

Itemsets for Real-valued Datasets

Nikolaj Tatti

HIIT, Department of Information and Computer Science Aalto University, Finland

nikolaj.tatti@aalto.fi

Abstract—Pattern mining is one of the most well-studied subfields in exploratory data analysis. While there is a significant amount of literature on how to discover and rank itemsets efficiently from binary data, there is surprisingly little research done in mining patterns from real-valued data. In this paper we propose a family of quality scores for real-valued itemsets. We approach the problem by considering casting the dataset into a binary data and computing the support from this data. This naive approach requires us to select thresholds. To remedy this, instead of selecting one set of thresholds, we treat thresholds as random variables and compute the average support. We show that we can compute this support efficiently, and we also introduce two normalisations, namely comparing the support against the independence assumption and, more generally, against the partition assumption. Our experimental evaluation demonstrates that we can discover statistically significant patterns efficiently.

Keywords—pattern mining, itemsets, real-valued itemsets

I. INTRODUCTION

Pattern mining is one of the most well-studied subfields in exploratory data analysis. While there is a significant amount of literature on how to discover and rank itemsets efficiently from binary data, there is surprisingly little research done in mining patterns from real-valued data. In this paper we propose a family of quality scores for real-valued itemsets.

In order to motivate our approach, assume that we are given a dataset D containing real numbers and a miner for mining itemsets from a binary data. The most straightforward way to use the miner to find patterns from D is to transform D into a binary data, and apply the miner. More formally, assume that we have selected a threshold t_i for every item i in the dataset. Then we define a binary data B by setting $b_{ji} = 1$, if $d_{ji} \geq t_i$, and 0 otherwise, where j ranges over all transactions of D .

This approach has two immediate setbacks. Firstly, we have to select the thresholds t_i . In addition, such a measure is coarse, any intricate interaction between items is destroyed as data values are categorised into two coarse categories, 0s and 1s. Hence, instead of selecting just one set of thresholds, we will vary t_i , and instead of computing support only for one dataset, we will compute an average support. More formally, we will attach a distribution $p(R_i = t_i)$ to each threshold and compute the mean $E[fr(X; B)]$, where $fr(X; B)$ is the frequency (support) of an itemset X in a binarized data B .

This approach has several benefits. First of all, the support is monotonically decreasing, which allows us to discover all frequent itemsets efficiently. On the other hand, we will show that we can compute the support efficiently, even though it involves taking an average over a complex function.

We still need to choose the threshold distribution $p(R_i = t_i)$. In this work we focus on a specific distribution involved with copulas [1]: roughly speaking, we will define $p(R_i \leq d_{ji}) = k/(|D| - 1)$, where k is the rank of the j th transaction after data is sorted w.r.t. the i th column. We will see that this distribution induces a support in which the actual values of individual items do not matter, instead the support is based on the ranks of the values. Interestingly enough, several popular statistical tests, such as the Mann-Whitney U test or the Wilcoxon signed-rank test, are also based on the ranks of values.

A standard technique in pattern mining is to compare the observed support against the expected value under some null hypothesis, where the hypothesis is typically an independence assumption. Here we consider two approaches, in the first approach we do a z -normalisation by comparing the support against the independence assumption. In our second approach, we generalise the null hypothesis to a partition model, where we assume that items from different parts of the partition are independent. A particular difficulty with these approaches is that in order to compute them we need to compute the expected mean and the variance. While this is trivial when dealing with simple transactional data, it becomes intricate since the threshold distribution actually depends on the dataset. Nevertheless, we can compute the exact mean and variance for the independence assumption and exact mean and asymptotic variance for the partition assumption. Interestingly enough, the independence test is non-parametric, that is the mean and the variance depend only on the number of datapoints, whereas in the partition assumption we need to estimate parameters from the dataset.

The rest paper of the paper is organized as follows. We introduce preliminary notation in Section II. We define our general measure in Section III and introduce copula support in Section IV. We present an independence test in Section V and test based on partitions in Section VI. We discuss related work in Section VII and present our experiments in Section VIII. Finally, we conclude our paper with remarks in Section IX.

II. PRELIMINARIES AND NOTATION

In this section we introduce the preliminary notation.

A *dataset* D is a multiset of N transactions d_1, \dots, d_N , where $d_j \in \mathbb{R}^K$ is a vector of length K . We will often use $N = |D|$ as the number of datapoints and K as the dimension of the dataset. We treat each vector d_i as a sample from an unknown distribution, $p(a_1, \dots, a_K)$. We refer to the random variables a_i as *items*, or as *features*.

Let $A = \{a_1, \dots, a_K\}$ be the set of all items. An *itemset* X is a set of items $X \subseteq A$. Assume that you are given an itemset X and a *binary* vector $d \in \{0, 1\}^K$. We say that d *covers* X if $d_i = 1$, for every $a_i \in X$. We will use standard notation, by writing $x_1 \cdots x_M$ to mean $\{x_1, \dots, x_M\}$.

Assume now that we are given a collection of binary vectors $D = d_1, \dots, d_N$. We define the *support* or the *frequency* of an itemset X as the proportion of transactions in D covering X ,

$$fr(X; D) = \frac{|\{1 \leq i \leq N; d_i \text{ covers } X\}|}{N} .$$

An important property of the support is that it is monotonically decreasing, that is, $fr(X; D) \leq fr(Y; D)$, if $Y \subseteq X$. This property allows us to use efficient techniques [2] to discover *all* itemsets whose frequency is higher than some given threshold.

III. ITEMSET SUPPORT FOR REAL-VALUED DATA

In this section we define our measure for real-valued data. In order to do so, let D be a dataset over K items, a_1, \dots, a_K , and N transactions. Assume that we are given a threshold $t_i \in \mathbb{R}$ for each item a_i . Let us write $T = (t_1, \dots, t_K)$. Given a vector $x \in \mathbb{R}^K$ of length K , we define $y = x_T$ to be a *binary* vector with $y_i = 1$ if $x_i \geq t_i$, and 0 otherwise. We now define a *binarized* data D_T to be

$$D_T = \{x_T \mid x \in D\} .$$

Essentially, D_T is a dataset where each value is binarized either to 0 or to 1, depending on the threshold. We can now compute a support for a given itemset X by computing $fr(X; D_T)$.

The problem with this approach is that we need to select a threshold set T . Additionally, once we have made this choice, the treatment of values in D is coarse: a value slightly higher than the threshold contributes to the support as much as the values that are significantly higher.

To remedy this, we treat thresholds as random variables. That is, we have K random variables, R_1, \dots, R_K . We will assume that each threshold is assigned independently, that is, R_i are independent variables. We will go over some of the natural choices for distributions of R_i later on. If we write $p(R_i = t_i)$ to be the density function of the i th threshold, we can now define support as an average support, where the

mean is taken over the possible thresholds, that is,

$$\begin{aligned} fr(X; D, p) &= \mathbb{E}[fr(X; D_T)] \\ &= \int_{t_1} \cdots \int_{t_K} fr(X; D_T) \prod_{i=1}^K p(R_i = t_i) dt_i . \end{aligned}$$

The important property of this support is that it is monotonically decreasing. This allows us to mine all frequent itemsets using the standard pattern mining search.

Proposition 1: Assume two itemsets X, Y such that $X \subseteq Y$. Then $fr(X; D, p) \geq fr(Y; D, p)$.

Proof: For any given threshold set T , we have $fr(X; D_T) \geq fr(Y; D_T)$. It follows immediately, that $\mathbb{E}[fr(X; D_T)] \geq \mathbb{E}[fr(Y; D_T)]$, which proves the proposition. ■

Computing the support from the definition is awkward as it requires taking $|X|$ integrals. Fortunately, we can rewrite the support in a much more accessible form.

Proposition 2: Assume a dataset D with N transactions and a distribution p over the thresholds. Then the support of itemset X is equal to

$$fr(X; D, p) = \frac{1}{N} \sum_{x \in D} \prod_{i \in X} p(R_i \leq x_i) .$$

Proof: We can rewrite the support as

$$\begin{aligned} fr(X; D, p) &= \mathbb{E}[fr(X; D_T)] \\ &= \frac{1}{N} \sum_{x \in D} p(x_T \text{ covers } X) . \end{aligned}$$

Transaction $y = x_T$ covers X if and only if $y_i \geq R_i$ for each $i \in X$. Since R_i are independent, it follows that

$$p(x_T \text{ covers } X) = \prod_{i \in X} p(R_i \leq x_i) .$$

This completes the proof. ■

IV. COPULA SUPPORT

Our measure depends on the threshold distribution. In this section we focus on a specific distribution related to copulas.

Assume that we are given a dataset $D = d_1, \dots, d_N$. Let us assume for simplicity that for each item, say a_j , the data points d_{ij} are unique. Fix an item a_j and for notational simplicity let us assume that the datapoints are ordered according to the j th item, $d_{ij} < d_{(i+1)j}$ for $i = 1, \dots, N-1$. Let us define the probability of a threshold R_i by requiring that the threshold will hit the interval $[d_{ij}, d_{(i+1)j}]$ with a probability of $1/(N-1)$, where $i = 1, \dots, N-1$. In other words, the cumulative distribution is equal to

$$p(R_j < d_{ij}) = \frac{i-1}{N-1} .$$

This gives us straightforward way of computing the support. Given a dataset D of N points, we compute $r_{ij} = (c-1)/(N-1)$, where c is the rank of the i th transaction

according to the j th column. We can now define a *copula*¹ support by

$$cp(X; D) = \frac{1}{N} \sum_{i=1}^N \text{rnk}(i; X, D),$$

where $\text{rnk}(i; X, D) = \prod_{j \in X} r_{ij}$.

Example 1: Consider that we are given a dataset with 4 items and 3 transactions

$$\{(1.2, 4.5, 3.8, 8.9), (4.4, 4.7, 1.9, 8.8), (8.2, 8.5, 3.0, 6.5)\} .$$

The corresponding ranks $\{r_{ij}\}$ are then

$$\{(0, 0, 1, 1), (0.5, 0.5, 0, 0.5), (1, 1, 0.5, 0)\} .$$

For example, the copula support for $\{a_2 a_3\}$ is then

$$cp(a_2 a_3) = \frac{1}{3} (0 \times 1 + 0.5 \times 0 + 1 \times 0.5) = \frac{1}{6} .$$

As we see in the experiments, using $cp(X, D)$ as a filtering condition is not enough. Consequently, we also define $cp(X; D, \alpha)$ by setting

$$p(R_i < d_i) = \max \left(\min \left(\frac{i-1-M}{N-1-2M}, 1 \right), 0 \right),$$

where $M = \lfloor \alpha N \rfloor$, that is, the top αN items will be always above threshold and the bottom αN will be always below threshold.

Copula support has some peculiar features. First of all, the support does not depend on the actual values of D , only on their ranks. This makes this support excellent for cases where computing the difference between the values of D does not make sense. In addition to that $cp(a_i; D) = 1/2$ for any item, hence the support is not useful for selecting itemsets of size 1. Even though, we assume that D has independent samples, the ranks r_{ij} are no longer independent. However, if we assume independence between the items, we can compute the mean and the variance as we will see in the next section.

V. COPULA SUPPORT AS A STATISTICAL TEST

A standard technique in pattern mining is to compare the observed support against the independence model. In this section we demonstrate how to do this comparison for copula support. More specifically, we are interested in the quantity

$$z_{\text{IND}}(X; D) = \sqrt{N} \frac{cp(X; D) - \mu}{\sigma},$$

where μ and σ are the mean and the variance of the copula support under the null hypothesis.

We will now show how to compute the mean and the variance of the copula support. In fact, if we set $M = |X|$, then we will show that $\mu = 1/2^M$ and

$$\sigma^2 = \frac{(2N-1)^M}{6^M(N-1)^M} + \frac{(N-2)^M(3N-1)^M}{12^M(N-1)^{2M-1}} - \frac{N}{4^M} .$$

¹Copula stands for a cumulative joint distribution of random variables that have gone through such a transformation [1].

We will also show that $z_{\text{IND}}(X; D)$ approaches the Gaussian distribution $N(0, 1)$ as the number of data points goes to infinity.

To simplify the analysis we will make an assumption that the probability of a tie between two values of an item is 0. This assumption is reasonable if the dataset is generated for example from sensor readings.

We will dedicate the remaining section to proving these results. Note that we cannot use Central Limit Theorem to prove the normality because the ranks of individual rows are not independent. Case in point, $cp(x)$ for a single item will always be $1/2$, hence the variance will be 0 for this case.

In order to prove the result, we will first need to establish some notation. Assume that we have N samples, independent and identically distributed random variables, $\mathcal{Y} = Y_1, \dots, Y_N$, each sample is a vector of size K . Define

$$S_{ij} = \text{rnk}(i; j, \mathcal{Y}) = \frac{1}{N-1} \sum_{k=1}^N I[Y_{ij} > Y_{kj}],$$

where $I[B]$ returns 1 if the statement B is true, and 0 otherwise. Note that the term $I[Y_{ij} = Y_{ij}] = 0$, however, we keep it in the sum for notational convenience. Similarly, we can now define

$$U = cp(X; \mathcal{Y}) = \frac{1}{N} \sum_{i=1}^N \prod_{j \in X} S_{ij} .$$

If we are given a dataset D , then $cp(X; D)$ is an estimate of the random variable U . Our goal is to compute $\mu = E[U]$ and $\sigma^2 = \text{Var}[\sqrt{N}U]$.

Note that since we assume that Y_{ij} and Y_{kl} are independent for $j \neq l$, it follows also that S_{ij} and S_{kl} are also independent for $j \neq l$. However, unlike Y_{ij} and Y_{kj} , S_{ij} and S_{kj} are *not* independent.

In order to continue we need the following lemma.

Lemma 3: Fix j and let i, k , and l be distinct integers. Then

$$\begin{aligned} p(Y_{ij} > Y_{kj}) &= 1/2, \\ p(Y_{ij} > Y_{lj}, Y_{kj} > Y_{lj}) &= 1/3, \\ p(Y_{ij} > Y_{kj}, Y_{kj} > Y_{lj}) &= 1/6 . \end{aligned}$$

Proof: Since the probability of a having a tie between variables is 0, using the symmetry argument, the probability Y_{ij} will be larger than Y_{kj} is $1/2$.

Similarly, if we sort the three variables based on their value, there are 6 possible permutations, each permutation has a probability of $1/6$. There are two permutations that satisfy the second event, namely $Y_{ij} > Y_{kj} > Y_{lj}$ and $Y_{kj} > Y_{ij} > Y_{lj}$. This shows that the probability of the second event is equal to $1/3$. Finally, there is only one permutation that satisfies the third event, namely, $Y_{ij} > Y_{kj} > Y_{lj}$, which proves the lemma. ■

We will first compute the mean of U .

Proposition 4: The average of U is $E[U] = 1/2^M$.

Proof: According to Lemma 3, $E[S_{ij}] = 1/2$. Since S_{ij} and S_{kl} are independent for $j \neq l$, we can write

$$E[U] = \frac{1}{N} \sum_{i=1}^N \prod_{j \in X} E[S_{ij}] = \frac{1}{N} \sum_{i=1}^N \prod_{j \in X} \frac{1}{2} = \frac{1}{2^M} .$$

This proves the result. \blacksquare

Our next step is to compute the variance of U . Since the variables S_{ij} are not independent, we will have to compute them in two stages. Our first step is to compute the second moment of S_{ij} .

Lemma 5: The second moment of S_{ij} is equal to

$$E[S_{ij}^2] = \frac{2N-1}{6(N-1)} .$$

Proof: Decompose the second moment into two sums,

$$\begin{aligned} E[S_{ij}^2] &= \frac{1}{(N-1)^2} E\left[\left(\sum_{k \neq i} I[Y_{ij} > Y_{kj}]\right)^2\right] \\ &= \frac{1}{(N-1)^2} \sum_{k \neq i} p(Y_{ij} > Y_{kj}) \\ &\quad + \frac{1}{(N-1)^2} \sum_{k \neq i} \sum_{l \neq k, i} p(Y_{ij} > Y_{kj}, Y_{ij} > Y_{lj}) . \end{aligned}$$

According to Lemma 3, the terms in the first sum are equal to $1/2$ while the terms in the second sum are equal to $1/3$. This gives us

$$\begin{aligned} E[S_{ij}^2] &= \frac{1}{(N-1)^2} ((N-1)/2 + (N-1)(N-2)/3) \\ &= \frac{1}{6(N-1)} (3 + 2(N-2)) = \frac{2N-1}{6(N-1)} . \end{aligned}$$

This completes the proof. \blacksquare

Our next step is to compute the cross-moment of S_{ij} .

Lemma 6: The cross-moment is equal to

$$E[S_{ij}S_{kj}] = \frac{(N-2)(3N-1)}{12(N-1)^2} .$$

Proof: Decompose the moment into four sums

$$\begin{aligned} E[S_{ij}S_{kj}] &= \frac{1}{(N-1)^2} E\left[\left(\sum_{m \neq i} I[Y_{ij} > Y_{mj}]\right)\left(\sum_{n \neq k} I[Y_{kj} > Y_{nj}]\right)\right] \\ &= \frac{A+B+C+D}{(N-1)^2}, \end{aligned}$$

where

$$\begin{aligned} A &= \sum_{m \neq i, k} p(Y_{ij} > Y_{mj}, Y_{kj} > Y_{mj}), \\ B &= \sum_{m \neq i, k} \sum_{n \neq m, i, k} p(Y_{ij} > Y_{mj}, Y_{kj} > Y_{nj}), \\ C &= \sum_{m \neq i, k} p(Y_{ij} > Y_{kj}, Y_{kj} > Y_{nj}), \quad \text{and} \\ D &= \sum_{m \neq i, k} p(Y_{ij} > Y_{mj}, Y_{kj} > Y_{ij}) . \end{aligned}$$

The random variables in the term of the sum of B are all independent, hence the probability is equal to $1/4$. According to Lemma 3 the term in the sum of A is equal to $1/3$ and the term in the sum for C and D is equal to $1/6$. This gives us

$$A = \frac{N-2}{3}, \quad B = \frac{(N-2)(N-3)}{4}, \quad C = D = \frac{N-2}{6} .$$

Grouping the terms gives us

$$\begin{aligned} E[S_{ij}S_{kj}] &= \frac{4(N-2) + 3(N-2)(N-3) + 4(N-2)}{12(N-1)^2} \\ &= \frac{(N-2)(3N-1)}{12(N-1)^2} . \end{aligned}$$

This completes the proof. \blacksquare

We can now use both lemmas in order to compute the variance.

Proposition 7: The variance $\text{Var}[\sqrt{N}U]$ is equal to

$$\sigma^2 = \frac{(2N-1)^M}{6^M(N-1)^M} + \frac{(N-2)^M(3N-1)^M}{12^M(N-1)^{2M-1}} - \frac{N}{4^M} .$$

Proof: We begin by splitting $E[(\sqrt{N}U)^2]$ into two sums and applying Lemma 5 and Lemma 6,

$$\begin{aligned} E[(\sqrt{N}U)^2] &= \frac{1}{N} \left(\sum_{i=1}^N \prod_{j \in X} E[S_{ij}^2] + \sum_{\substack{i, k \\ i \neq k}} \prod_{j \in X} E[S_{ij}S_{kj}] \right) \\ &= \frac{(2N-1)^M}{6^M(N-1)^M} + \frac{(N-2)^M(3N-1)^M}{12^M(N-1)^{2M-1}} . \end{aligned}$$

We can now use this to express the variance as

$$\begin{aligned} \sigma^2 &= E[(\sqrt{N}U)^2] - N E[U]^2 \\ &= \frac{(2N-1)^M}{6^M(N-1)^M} + \frac{(N-2)^M(3N-1)^M}{12^M(N-1)^{2M-1}} - \frac{N}{4^M} . \end{aligned}$$

This proves the result. \blacksquare

Finally, we show that $z_{\text{IND}}(X; \mathcal{Y})$ approaches a Gaussian distribution. Note that this result does not depend on the assumption that items are independent. Hence, we will be able to use the same result in the next section.

Proposition 8: The quantity $\sqrt{N}(U - E[U])$ approaches a Gaussian distribution as N approaches infinity.

We postpone the proof of this proposition to Appendix.

VI. PRODUCTIVE ITEMSETS AND COPULA SUPPORT

In the previous section we tested the support against the independence assumption. A natural extension of this is to assume a partition of the given itemset such that items are independent only when they belong to different blocks of the partition. In fact, an approach suggested in [3] mines itemsets from binary data whose support is substantially larger than the expectation given by the partition. In order to mimic this for real-valued data, we define

$$z_{\text{PRT}}(X, P; D) = \frac{cp(X; D) - \mu}{\sigma},$$

where P is a partition of X and where μ and σ is the mean and the variance under the assumption that items belonging to different blocks in P are independent. Our final goal is to find a partition that produces the lowest score, that is, a partition that explains the support the best, $z_{\text{PRT}}(X; D) = \min_P z_{\text{PRT}}(X, P; D)$, where P goes over all partitions of at least size 2. Note that we are only interested in one-side test. However, we can easily adjust the formula for a symmetrical two-side test. In addition, in [3] the authors were looking only at partitions of size 2, whereas we go over all non-trivial partitions.

In this section we show how we can compute the needed mean and the variance in order to normalise the support. Unlike with the independence model, the test is no longer non-parametric and we will have to estimate several parameters for each subitemset in the partition. Moreover, we will only provide the variance only when N approaches infinity as the interactions between variables are complex and hard to compute exactly for finite N .

We proceed as follows: We will first show what statistics we need from each subitemset and how to compute them. Then we will show how to use these statistics in order to compute the mean and the variance.

A. Statistics needed to compute the rank

Assume that we are given an itemset $X = x_1 \cdots x_M$. This itemset will eventually be a block in the partition. Let $\mathcal{Y} = Y_1, \dots, Y_N$ be N data samples. Let us shorten $O_{ijx} = \frac{1}{N-1} I[Y_{ix} > Y_{jx}]$. Let us define

$$T_i = \text{rank}(i; X, \mathcal{Y}) = \prod_{x \in X} \sum_{j=1}^N O_{ijx},$$

which is essentially a product of normalised ranks of the i th datapoint. Similar to Section V, let $U = \frac{1}{N} \sum_{i=1}^N T_i$, a random variable corresponding to the copula support $cp(X)$.

Ultimately, we will need three statistics from X , namely $\mu = E[U]$, $\alpha = E[T_1^2]$, and $\beta = \text{Var}[\sqrt{N}U]$. We will discuss how to estimate these statistics in the next subsection. If T_i were distributed independently, then $\beta = \alpha - \mu^2$. However, T_i are dependent. Fortunately, we know enough about the dependency so that we can compute β .

In order to compute β we need to introduce several random variables. Let

$$T_{ix} = \text{rank}(i, X \setminus \{x\}, \mathcal{Y}) = \prod_{y \in X, y \neq x} \sum_{j=1}^N O_{ijy}$$

be the rank of the i th transaction for an itemset $X \setminus \{x\}$. In addition, let us define $C_{kx} = \sum_{i=1}^N T_{ix} O_{ikx}$. We can express the variance β with α , μ and C_{kx} . The benefit of this is that we can estimate these parameters, and by doing so estimate β , as we will demonstrate in the next subsection.

Proposition 9: The variance β approaches

$$\alpha - (M + 1)^2 \mu^2 + \frac{2}{N} \sum_{k=1}^N E\left[\left(\sum_{x \in X} C_{kx}\right)^2\right]$$

as N approaches infinity.

We postpone the proof of this proposition to Appendix.

B. Estimating statistics

Unlike with $z_{\text{IND}}(X)$, the mean and the variance of $z_{\text{IND}}(X; P)$ depend on the underlying distribution, and we are forced to estimate the statistics, namely α , β , μ described in the previous section. These estimates are given in Algorithm 1. Estimating μ and α is trivial. However, estimating β is more intricate due to the last term given in Proposition 9.

Assume that we are given a dataset D and itemset X . Fix $x \in X$ and assume that D is sorted based on x th column, largest first. Let $Z = X \setminus \{x\}$. Note that $\text{rank}(k; Z, D)$ is an estimate for T_{kx} . Hence, we can estimate C_{kx} as

$$\begin{aligned} c_{kx} &= \frac{1}{N-1} \sum_{i=1}^{k-1} \text{rank}(i; Z, D) \\ &= c_{(k-1)x} + \frac{1}{N-1} \text{rank}(k-1; Z, D) \end{aligned}$$

We can use the right-hand side to compute c_{kl} for every k efficiently, and then use c_{kl} to estimate β . We can assume that we have precomputed the order w.r.t. each item x_i before the actual mining. Hence, the cost of estimating the parameters is $O(N|X|)$.

Algorithm 1: ESTIMATE, estimates the statistics needed for z_{PRT} .

input : dataset D , itemset X

output: estimates μ , α , and β

```

1  $\mu \leftarrow cp(X; D)$ ;
2  $\alpha \leftarrow \frac{1}{N} \sum_{i=1}^N \text{rank}(i; X, D)^2$ ;
3  $c_{ix} \leftarrow 0$ ,  $i = 1, \dots, N$ ,  $x \in X$ ;
4 foreach  $x \in X$  do
5    $\left[ \begin{array}{l} \text{sort } D \text{ according to } x, \text{ largest first;} \\ \text{foreach } k \in [2, N] \text{ do} \\ \quad \left[ \begin{array}{l} c_{kx} \leftarrow c_{(k-1)x} + \frac{1}{N-1} \text{rank}(k-1, X \setminus \{x\}, D); \end{array} \right. \end{array} \right.$ 
6
7
8  $\beta \leftarrow \alpha - (|X| + 1)^2 \mu^2 + \frac{2}{N} \sum_{k=1}^N \left( \sum_{x \in X} c_{kx} \right)^2$ ;
9 return  $\mu, \alpha, \beta$ ;
```

We should stress that we use the same dataset to compute the estimates and to compute $z_{\text{PRT}}(X; D)$. This means that $z_{\text{PRT}}(X, P; D)$ will be somewhat skewed and we cannot interpret $z_{\text{PRT}}(X, P; D)$ as a p -value. However, our main goal is not to interpret the obtained values as a statistical test, rather our goal is to rank patterns.

C. Computing z-score

Now that we have computed statistics for each itemset occurring in a partition, we can combine them in order to compute the mean and the variance needed for $z_{\text{PRT}}(X)$.

Proposition 10: Assume that we are given an itemset X and a partition P_1, \dots, P_L of X . Let $\mathcal{Y} = Y_1, \dots, Y_N$ be N random data points. Let $U = cp(X; \mathcal{Y})$, and let $U_i = cp(P_i; \mathcal{Y})$. Let $\mu_i = E[U_i]$, $\alpha_i = E[\text{rnk}(1; P_i, \mathcal{Y})^2]$, $\beta_i = \lim_{N \rightarrow \infty} \text{Var}[\sqrt{N}U_i]$.

Under the assumption that P_i are independent, we have $E[U] = \prod_{i=1}^L \mu_i$ and

$$\text{Var}[\sqrt{N}U] \rightarrow \prod_{i=1}^L \alpha_i + (L-1)\mu^2 + \mu^2 \sum_{i=1}^L \frac{\beta_i - \alpha_i}{\mu_i^2}$$

as N approaches infinity.

We postpone the proof of this proposition to Appendix.

VII. RELATED WORK

While pattern mining has been well researched for binary data, the problem of discovering patterns from real-valued data is open. The most straightforward approach to mine patterns is to discretize data using threshold, see for example [4]. Among methods that do not use thresholds, Calders et al. [5] proposed 3 quality measures for itemsets from numerical attributes. The first two measures were based on the extrema values of the items in an itemset. The most related measure to our work is the third measure, $supp_\tau$, which is a generalisation of Kendall's τ , essentially the number of pairs in which all items are concordant. Interestingly enough, similar to the copula support, $supp_\tau$ also depends only the order of values not on the actual values. In this work we were able to define two normalisations $z_{\text{IND}}(X)$ and $z_{\text{PRT}}(X)$ for our approach, while the authors did not introduce any statistical normalisation for $supp_\tau$. We conjecture that a similar normalisation can be done also for $supp_\tau$.

Jaroszewicz and Korzen [6] suggested discovering polynomial itemsets, essentially cross-moments from real-valued data. We can show that for a certain threshold distribution, our support is equal to the support of polynomial itemsets. Steinbach et al. [7] considered several support functions for itemsets, such as, taking the smallest value in a transaction among the items in the itemset.

Ranking and filtering patterns based on a statistical test has been well studied. Brin et al. compared likelihood-ratio against independence assumption [8]. Webb proposed, among many other criteria, to compare the observed support to an expected support of a partition of size 2 that fits best [3]. More complex null hypotheses such as Bayesian networks [9] or Maximum Entropy models [10] have been also suggested.

Our approach has similarities with mining itemsets from uncertain data [11], where instead of binary data, we have real-valued values between $[0, 1]$ expressing the likelihood

Table I
BASIC STATISTICS OF DATASETS AND EXPERIMENTS

Name	Size	Threshold	Time	patterns
<i>Ind</i>	10 000 × 100	0.1	7m37s	166 750
<i>Plant</i>	10 000 × 100	0.1	7m14s	171 303
<i>Alon</i>	2000 × 62	0.26	8m17s	393 683
<i>Thalia</i>	734 × 69	0.12	2m10s	148 334
<i>Yeast</i>	2993 × 173	0.2	19m50s	529 872

of the entry being equal to 1. In fact, if we interpret r_{ij} values computed in Section IV as probabilistic dataset, then $cp(X)$ will be the same as the expected support computed from probabilistic dataset. However, in probabilistic setting the entries are assumed to be independent, whereas in our case they have an intricate dependency. Consequently, the variance given by Propositions 7 and 9 do not hold for probabilistic datasets. In addition, we cannot compute frequentness measure suggested by Bernecker et al. [12] in our case, however we can estimate it by a normal distribution as suggested by Calders et al. [13].

Defining and computing a quality score for two real-valued variables, essentially an itemset of length 2, is a surprisingly open problem. The approach based on Information Theory was suggested in [14]. An interesting starting point is also a measure of concordance, see Definition 5.1.7 in [1]. These approaches are suitable only for itemsets of size 2 whereas we are interested in measuring the quality of itemset of any size. Finally, Szeékely and Rizzo [15] suggested a measure based on how pair-wise distances correlate. This measure is symmetric while our measure was specifically designed to focus on large values.

VIII. EXPERIMENTS

In this section we present our experiments.

Datasets: We used 2 synthetic and 3 real-world data sets as our benchmark data. The first dataset *Ind* consists of 10 000 data points, each of 100 items, generated independently uniformly from the interval $[0, 1]$. The second dataset *Plant* has the same dimensions as the first dataset. In this dataset we planted 5 subspace clusters each having 4 items: We generated independently $5 \times 10\,000$ boolean variables B_{ti} indicating whether a transaction t belongs to the i th cluster, a transaction can belong to multiple clusters. We set $p(B_{ti} = 1) = 0.4$. If $B_{ti} = 1$, then we set the corresponding items to 0.5. All other values were set to 0. Finally, we added noise sampled uniformly from $[0, 1]$. As real-world benchmark datasets we used the following 3 gene expression data sets: *Alon* [16], *Arabidopsis thaliana* or *Thalia*, and *Saccharomyces cerevisiae* or *Yeast*.² The sizes of the datasets are given in Table I.

Setup: For each dataset we computed frequent itemsets using $cp(X; D, 0.25)$ as a support. We set the threshold such

²*Thalia* and *Yeast* are available at <http://www.tik.ee.ethz.ch/~sop/bimax/>

that we get roughly several hundred thousand itemsets, see Table I. We then ranked itemsets using $z_{\text{IND}}(X)$ and $z_{\text{PRT}}(X)$. The results are given in Figure 1.

Support comparison: Let us first compare supports $cp(X)$ and $cp(X; 0.25)$, given in the top row of Figure 1. We see that for a fixed itemset length there is a strong linear correlation between the supports. The histograms reveal why we should consider $cp(X; 0.25)$ as a stopping criterion instead of $cp(X)$. A significantly large number of itemsets of length, say M , will have larger support than any itemset of length $M + 1$ or higher, that is, in order to discover any itemset of length 3, we will have to discover all itemsets of length 2. This problem does not occur with $cp(X; D, 0.25)$.

Normalisation comparison: Our next step is to compare ranks, given on the second row of Figure 1. As expected $z_{\text{PRT}}(X)$ is more conservative than $z_{\text{IND}}(X)$. For example, in *Ind*, $z_{\text{IND}}(X)$ is distributed as $N(0, 1)$, as predicted by Proposition 8, whereas $z_{\text{PRT}}(X)$ is skewed towards negative values. In general, $z_{\text{IND}}(X)$ prefers large itemsets whereas $z_{\text{PRT}}(X)$ prefers small ones. This can be beneficial as seen with *Plant* dataset. The first 5 itemsets \mathcal{A} according to $z_{\text{IND}}(X)$ are the itemsets related to subspace clusters. However, the next itemsets \mathcal{B} are the clusters with some additional unrelated items, on the other hand, $z_{\text{PRT}}(X)$ will assign a low score to \mathcal{B} . In addition, $z_{\text{PRT}}(X)$ favours sets \mathcal{C} and \mathcal{D} itemsets of size 2–3 that are subitemsets of \mathcal{A} .

Computational complexity: While optimising for speed is not the focus in this work, our implementation³ is able to discover several hundred thousand patterns in minutes. The datasets we consider here are relatively small when compared to the size of the binary datasets used for mining normal patterns. However, the speed of traditional miners is based on the fact that binary datasets are typically very sparse. We do not have the same luxury and computing of each itemset requires a full scan. On the other hand, the cost for computing the support a single itemset depends only on the size of the itemset and the number of datapoints whereas the performance of traditional itemset miners depends heavily on how 1s are distributed in the dataset.

IX. CONCLUDING REMARKS

In this paper we proposed a measure of quality for itemsets mined from real-valued dataset. Our approach was to compute the average support from binarized data with random thresholds. Despite the complex definition we can compute the measure efficiently. As a distribution for a threshold we considered a special distribution related to copulas. We normalised the support by comparing the observed support to the expected support according to a null hypothesis. We considered two hypotheses: the first assumption is that all items are independent, while the

second assumption is more general—we assume that items are independent w.r.t. to a given partition.

This research opens up several directions for future work.

Firstly, we considered one specific threshold distribution. This distribution is a good choice if you do not have any information about the distribution of individual items. However, there are other choices. For example, if we know that data is distributed between $[a, b]$, we can consider a uniform distribution over the interval, see [6], or possibly a shorter interval that excludes the extreme values.

The speed-up techniques used for mining sparse binary data no longer apply. This raises a question whether we can speed up significantly the mining procedure.

Lastly, the distribution of itemsets is different than of those that are obtained from binary data. Typically, in binary data, the margins of the items are distributed unevenly: there will be a lot of items that are rare and some items that are frequent. This means a lot of itemsets will be pruned in first steps. This is not the case with the copula support, where typically you will pass almost all items of size 2. This emphasizes the need for ranking itemsets, in our case, we used $z_{\text{IND}}(X)$ and $z_{\text{PRT}}(X)$. However, as future work it would be interesting to see what type of constraints one can impose on itemsets in order to reduce the output.

ACKNOWLEDGMENT

This work was supported by Academy of Finland grant 118653 (ALGODAN)

REFERENCES

- [1] R. B. Nelsen, *An introduction to Copulas*. Springer, 2006.
- [2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, “Fast discovery of association rules,” in *Advances in Knowledge Discovery and Data Mining*, 1996, pp. 307–328.
- [3] G. I. Webb, “Self-sufficient itemsets: An approach to screening potentially interesting associations between items,” *TKDD*, vol. 4, no. 1, pp. 3:1–3:20, 2010.
- [4] R. Srikant and R. Agrawal, “Mining quantitative association rules in large relational tables,” in *SIGMOD*, 1996, pp. 1–12.
- [5] T. Calders, B. Goethals, and S. Jaroszewicz, “Mining rank-correlated sets of numerical attributes,” in *KDD*, 2006, pp. 96–105.
- [6] S. Jaroszewicz and M. Korzen, “Approximating representations for large numerical databases,” in *SDM*, 2007.
- [7] M. Steinbach, P.-N. Tan, H. Xiong, and V. Kumar, “Generalizing the notion of support,” in *KDD*, 2004, pp. 689–694.
- [8] S. Brin, R. Motwani, and C. Silverstein, “Beyond market baskets: Generalizing association rules to correlations,” in *SIGMOD*, 1997, pp. 265–276.

³Python implementation available at <http://users.ics.aalto.fi/ntatti/>

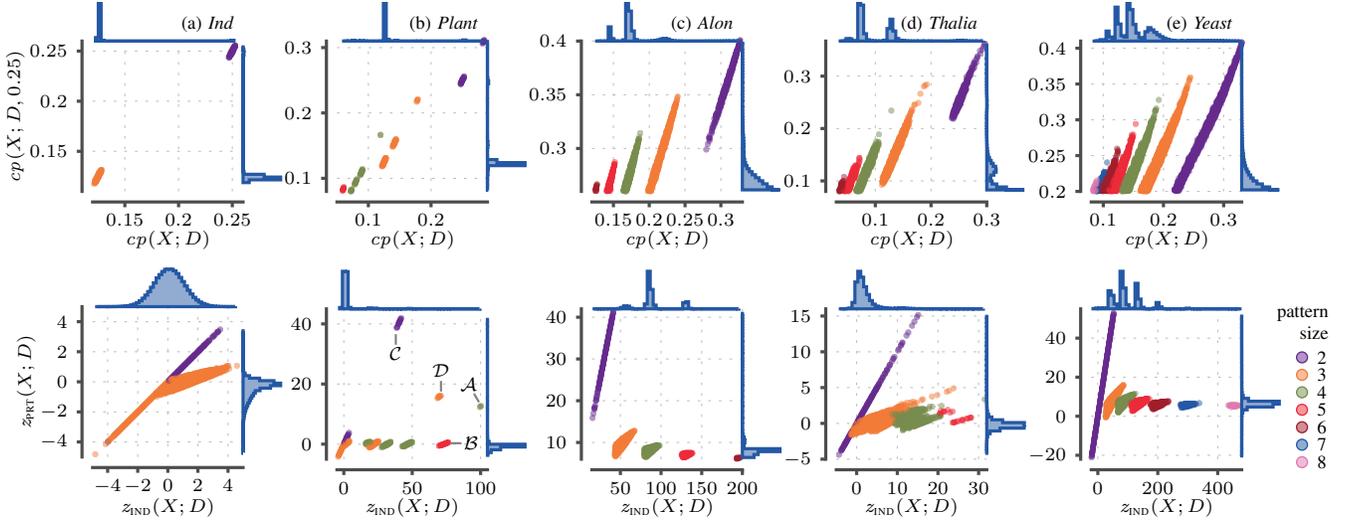


Figure 1. Scatter plots and histograms of supports and ranks. Each plot contains a scatter plot of two variables and the corresponding marginal histograms. The top row contains $cp(X; D, 0.25)$ plotted as a function of $cp(X; D)$. The bottom row contains $z_{PRT}(X; D)$ as a function of $z_{IND}(X; D)$. Each column corresponds to a single dataset. Pattern sizes are encoded with different colours. Note that the axis' ranges vary.

- [9] S. Jaroszewicz and D. A. Simovici, “Interestingness of frequent itemsets using bayesian networks as background knowledge,” in *KDD*, 2004, pp. 178–186.
- [10] N. Tatti, “Maximum entropy based significance of itemsets,” *KAIS*, vol. 17, no. 1, pp. 57–77, 2008.
- [11] C. K. Chui, B. Kao, and E. Hung, “Mining frequent itemsets from uncertain data,” in *PAKDD*, 2007, pp. 47–58.
- [12] T. Bernecker, H.-P. Kriegel, M. Renz, F. Verhein, and A. Züfle, “Probabilistic frequent itemset mining in uncertain databases,” in *KDD*, 2009, pp. 119–128.
- [13] T. Calders, C. Garboni, and B. Goethals, “Approximation of frequentness probability of itemsets in uncertain data,” in *ICDM*, 2010, pp. 749–754.
- [14] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, “Detecting novel associations in large data sets,” *Science*, vol. 334, no. 6062, pp. 518–1524, 2011.
- [15] G. J. Székely and M. L. Rizzo, “Brownian distance covariance,” *The Annals of Applied Statistics*, vol. 3, no. 4, pp. 1236–1265, 2009.
- [16] U. Alon, N. Barkai, D. A. Notterman, K. Gish, D. M. S. Ybarra, and A. J. Levine, “Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays,” *PNAS*, vol. 96, no. 12, pp. 6745–6750, 1999.
- [17] A. W. van der Vaart, *Asymptotic Statistics*. Cambridge University Press, 1998.

APPENDIX

A. Proof of Proposition 8

In order to prove the proposition we need the following proposition.

Proposition 11 (Theorem 12.3 in [17]): Let h be a function (called kernel) of L parameters. Assume that h is symmetric w.r.t. its parameters (that is, any permutation of parameters will yield the same result). Let Y_1, \dots, Y_N be N i.i.d. variables such that $E[h^2(Y_1, \dots, Y_L)] < \infty$. Then

$$\frac{\sqrt{N}}{\binom{N}{L}} \sum_{i_1, \dots, i_L} h(Y_{i_1}, \dots, Y_{i_L}) - \mu,$$

where the sum goes over all subsets of size L and $\mu = E[h(Y_1, \dots, Y_L)]$, approaches a Gaussian distribution as N goes to infinity.

Proof of Proposition 8: Note that since S_{ij} and S_{kj} are not independent, we cannot use Central Limit theorem to prove normality. Instead we will use U -statistics to prove the result. In order to do that let us first define a function of $M + 1$ vectors of length M ,

$$g(y_0, y_1, \dots, y_M) = \prod_{i=1}^M I[y_{0x_i} > y_{ix_i}],$$

where x_i are the items of $X = x_1 \dots x_M$. Note that

$$U = \frac{1}{N(N-1)^M} \sum_{i_0=1}^N \dots \sum_{\substack{i_M=1, \\ i_M \neq i_0}}^N g(Y_{i_0}, \dots, Y_{i_M}).$$

Proposition 11 requires a kernel to be symmetric w.r.t. its parameters. In order to do that, let us define

$$h(y_0, y_1, \dots, y_M) = \sum_{\tau} g(y_{\tau(0)}, \dots, y_{\tau(M)}),$$

where the sum goes over all permutations τ of size $M + 1$. Then according to Proposition 11 a statistic $\sqrt{N}U'$, where

$$U' = \frac{1}{\binom{N}{M+1}} \sum_{i_0, \dots, i_M} h(Y_{i_0}, \dots, Y_{i_M}),$$

where the sum goes over all $M + 1$ subsets of $(1, \dots, N)$, converges to a Gaussian distribution.

The statistics U and U' have the same mean, say $\mu = \mathbb{E}[U] = \mathbb{E}[U']$, but they are different. We will show next that this difference becomes minute as N approaches infinity. In order to do that, let us define

$$a(N) = N(N-1)^M \quad \text{and} \quad b(N) = \binom{N}{M+1} (M+1)! .$$

The sum of U' requires that all rows Y_{i_k} for must be different where as U only requires that Y_{i_0} is different from the remaining rows. Hence, there are $a(N) - b(N)$ less terms in U' . Let Z be the sum of these terms. We have

$$U = \frac{b(N)}{a(N)} U' + \frac{Z}{a(N)} .$$

Let us write $r(N) = (a(N) - b(N))/a(N)$. Both $a(N)$ and $b(N)$ are polynomials of degree $M + 1$ and the coefficient of the highest term is 1 for both polynomials. Consequently, $a(N) - b(N)$ is a polynomial of degree M . This implies that $r(N)$ and $r(N)\sqrt{N}$ both go to 0 as N approaches infinity.

We can express the difference as

$$\sqrt{N}(U - U') = r(N)\sqrt{N}(U' - \mu) + r(N)\sqrt{N}\mu + \frac{Z}{a(N)},$$

According to Proposition 11, $\sqrt{N}(U' - \mu)$ converges to a Gaussian distribution and since $r(N)$ converges to 0, it follows that the first term goes to 0 as N goes to infinity. Similarly, the second term goes to 0 since $r(N)\sqrt{N}$ goes to 0. Finally, to bound the last term note that

$$0 \leq \frac{Z}{a(N)} \leq \frac{a(N) - b(N)}{a(N)} = r(N)$$

which implies that $Z/a(N)$ goes to 0 as N approaches infinity. We have shown that $\sqrt{N}(U - \theta)$ and $\sqrt{N}(U' - \theta)$ converge to each other in probability and that the latter approaches a Gaussian distribution. ■

B. Proof of Proposition 9

First, we need the following technical proposition.

Proposition 12: Assume that we a given integers N and K let $\Omega = [1, \dots, N]^K$ be the set of integer vectors of length K . Let $f_N : \Omega \rightarrow [0, 1]$ be a function such that $\max_{\omega \in \Omega} |f_N(\omega)| \in O(N^{-K+1})$. Let $P \subset \Omega$ be the subset containing only vectors with distinct entries. Assume that we are given $K(K-1)/2$ subsets Ω_{ij} such that

$$\{\omega \in \Omega \mid \omega_i = \omega_j\} \subseteq \Omega_{ij} \subseteq \Omega \setminus P,$$

that is, Ω_{ij} contains vectors for which i th and j th entries are the same and has no vectors from P . Then

$$\sum_{\omega \in \Omega \setminus P} f(\omega) \rightarrow \sum_{i=1}^{K-1} \sum_{j=i+1}^K \sum_{\omega \in \Omega_{ij}} f(\omega)$$

as N approaches infinity.

Proof: Partition Ω into K groups $\Omega^0, \dots, \Omega^{K-1}$ such that

$$\Omega^i = \{\omega \in \Omega \mid \omega \text{ has } K - i \text{ distinct entries}\} .$$

A direct computation shows that

$$|\Omega^i| = S(K, K - i) \frac{N!}{(N - K + i)!} \in O(N^{K-i}),$$

where $S(K, K - i)$ is a Sterling number of the second kind. Let $\Theta = \Omega^2 \cup \dots \cup \Omega^{K-1}$. This immediately implies that $|\Theta| \in O(N^{K-2})$. Note that $P = \Omega^0$ and that $\Omega^1 \subseteq \bigcup_{i,j} \Omega_{ij}$. Let $\Delta_{ij} = \Omega_{ij} \setminus \Omega_1$ be the set of vectors that have i and j as common entries and have less than $K - 1$ unique entries. Note that $\Delta_{ij} \subset \Theta$. Consequently $|\Delta_{ij}| \in O(N^{K-2})$. We can now write the sum as

$$\sum_{\omega \in \Omega \setminus P} f(\omega) = \sum_{i \neq j} \sum_{\omega \in \Omega_{ij}} f(\omega) - \sum_{i \neq j} \sum_{\omega \in \Delta_{ij}} f(\omega) + \sum_{\omega \in \Theta} f(\omega) .$$

Since f is bounded by $O(N^{-K+1})$, the third and fourth terms vanish as N goes to infinity. ■

Proof of Proposition 9: For notational simplicity, let us assume that $X = 1 \dots M$ and define f and g as functions of $2M + 2$ variables,

$$f(i_0, \dots, i_M, j_0, \dots, j_M) = \frac{1}{N} \mathbb{E} \left[\prod_{k=1}^M O_{i_0 i_k k} \prod_{k=1}^M O_{j_0 j_k k} \right]$$

and

$$g(i_0, \dots, j_0, \dots, j_M) = \frac{1}{N} \mathbb{E} \left[\prod_{k=1}^M O_{i_0 i_k k} \right] \mathbb{E} \left[\prod_{k=1}^M O_{j_0 j_k k} \right] .$$

Let $\Omega = [1, \dots, N]^{2M+2}$. Note that

$$\mathbb{E}[NU^2] = \sum_{\omega \in \Omega} f(\omega) \quad \text{and} \quad \mathbb{E}[\sqrt{N}U]^2 = \sum_{\omega \in \Omega} g(\omega) .$$

Let Ω^i as defined in Proposition 12. Let us define

$$\Omega_{ij} = \{\omega \in \Omega \mid \omega_i = \omega_j\}$$

for $1 \leq i \leq M$ and $M + 1 \leq j \leq 2M$, and also

$$\Omega_{ij} = \{\omega \in \Omega^1 \mid \omega_i = \omega_j\}$$

whenever $1 \leq i, j \leq M$ or $M + 1 \leq i, j \leq 2M$. Note that $f(\omega) = g(\omega)$ whenever ω does not share any entry between the first M entries and the last M entries. This holds when $\omega \in \Omega^0$ or when $\omega \in \Omega_{ij}$ for $1 \leq i, j \leq M$ or $M + 1 \leq i, j \leq 2M$.

We can now apply Proposition 12 to conclude that

$$\sum_{\omega \in \Omega} f(\omega) - \sum_{\omega \in \Omega} g(\omega) \rightarrow \sum_{i=1}^M \sum_{j=M+1}^{2M} \sum_{\omega \in \Omega_{ij}} (f(\omega) - g(\omega))$$

as N approaches infinity.

Our final step is to compute the sums in the above equation.

First note that

$$\sum_{\omega \in \Omega_{1(M+1)}} f(\omega) - g(\omega) = \frac{1}{N} \sum_{k=1}^N \mathbb{E}[T_k^2] - \mathbb{E}[T_k]^2 = \alpha - \mu^2.$$

Let $1 < i, j \leq M$. Then

$$\begin{aligned} \sum_{\omega \in \Omega_{i(j+M)}} f(\omega) &= \frac{1}{N} \sum_{k,l,m=1}^N \mathbb{E}[T_{ki} T_{lj} O_{kmi} O_{lmj}] \\ &= \frac{1}{N} \sum_{m=1}^N \mathbb{E} \left[\left(\sum_{k=1}^N T_{ki} O_{kmi} \right) \left(\sum_{l=1}^N T_{lj} O_{lmj} \right) \right] \\ &= \frac{1}{N} \sum_{m=1}^N \mathbb{E}[C_{mi} C_{mj}] = \gamma_{ij}, \end{aligned}$$

where $\gamma_{ij} = \frac{1}{N} \sum_{k=1}^N \mathbb{E}[C_{ki} C_{kj}]$, and

$$\begin{aligned} \sum_{\omega \in \Omega_{i(j+M)}} g(\omega) &= \frac{1}{N} \sum_{k,l,m=1}^N \mathbb{E}[T_{ki} O_{kmi}] \mathbb{E}[T_{lj} O_{lmj}] \\ &= \frac{1}{N} \sum_{m=1}^N \mathbb{E} \left[\sum_{k=1}^N T_{ki} O_{kmi} \right] \mathbb{E} \left[\sum_{l=1}^N T_{lj} O_{lmj} \right] \\ &= \frac{1}{N} \sum_{m=1}^N \mathbb{E}[C_{mi}] \mathbb{E}[C_{mj}] = \mu^2. \end{aligned}$$

The last equality follows from the fact that $\frac{1}{N} \sum_{k=1}^N C_{kl} = U$. Since $\mathbb{E}[C_{kl}]$ does not depend on k , this immediately implies that $\mathbb{E}[C_{kl}] = \mathbb{E}[U] = \mu$.

Let $1 < i \leq M$, Then

$$\begin{aligned} \sum_{\omega \in \Omega_{i(M+1)}} f(\omega) &= \frac{1}{N} \sum_{k,l=1}^N \mathbb{E}[T_{ki} O_{kli} T_l] \\ &= \frac{1}{N} \sum_{k,l,m=1}^N \mathbb{E}[T_{ki} O_{kli} T_{li} O_{lmi}] \end{aligned}$$

and similarly

$$\begin{aligned} \sum_{\omega \in \Omega_{1(M+i)}} f(\omega) &= \frac{1}{N} \sum_{k,l=1}^N \mathbb{E}[T_{li} O_{lki} T_k] \\ &= \frac{1}{N} \sum_{k,l,m=1}^N \mathbb{E}[T_{li} O_{lki} T_{ki} O_{kmi}]. \end{aligned}$$

Since $O_{kli} O_{lmi} + O_{lki} O_{kmi} = O_{kmi} O_{lmi}$ for $k \neq l$ and 0 for $k = l$, summing two previous sums leads to

$$\begin{aligned} &\frac{1}{N} \sum_{k,l,m=1}^N \mathbb{E}[T_{ki} O_{kli} T_{li} O_{lmi} + T_{li} O_{lki} T_{ki} O_{kmi}] \\ &= \frac{1}{N} \sum_{k,l,m=1}^N \mathbb{E}[T_{ki} T_{li} O_{lmi} O_{kmi}] - \frac{1}{N} \sum_{k,m=1}^N \mathbb{E}[T_{ki}^2 O_{kmi}^2] \\ &= \gamma_{ii}^2 - \frac{1}{N(N-1)} \sum_{k=1}^N \mathbb{E}[T_{ki} T_k]. \end{aligned}$$

The last term goes to 0 as N approaches infinity. Hence we have

$$\sum_{\omega \in \Omega_{i(M+1)}} f(\omega) + \sum_{\omega \in \Omega_{1(M+i)}} f(\omega) \rightarrow \gamma_{ii}^2.$$

On the other hand,

$$\begin{aligned} \sum_{\omega \in \Omega_{i(M+1)}} g(\omega) &= \frac{1}{N} \sum_{k,l=1}^N \mathbb{E}[T_{ki} O_{kli}] \mathbb{E}[T_l] \\ &= \frac{1}{N} \sum_k \mathbb{E}[T_k] \mu = \mu^2. \end{aligned}$$

and a similar result holds for $\Omega_{1(M+i)}$. Combining all these equations proves the proposition. ■

C. Proof of Proposition 10

Proof: Since the blocks P_i are independent, it follows immediately that $\mu = \mathbb{E}[U] = \prod_{i=1}^L \mu_i$. In order to prove the result for the variance, let $T_k = \text{rnk}(k; X, \mathcal{Y})$ and $T_{ki} = \text{rnk}(i; P_i, \mathcal{Y})$. Let us define

$$\alpha = \mathbb{E}[T_1^2], \gamma^{(N)} = \mathbb{E}[T_1 T_2], \gamma_i^{(N)} = \mathbb{E}[T_{1i} T_{2i}].$$

We see that $\alpha = \prod_{i=1}^L \alpha_i$ and $\gamma^{(N)} = \prod_{i=1}^L \gamma_i^{(N)}$.

Let us define

$$\beta_i^{(N)} = \text{Var}[\sqrt{N} U_i] \quad \text{and} \quad \beta^{(N)} = \text{Var}[\sqrt{N} U].$$

A straightforward calculation reveals that

$$\beta_i^{(N)} = \alpha_i + (N-1)\gamma_i^{(N)} - N\mu_i^2$$

and

$$\beta^{(N)} = \alpha + (N-1)\gamma^{(N)} - N\mu^2.$$

We can express the variance $\beta^{(N)}$ as

$$\begin{aligned} \beta^{(N)} &= \alpha + (N-1) \prod_{i=1}^L \frac{\beta_i^{(N)} - \alpha_i + N\mu_i^2}{N-1} - N\mu^2 \\ &= \alpha + \frac{\prod_{i=1}^L (\beta_i^{(N)} - \alpha_i + N\mu_i^2) - N(N-1)^{L-1} \mu^2}{(N-1)^{L-1}}. \end{aligned}$$

Let us now consider the right-hand side as a function of N . Both terms in the numerator contain $\mu^2 N^L$, consequently this term is annihilated and the highest term in the numerator has degree of $L-1$, its coefficient is equal to

$$c_N = (L-1)\mu^2 + \mu^2 \sum_{i=1}^L \frac{1}{\mu_i^2} (\beta_i^{(N)} - \alpha_i).$$

Since the highest term in the demoninator is N^{L-1} , the fraction converges to $\lim_{N \rightarrow \infty} c_N$. ■