither are the values of the ARSD comparable, as only the stabilization of the ARSD is a proxy for the mixing time, but its value is not a proxy for the distance between the state distribution and the stationary distribution. Therefore, we do not make such comparisons and only include the results from SWAPRAND for completeness (the mixing time for SWAPRAND is the same observed by Gionis et al (2007, Sect. 5.1)). On BMS 1, the ARSD converges to a different value for SWAPRAND, which we take as another indication that ROE and ROA models are different.

Figure 2 shows that in all cases, the ARSD stabilizes by $s = 2w$ swaps or earlier (by $s = w$), i.e., the mixing time appears to be linear in w. For CHESS, the fluctuations in the ARSD may seem large due to the scale of the y-axis, which is much smaller in Fig. 2b than in the other subfigures. The fact that DIRECTROA requires approximately the same number of steps as ROHAN-M to converge, combined with the fact that each step of DIRECTROA takes

longer (Table 3 and Fig. 1), support the design decisions behind ROhAN-m, as we argued in Sect. 5.4.

7 Conclusion

We introduce a novel type of null models for transactional and sequence datasets, which is *Row-order Agnostic (ROA)*, i.e., does not consider the order of the rows as fixed in the original dataset. These null models expand the collection of null models available to the users to test the significance of results obtained from the datasets, i.e., to perform *statistically-sound KDD*. We present ROhAN, an algorithmic framework for drawing samples from ROA models according to a user-specified distribution, which is a necessary step to assess the significance using resampling-based statistical hypothesis tests. ROhAN employs algorithms for sampling from *Row-Order Enforcing (ROE)* null models as subroutines: it uses the Metropolis-Hastings approach to adapt Markov-Chain-Monte-Carlo algorithms, and rejection sampling for the others. ROhAN is "future-proof" in the sense that even *future* algorithms for *future* ROE models can be easily adapted to be used by ROhAN.

Our experimental evaluation shows that ROA and ROE models are quite different, and this difference impacts the outcomes of the statistical validation of results. We also show that ROhAN is fast, and scales well.

Interesting directions for future work include the definition of ROA null models for other kind of data (e.g., real-valued datasets) and of maximum-entropy ROA models, and efficient algorithms to sample from these null models.

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

Conflict of interest. The authors declare that they have no conflict of interest.

References

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

Besag J, Clifford P (1989) Generalized monte carlo significance tests. Biometrika 76(4):633–642

Casella G, Robert CP, Wells MT (2004) Generalized accept-reject sampling schemes. In: A Festschrift for Herman Rubin, IMS Lecture Notes - Monograph Series, vol 45. IMS, p 342–347

Chen Y, Diaconis P, Holmes SP, et al (2005) Sequential Monte Carlo methods for statistical analysis of tables. Journal of the American Statistical Association 100(469):109–120

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

Connor EF, Simberloff D (1979) The assembly of species communities: chance or competition? Ecology 60(6):1132–1140

Dalleiger S, Vreeken J (2022) Discovering significant patterns under sequential false discovery control. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, KDD '22

De Bie T (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, URL https://doi.org/10.1007%2Fs10618-010-0209-3

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

Fout AM (2022) New methods for fixed-margin binary matrix sampling, Fréchet covariance, and MANOVA tests for random objects in multiple metric spaces. PhD thesis, Colorado State University

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

Gwadera R, Crestani F (2010) Ranking sequential patterns with respect to significance. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, pp 286–299

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

Hrovat G, Fister IJr, Yermak K, et al (2015) Interestingness measure for mining sequential patterns in sports. Journal of Intelligent & Fuzzy Systems 29(5):1981–1994

Jenkins S, Walzer-Goldfeld S, Riondato M (2022) SPEck: mining statistically-significant sequential patterns efficiently with exact sampling. Data Mining and Knowledge Discovery 36(4):1575—1599

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

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

Méger N, Rigotti C, Pothier C (2015) Swap randomization of bases of sequences for mining satellite image times series. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, pp 190–205

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

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

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, Vuokko N, Kallio A, et al (2008) Randomization of real-valued matrices for assessing the significance of data mining results. In: Proceedings of the 2008 SIAM International Conference on Data Mining, SDM '08, pp 494–505, https://doi.org/10.1137/1.9781611972788.45, https://epubs.siam.org/doi/pdf/10.1137/1.9781611972788.45

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

Pei J, Han J, Mortazavi-Asl B, et al (2004) Mining sequential patterns by pattern-growth: The PrefixSpan approach. IEEE Transactions on knowledge and data engineering 16(11):1424–1440

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

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

Ryser HJ (1963) Combinatorial Mathematics. American Mathematical Society

Stanley RP (2011) Enumerative Combinatorics, vol 1, 2nd edn. Cambridge University Press

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

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

Wang G (2020) A fast MCMC algorithm for the uniform sampling of binary matrices with fixed margins. Electronic Journal of Statistics 14(1):1690–1706

Westfall PH, Young SS (1993) Resampling-based multiple testing: Examples and methods for p-value adjustment. John Wiley & Sons

Zimmermann A (2014) The data problem in data mining. SIGKDD Explor 16(2):38–45