Data Min Knowl Disc (2014) 28:238–263
DOI 10.1007/s10618-012-0298-2

A statistical significance testing approach to mining the most informative set of patterns

Jefrey Lijffijt · Panagiotis Papapetrou · Kai Puolamäki

Received: 29 December 2011 / Accepted: 21 November 2012 / Published online: 19 December 2012

Abstract Hypothesis testing using constrained null models can be used to compute the significance of data mining results given what is already known about the data. We study the novel problem of finding the smallest set of patterns that explains most about the data in terms of a global *p* value. The resulting set of patterns, such as frequent patterns or clusterings, is the smallest set that statistically explains the data. We show that the newly formulated problem is, in its general form, NP-hard and there exists no efficient algorithm with finite approximation ratio. However, we show that in a special case a solution can be computed efficiently with a provable approximation ratio. We find that a greedy algorithm gives good results on real data and that, using our approach, we can formulate and solve many known data-mining tasks. We demonstrate our method on several data mining tasks. We conclude that our framework is able to identify in various settings a small set of patterns that statistically explains the data and to formulate data mining problems in the terms of statistical significance.

Keywords Data mining algorithms · Pattern mining · Statistical significance testing

Responsible editor: Bart Goethals

J. Lijffijt (✉) · P. Papapetrou · K. Puolamäki
Department of Information and Computer Science,
Aalto University, P.O. Box 15400, 00076 Aalto, Finland
e-mail: jefrey.lijffijt@aalto.fi

P. Papapetrou
Department of Computer Science and Information Systems, Birkbeck, University of London
Malet street, LondonWCIE 7HX, UK

K. Puolamäki
Finnish Institute of Occupational Health, Topeliuksenkatu,
41 a A, FI-00025 Helsinki, Finland

1 Introduction

Assessing the significance of data mining results, such as frequent patterns or clusterings, has recently gained more attention in the data mining community. However, significant results may still have interactions so that one set of results is a consequence of another. In this paper, we formulate a novel approach for finding the smallest set of results that describes the data using statistical significance testing. Our approach is applicable to several data mining problems, such as frequent itemset mining, where the number of results is often prohibitively large.

Several approaches have been proposed to reduce the number of extracted patterns. For example, by using condensed representations, such as non-derivable itemsets (Calders and Goethals 2007), or by mining only significant itemsets (Webb 2007). Although these approaches can reduce the number of patterns, the problem remains that there are very many itemsets and these itemsets may have significant correlations, i.e., one itemset may be highly correlated with other itemsets. Several works have targeted further reduction of the set of patterns and accounting for redundancy, for example by ranking patterns (Mielikäinen and Mannila 2003), by using local objective functions to mine pattern sets directly (Gallo et al. 2007; Bringmann and Zimmermann 2009), by mining patterns iteratively (Hanhijärvi et al. 2009b), or by defining a global objective function and finding a small set of patterns that optimize the criterion (Knobbe and Ho 2006; Vreeken et al. 2011; De Bie 2011a). For a wider overview of the related work see Sect. 5.

Our solution overcomes the redundancy problem by directly optimizing a global p value defined over the data that takes into account the correlations between the patterns. This p value describes the statistical significance of the data and serves as an objective function defined on the data. Each pattern corresponds to a possible constraint on the null model. When imposing a constraint, the global p value of the data will increase.

Consider, for example, the 0–1 data shown in Table 1. Each row corresponds to a transaction and each column corresponds to an item which may be present in or absent from the transaction, indicated by 1 or 0, respectively. The data can be described by 16 frequent itemsets (using an absolute frequency threshold of 3). However, there may

Table 1 An example of a 0–1 dataset

A	B	C	D	E	F	G	H
1	1	1	0	0	0	1	1
1	1	1	0	0	1	0	0
1	1	1	0	0	0	0	1
1	1	1	0	1	1	0	1
0	0	0	1	0	0	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	1	1	0
0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	1
0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	1

Each row corresponds to a transaction and each column corresponds to an item which may be present (indicated by 1) or absent (indicated by 0) in the transaction exist dependencies between these itemsets that could be used to prune the result: if two itemsets are highly correlated then there is no need to report both. Following our approach and using *lift* to define an appropriate test statistic, we can conclude that only two itemsets, ABC and CH, are enough to explain the data. Note that the itemset with the maximum lift (6.24) is *ABCH*, though it is not reported by our algorithm, since it is already explained by the other two. For more details on the method see Sect. 4.1.

Given a null model, a set of predefined patterns, and a test statistic, our goal is to identify the smallest set of patterns that, when imposed as constraints to the null model, leads to the data no longer being significant. We argue that the set of constraints suffices to explain the statistical significance of the data under the null model. The constructed model is the shortest and most informative description of the data, in terms of statistical significance.

Our approach allows to formulate new problems (Sect. 4.1) and reformulate old ones (Sects. 4.2 and 4.3). It can provide insights into the known problems which are not apparent in the original formulation. For example, we can readily assess the statistical significance of the results. The advantage of using several approaches is well recognized in data analysis: consider for example Bayesian, MDL, and frequentist methods, which often lead to similar conclusions and provide complementary views to the same problems.

The main contributions of this paper can be summarized as follows:

- we formulate an approach that employs a global objective function to find the smallest set of patterns that can statistically explain the data,
- we present a greedy algorithm for finding those patterns by optimizing the objective function,
- we provide a theoretical validation of the proposed framework where we show that the problem is NP-hard in general and that in a special case the proposed greedy algorithm is optimal, and
- we demonstrate the applicability and usefulness of the proposed approach in three real applications: frequent itemset mining, time series segmentation, and clustering.

2 Definitions and algorithms

2.1 Definitions

Let Ω denote the sample space, which includes all possible data samples, and let $\omega_0 \in \Omega$ denote the original test sample to be assessed for statistical significance. The null hypothesis is defined by a probability function Pr over the sample space Ω. We use $Pr(\omega)$, with $\omega \in \Omega$, to denote the probability of a single data sample ω, and $Pr(Q)$, where $Q \subseteq \Omega$, to denote the probability mass in Q. $Pr(Q)$ satisfies $Pr(Q) = \sum_{\omega \in Q} Pr(\omega)$.

Let n_C be the number of predefined constraints. Each constraint is indexed in $[n_C] = \{1, \ldots, n_C\}$. Also, let $C_i \subseteq \Omega$ (with $i \in [n_C]$) denote the set of samples in Ω that satisfy constraint i. We require that $\omega_0 \in C_i, \forall i \in [n_C]$ because ω_0 must by definition satisfy all the constraints. A set of constraint indices is denoted by $I \subseteq [n_C]$.

We also assume that each data sample $\omega \in \Omega$ is associated with a test statistic $T(\omega) \in \mathbb{R}$ and define

$$\Omega_- = \{\omega \in \Omega \mid T(\omega) - T(\omega_0) < 0\},$$
$$\Omega_+ = \{\omega \in \Omega \mid T(\omega) - T(\omega_0) \geq 0\}.$$

Ω_- includes all samples where the test statistic is smaller than the test statistic $T(\omega_0)$ and Ω_+ includes all samples where the test statistic is greater than or equal to $T(\omega_0)$. Examples of useful test statistics are given in Sects. 3.3 and 4. We further define $\Omega_I = \cap_{i \in I} C_i$ with $\Omega_\emptyset = \Omega$.

The p value of a set of constraints indexed in I is defined as

$$p(I) = Pr\left(\Omega_+ \mid \Omega_I\right), \tag{1}$$

which is the probability of observing samples with a test statistic higher than or equal to the test statistic of the observed data. The p value serves as the objective function. Typically, the p value in Eq. (1) cannot be solved analytically and has to be approximated by the empirical p value, as discussed in North et al. (2002).

We study the following maximization problem.

Problem 1 *(maximization problem)* For a given k, find a set of constraints $I \subseteq [n_C]$ of size k such that $p(I)$ is maximized.

The following example illustrates how the problem setting can be used in practice.

Example. Consider the problem of randomizing an $m \times n$ binary matrix. The sample space Ω would now contain the set of all $m \times n$ binary matrices. Assuming that the null distribution is the uniform distribution of binary matrices, the probability measure that describes the null hypothesis is defined as $Pr(\omega) = 1/|\Omega| = 2^{-mn}, \forall \omega \in \Omega$. Depending on the quantities of interest, one should define an appropriate test statistic, an example is given in Sect. 4. Also, several types of constraints can be considered here, e.g., row and column margins, itemset frequencies, etc.

For simplicity, let us consider row and columns margins to be the set of constraints, thus introducing a total of $n_C = m + n$ constraints. Each of the m row margin constraints corresponds to a subset $C_i \subseteq \Omega$ that includes all $m \times n$ binary matrices for which the margins of row i are equal to the margin of row i in the input matrix ω_0. The same holds for the set of n column margin constraints. Each time a new constraint is recorded in I, the space of available binary matrices shrinks, and Problem 1 corresponds to finding a set of k row and column margins, such that the p value of the input data ω_0 is maximized.

2.2 Algorithms

A straightforward solution to Problem 1 would be to perform an exhaustive search over all sets of constraints (for which $|I| = k$) and select the subset with the maximal global p value. However, this would require $O(n_C^k)$ running time. We denote this optimal solution by I^*. A naive algorithm with running time $O(n_C)$, denoted as

INDEPENDENT, is to select the k constraints with the highest p values $p(\{i\})$; we denote this solution as I_{IND}.

The GREEDY algorithm (Algorithm 1) solves Problem 1 in a greedy fashion with running time $O(k \cdot n_C)$. At each iteration, the algorithm selects the next constraint that maximizes the objective function, i.e., the global p value, and terminates when $|I| = k$. In the case where more than one constraints maximize the objective function, ties are broken either at random or by choosing the constraint that maximizes the test statistic. We denote this solution as I_{GREEDY}.

Algorithm 1 The greedy algorithm for Problem 1.

```
GREEDY($k$)
*Input: k, number of constraints.*
*Output: I, the set of k constraint indices.*
Let $I \leftarrow \emptyset$.
**for** $j = 1$ to $k$ **do**
    Find $i \in [n_C] \setminus I$ such that $p(I \cup \{i\})$ is maximal. {Ties are broken either at random or by choosing the constraint that maximizes the test statistic.}
    Let $I \leftarrow I \cup \{i\}$.
**end for**
Let $I_{IND} \leftarrow$INDEPENDENT($k$). {The p-values needed by INDEPENDENT are computed at the first iteration of the GREEDY, therefore no additional p-value calculations are needed in this line.}
Let $I \leftarrow I_{IND}$ if $p(I) \le p(I_{IND})$. {$p(I)$ has already been computed, but we may need to compute $p(I_{IND})$.}
**return** $I$
```

In the experiments in Sect. 4, we study only the performance of GREEDY, since the naive algorithm does not take into account any relations that possibly exist between the constraints. The naive algorithm is defined only for illustrative purposes and used in Sects. 3.3 and 3.3.1. Because of the last line of Algorithm 1, we have that GREEDY always outperforms the naive approach: $p(I_{IND}) \le p(I_{GREEDY})$.

3 Theoretical results

In the theoretical analysis that follows, we show that Problem 1 is NP-hard (Sect. 3.1), we prove that there exists no finite approximation algorithm for the general problem (Sect. 3.2), and we show that such an approximation ratio however exists under the assumption that the constraints are approximately "independent" (Sect. 3.3).

3.1 NP-hardness

Theorem 1 *the maximization problem (Problem 1) is NP-hard.*

Proof It is sufficient to show that the corresponding decision problem is NP-hard. A special case of the decision problem is the following: for a finite Ω and a probability measure that satisfies $Pr(\omega) > 0$ for all $\omega \in \Omega$, does there exist a set $I \subseteq [n_C]$ of size of at most k, such that $p(I) = 1$? We can have such a solution only if there exists a set of k constraints $I \subseteq [n_C]$, such that the intersection of all C_i's ($\forall i \in I$) with Ω_- is the empty set.

Equivalently,

$$\Omega_- \cap (\cap_{i \in I} C_i) = \emptyset.$$

Taking the complement on both sides of the equation with respect to Ω, results in

$$\Omega_-^c \cup \left(\cup_{i \in I} C_i^c\right) = \Omega.$$

Next, taking the intersection of both sides of the above equation with Ω_-, results in

$$\Omega_- \cap \left(\cup_{i \in I} C_i^c\right) = \Omega_- \Leftrightarrow$$
$$\cup_{i \in I} \left(\Omega_- \cap (\Omega \setminus C_i)\right) = \Omega_-.$$

Denoting $T_i = \Omega_- \cap (\Omega \setminus C_i)$ we finally obtain $\cup_{i \in I} T_i = \Omega_-$. Therefore, our problem is equivalent to the set cover problem over T_i, where the universe is Ω_-, or in other words, “does there exist a set of k sets T_i such that their union is Ω_-?”. Thus, our problem is NP-hard. □

3.2 Non-approximability

Theorem 2 *Any algorithm to solve Problem 1 using only the p values given by $p(I)$ for any $I \subseteq [n_C]$, and that has lower computational complexity than exhaustive search, can have no finite approximation ratio.*

Proof It is sufficient to construct one adversarial example that has no approximation ratio. Given a set of constraint indices $B \subseteq [n_C]$, we define three auxiliary sets:

- $C(B)$: all samples from Ω that satisfy the constraints indexed by B but no other constraint in $[n_C]$.
- $D(B)$: all samples in $C(B)$ that are also in Ω_+.
- $E(B)$: all samples in $C(B)$ that are also in Ω_-.

More formally, let

$$C_i'(B) = \begin{cases} C_i & , \quad i \in B, \\ \Omega \setminus C_i \ , & \quad \text{otherwise.} \end{cases} \tag{2}$$

Then we define: $C(B) = \cap_{i \in [n_C]} C_i'(B)$, $D(B) = C(B) \cap \Omega_+$ and $E(B) = C(B) \cap \Omega_-$. Also, Ω_I can now be expressed as $\Omega_I = \cup_{B \supseteq I} C(B)$; hence $Pr(\Omega_I) = \sum_{B \supseteq I} Pr(C(B))$, where $C(B) = D(B) + E(B)$, and $Pr(\Omega_I \cap \Omega_+) = \sum_{B \supseteq I} Pr(D(B))$.

In addition, $Pr(D(B))$ and $Pr(E(B))$ can be any non-negative real numbers that sum to unity, that is, $\sum_B (Pr(D(B)) + Pr(E(B))) = 1$, since we are free to choose any set of constraints and test statistic. These probabilities define the problem setting completely.

Using the above formulation and Eq. (1), Problem 1 is to find set $I \subseteq [n_C]$ of size $k \leq n_C$ such that

$$p(I) = Pr(\Omega_+ \mid \Omega_I) = \frac{Pr(\Omega_I \cap \Omega_+)}{Pr(\Omega_I)} = \frac{\sum_{B \supseteq I} Pr(D(B))}{\sum_{B \supseteq I} (Pr(D(B)) + Pr(E(B)))} \quad (3)$$

is maximized.

The idea of the proof is to first randomly choose a set of constraints $S \subseteq [n_C]$ of size k which will correspond to a solution for Problem 1, and then construct a set of probabilities $Pr(D(B))$ and $Pr(E(B))$ such that based only on the values of $p(I)$, $I \subseteq [n_C]$, S can be found only by performing an exhaustive search.

More specifically, we build an example for which the p values satisfy

$$p(I) = \begin{cases} 1 \quad , & (I = S) \vee (|I| > k), \\ x_{|I|} \ , & (I \neq S) \wedge (|I| \leq k), \end{cases} \quad (4)$$

where $x_{|I|}$ is a non-negative constant dependent on the size of I. If $x_{|I|}$ can be arbitrarily close to zero, then it is clear that if such an adversarial example can be constructed so that the algorithm is only allowed to use the values of $p(I)$, then S cannot be found except by exhaustive search. If the algorithm returns any other solution $S_{ALG} \neq S$ with $|S_{ALG}| = k$, the approximation ratio $p(S)/p(S_{ALG})$ cannot be bounded.

We define the probabilities related to Ω_+ as follows:

$$Pr(D(B)) = \begin{cases} y, & B = [n_C], \\ 0, & \text{otherwise}, \end{cases} \quad (5)$$

where y is a real number satisfying $0 < y < 1$. Using Eq. (5) we can re-write Eq. (3) as follows:

$$p(I) = \frac{y}{y + \sum_{B \supseteq I} Pr(E(B))}. \quad (6)$$

Next, we construct the probabilities related to Ω_- using Eq. (6) so that the p values satisfy Eq. (4). By Eq. (6), in order for $p(I) = 1$ for all $|I| > k$ and $p(S) = 1$, we must have

$$Pr(E(B)) = 0\ , \quad \mathrm{B = S \vee |B| > k}. \quad (7)$$

We construct the remaining probabilities $Pr(B)$ that satisfy Eq. (4) starting from $|B| = k$ and then decreasing the size of B by 1 until $|B| = 1$. Algorithm 2 shows the steps of this process. Initially, $Pr(D(I))$ is set to y for $I = [n_C]$, while for all other I's it is set to 0 (by Eq. 5). Accordingly, $Pr(E(I))$ is set to 0, for all I's of size greater than k. At each step i, where $i = |B|$, each $Pr(E(B))$ is defined so that all sums $\sum_{B \supseteq I} Pr(E(B))$ are equal for all I's of size i. Hence, by Eq. (6) the p values $p(I)$ are also equal. The non-approximability limit $x_{|I|} \to 0$ is reached when $y \to 0$. An adversarial setting constructed by Algorithm 2 is given in Table 2. □

Algorithm 2 Construction of constraints that satisfy Equation (4).

```
CONSTRUCTION($n_C$, $S$, $y$)
Input: $n_C$, the number of constraints; $S$, a subset of $[n_C]$ of size $k$, where $1 < k \leq n_C$; $y$, $0 < y = Pr(D(S)) < 1$.
Output: $Z$, $Z(B) = \{Pr(E(B))\}$, the set of probabilities for all $B \subseteq [n_C]$.
Let $k \leftarrow |S|$. {It is assumed that $1 < k \leq n_C$.}
Initialize array $Z$ of size $2^{n_C}$, indexed by $B \subseteq [n_C]$. {$Z(B)$ will eventually correspond to $Pr(E(B))$, up to a normalization constant.}
Let $Z(B) \leftarrow 0$ for all $B \subseteq [n_C]$. {Initialize array to zero.}
Let $Z(I) \leftarrow 1$ for all $I \in \{B \subseteq [n_C] \mid B \neq S \wedge |B| = k\}$.
for $i = k - 1$ to 1 do
    Initialize array $Y$, indexed by $I \in \{B \subseteq [n_C] \mid |B| = i\}$.
    Let $Y(I) \leftarrow \sum_{B \supseteq I} Z(B)$ for all $I \in \{B \subseteq [n_C] \mid |B| = i\}$. {$Y(I)$ contains the unnormalized sum $\sum_{B \supseteq I} Pr(E(B))$.}
    Let $MAX \leftarrow \max_{I \subseteq \{B \subseteq [n_C] \| B| = i\}} Y(I)$.
    Let $Z(I) \leftarrow MAX - Y(I)$ for all $I \in \{B \subseteq [n_C] \mid |B| = i\}$. {Here we modify each unnormalized $Pr(E(B))$ where $|B| = i$ so that all sums $\sum_{B \supseteq I} Pr(E(B))$ are equal for all $I$ of size $i$; hence by Equation (6) the p-values $p(I)$ are also equal.}
end for
Let $C \leftarrow \sum_{B \subseteq [n_C]} Z(B)$
Let $Z(B) \leftarrow (1 - y)Z(B)/C$ for all $B \subseteq [n_C]$ {Normalize $Z$ to be a proper probability so that $\sum_{B \subseteq [n_C]} (Pr(D(B)) + Pr(E(B))) = 1$.}
return $Z$ {$Z(B) = Pr(E(B))$.}
```

Note that Theorem 2 does not imply impossibility of finding an approximate algorithm for Problem 1. However, it shows that such an algorithm cannot be constructed using only the p values; some additional properties of the problem should be used. One such property is discussed in the following section.

3.3 Non-discrimination of constraints

We show that there exists an approximation ratio for Problem 1, if the constraints are *non-discriminant*.

Let factor $\beta(I)$ be defined as follows:

$$\beta(I) = \frac{Pr(\cap_{i \in I} C_i \mid \Omega_+)}{Pr(\cap_{i \in I} C_i)} \prod_{i \in I} \frac{Pr(C_i)}{Pr(C_i \mid \Omega_+)}. \tag{8}$$

Using Bayes rule and Eq. (1), we can re-write Eq. (8) as follows:

$$\beta(I) = p(I) p(\emptyset)^{|I|-1} \prod_{i \in I} p(\{i\})^{-1}. \tag{9}$$

Definition 1 The set of constraints in I is said to be *non-discriminant if* $\beta(I) = 1$.

In other words, a set of constraints I is non-discriminant if the ratio of the probability of the intersection of the constraints in I over the products of the probabilities of the constraints does not change when conditioned on Ω_+.

Table 2 An adversarial example generated by Algorithm 2 where the p values $p(I)$ cannot be used to infer the optimal solution S except by exhaustive search

I	$Pr(D(I))$	$Pr(E(I))$	$\sum_{B \supseteq I} Pr(E(B))$	$p(I)$
∅	0.0000	0.0000	0.9990	0.0010
{1}	0.0000	0.0000	0.4995	0.0020
{2}	0.0000	0.0000	0.4995	0.0020
{3}	0.0000	0.0000	0.4995	0.0020
{4}	0.0000	0.0714	0.4995	0.0020
{5}	0.0000	0.0714	0.4995	0.0020
{1,2}	0.0000	0.0714	0.2141	0.0046
{1,3}	0.0000	0.0714	0.2141	0.0046
{2,3}	0.0000	0.0714	0.2141	0.0046
{1,4}	0.0000	0.0000	0.2141	0.0046
{2,4}	0.0000	0.0000	0.2141	0.0046
{3,4}	0.0000	0.0000	0.2141	0.0046
{1,5}	0.0000	0.0000	0.2141	0.0046
{2,5}	0.0000	0.0000	0.2141	0.0046
{3,5}	0.0000	0.0000	0.2141	0.0046
{4,5}	0.0000	0.0000	0.2141	0.0046
{1,2,3}	0.0000	0.0000	0.0000	1.0000
{1,2,4}	0.0000	0.0714	0.0714	0.0138
{1,3,4}	0.0000	0.0714	0.0714	0.0138
{2,3,4}	0.0000	0.0714	0.0714	0.0138
{1,2,5}	0.0000	0.0714	0.0714	0.0138
{1,3,5}	0.0000	0.0714	0.0714	0.0138
{2,3,5}	0.0000	0.0714	0.0714	0.0138
{1,4,5}	0.0000	0.0714	0.0714	0.0138
{2,4,5}	0.0000	0.0714	0.0714	0.0138
{3,4,5}	0.0000	0.0714	0.0714	0.0138
{1,2,3,4}	0.0000	0.0000	0.0000	1.0000
{1,2,3,5}	0.0000	0.0000	0.0000	1.0000
{1,2,4,5}	0.0000	0.0000	0.0000	1.0000
{1,3,4,5}	0.0000	0.0000	0.0000	1.0000
{2,3,4,5}	0.0000	0.0000	0.0000	1.0000
{1,2,3,4,5}	0.0010	0.0000	0.0000	1.0000

We have used the following parameters: $n_C = 5$, $k = 3$, $y = 0.001$, and $S = \{1, 2, 3\}$

If all sets of constraints are non-discriminant, then the p value of I can be written using Eq. (9) as

$$p(I) = p(\emptyset)^{1-|I|} \prod_{i \in I} p(\{i\}). \tag{10}$$

Lemma 1 *If all sets of constraints are non-discriminant, then for any two sets of constraints $I, J \subseteq [n_C]$, $I \cap J = \emptyset$, the p values satisfy $p(I \cup J) = p(\{\emptyset\})^{-1} p(I) p(J)$.*

Proof The proof follows directly from Definition 1 and Eq. (10). □

The following theorem demonstrates that GREEDY produces the optimal solution, if the constraints are non-discriminant.

Theorem 3 *If all sets of constraints, up to size k, are non-discriminant, then algorithm* GREEDY *gives the optimal solution.*

Proof Assume that the constraints up to size k are non-discriminant. It follows from Eq. (10) that to obtain a maximal $p(I)$ we must pick k constraints that have maximal p values.

Choosing $J = \{i\}$ and using Lemma 1 we have

$$p(I \cup \{i\}) = p(\emptyset)^{-1} p(I) p(\{i\}). \tag{11}$$

Consider an iteration of GREEDY, where we have l entries in I, with $l < k$. At the next iteration, we will add the constraint with index $i \in [n_C] \setminus I$ that maximizes $p(I \cup \{i\})$.

We notice from Eq. (11) that the algorithm always picks the constraint index i with the largest $p(\{i\})$. Hence, after k iterations, GREEDY selects those k constraints that have the largest p values, which is the optimal solution. □

We can derive an approximation ratio by which the solution I_{IND} approximates the optimal solution I^*. In the following two Theorems we show an approximation ratio for the solutions found by the greedy algorithm when the constraints are not non-discriminant.

Theorem 4 *The approximate solution I_{IND} satisfies $p(I^*) \leq \alpha p(I_{IND})$, where the approximation ratio α is given by $\alpha = \max_{I \subseteq [n_C], |I|=k} \beta(I) / \min_{I \subseteq [n_C], |I|=k} \beta(I)$.*

Proof We can rewrite Eq. (9) as

$$\beta(I^*) = p(I^*) p(\emptyset)^{k-1} \prod_{i \in I^*} p(\{i\})^{-1}, \tag{12}$$

and

$$\beta(I_{IND}) = p(I_{IND}) p(\emptyset)^{k-1} \prod_{i \in I_{IND}} p(\{i\})^{-1}. \tag{13}$$

Because I_{IND} by definition contains the largest p values $p(\{i\})$, then also the product $\prod_{i \in I_{IND}} p(\{i\})$ is maximal. Using $\prod_{i \in I^*} p(\{i\}) \leq \prod_{i \in I_{IND}} p(\{i\})$ together with Eqs. (12) and (13) we obtain

$$p(I^*) \leq \frac{\beta(I^*)}{\beta(I_{IND})} p(I_{IND}) = \alpha'(I^*, I_{IND}) p(I_{IND}), \tag{14}$$

where we have used $\alpha'(I^*, I_{IND}) = \beta(I^*)/\beta(I_{IND})$. Note that $\alpha'(I^*, I_{IND})$ always satisfies $\alpha'(I^*, I_{IND}) \leq \alpha = \max_{I\subseteq[n_C],|I|=k} \beta(I)/\min_{I\subseteq[n_C],|I|=k} \beta(I)$, which proves the theorem. □

A similar approximation ratio can be defined for I_{GREEDY}.

Theorem 5 *The approximate solution* I_{GREEDY} *satisfies* $p(I^*) \leq \alpha p(I_{GREEDY})$, *with the approximation ratio* α *given by* $\alpha = \max_{I\subseteq[n_C],|I|=k} \beta(I)/\min_{I\subseteq[n_C],|I|=k} \beta(I)$.

Proof This is a direct consequence of the fact that $p(I_{IND}) \leq p(I_{GREEDY})$ and Theorem 4. □

In practice the computation of the approximation ratio α defined above may be difficult as all possible subsets of $[n_C]$ of size k need to be considered. In practical applications where the number of constraints n_C is relatively large such computation may be prohibitive.

3.3.1 Example of full non-discrimination

Let our sample space Ω be the set of all binary matrices of size $n \times m$. Assume that the null distribution is the uniform distribution of binary matrices, thus the null model is described by $Pr(\omega) = 2^{-mn}, \forall\omega \in \Omega = \{0, 1\}^{n\times m}$.

Let $\omega_{ij} \in \{0, 1\}$ denote the element in the ith row and jth column of ω, and let r_i denote the margin of row i in the input matrix ω_0, i.e., $r_i = \sum_{j=1}^{m} \omega_{0ij}$. Let the constraints correspond to fixing the row-sums (Gionis et al. 2007) of the first $n_C = \lfloor n/2 \rfloor$ rows and denote all data sets satisfying the ith constraint, $i \in [n_C]$, as $C_i \subseteq \Omega$, where

$$C_i = \left\{ \omega \in \Omega \mid \sum_{j=1}^{m} \omega_{ij} = r_i \right\}. \tag{15}$$

Then, let the test statistic be the sum of ones in the lower half of ω, i.e., $T(\omega) = \sum_{i=n_C+1}^{n} \sum_{j=1}^{m} \omega_{ij}$.

Lemma 2 *In this case the constraints are non-discriminant.*

Proof Since the relative number of data sets satisfying a constraint does not depend on the assignment of other rows, it holds that $Pr(C_i \mid \Omega_+) = Pr(C_i)$ and also that $Pr(\cap_{i\in I} C_i \mid \Omega_+) = Pr(\cap_{i\in I} C_i)$. Inserting this into Eq. (8), we find that for any set of constraints $I \subseteq [n_C]$:

$$\beta(I) = \frac{Pr(\cap_{i\in I} C_i \mid \Omega_+)}{Pr(\cap_{i\in I} C_i)} \prod_{i\in I} \frac{Pr(C_i)}{Pr(C_i \mid \Omega_+)} = \frac{Pr(\cap_{i\in I} C_i)}{Pr(\cap_{i\in I} C_i)} \prod_{i\in I} \frac{Pr(C_i)}{Pr(C_i)} = 1.$$

Thus the constraints are non-discriminant. □

It may be that in practice non-discrimination of constraints only holds for trivial problems. In fact, the requirements for non-discrimination suggest that the constraints have no interactions, while the motivation for our approach is actually to find non-redundant sets of patterns. Therefore, in the next section, we present an example where the constraints are not non-discriminant, but they become non-discriminant at the limit of infinite data.

3.3.2 Example of approximate non-discrimination

Consider the same example as in Sect. 3.3.1, however, let the row-sums of *all* rows be used as constraints, i.e., $n_C = n$. As a test statistic, we use the count of ones in the matrix:

$$T(\omega) = \sum_{i=1}^{n} \sum_{j=1}^{m} \omega_{ij}. \tag{16}$$

The constraints satisfy $Pr(\cap_{i \in I} C_i) = \prod_{i \in I} Pr(C_i)$ as in the previous section, but the constraints are not non-discriminant because they are coupled with the test statistic, i.e., $Pr(\cap_{i \in I} C_i \mid \Omega_+) = \prod_{i \in I} Pr(C_i \mid \Omega_+)$ is not satisfied, hence by Eq. (8) the non-discrimination of constraints is not satisfied either.

Lemma 3 *The constraints in this example are approximately non-discriminant, i.e., if the fraction of ones in* ω *is bound into a pre-defined interval* $[x, y]$ *where* $0 < x < y < 1$*, then* $\beta(I) \to 1$ *as* $n_C \to \infty$ *for all* $I \subseteq [n_C]$ *with* $|I| = k$*.*

Proof Using the definition of conditional probability and $Pr(\cap_{i \in I} C_i) = \prod_{i \in I} Pr(C_i)$ we can rewrite Eq. (8) as

$$\beta(I) = \frac{Pr(\cap_{i \in I} C_i \cap \Omega_+)}{Pr(\Omega_+)} \prod_{i \in I} \frac{Pr(\Omega_+)}{Pr(C_i \cap \Omega_+)}. \tag{17}$$

We define a function $Q(J)$ that is expressed in terms of constraints $J \subseteq [n_C]$ as

$$Q(J) = BC\left(\sum_{i \in [n_C] \setminus J} r_i;\ (n - |J|)m,\ \frac{1}{2} \right), \tag{18}$$

where $BC(x; n, p) = \sum_{i=x}^{n} Bin(i; n, p)$ is the cumulative binomial distribution with parameters n and p. Ω_+ is the set of binary matrices which have at least as many ones as the test matrix. The probability $Pr(\Omega_+)$ is therefore given by the cumulative binomial distribution defined above as $Pr(\Omega_+) = Q(\emptyset)$. Equation (18) can equivalently be expressed as

$$Q(J) = BC\left(x(1 - \epsilon);\ nm(1 - \phi),\ \frac{1}{2} \right), \tag{19}$$

where $x = \sum_{i \in [n_C]} r_i$, $\epsilon = \sum_{i \in J} r_i / x$, and $\phi = |J|/n$. At the limit of $n_C = n \rightarrow \infty$ when $k = |I|$ is fixed, and the fraction of ones is bounded to a pre-defined fixed interval, ϵ and ϕ go to zero. From

$$\lim_{\epsilon, \phi \rightarrow 0} BC\left(x(1-\epsilon);\ nm(1-\phi), \frac{1}{2}\right) = BC\left(x; nm, \frac{1}{2}\right) \tag{20}$$

it follows that at the limit of $n_C \rightarrow \infty$ $Q(J)$ and $Q(\{i\})$—with the above mentioned assumptions—go to $Q(\emptyset)$ as well.

Using the fact that $Pr(\cap_{i \in I} C_i \cap \Omega_+) = Q(I) \prod_{i \in I} Bin(r_i; m, \frac{1}{2})$ and $Pr(C_i \cap \Omega_+) = Q(\{i\}) Bin(r_i; m, \frac{1}{2})$ allows us to rewrite Eq. (17) as

$$\beta(I) = \frac{Q(I)}{Q(\emptyset)} \prod_{i \in I} \frac{Q(\emptyset)}{Q(\{i\})}. \tag{21}$$

When $n_C \rightarrow \infty$, $k = |I|$ is fixed, and the fraction of ones will not become arbitrarily close to one or zero (hence will be bound into a predefined interval), the cumulative distribution function obeys $Q(I) \rightarrow Q(\emptyset)$ and $Q(\{i\}) \rightarrow Q(\emptyset)$, from which it follows that $\beta(I) \rightarrow 1$. In other words, non-discrimination holds approximately when ω has a sufficient number of rows. The intuitive explanation for this is that each constraint covers only a small fraction of the test statistic. Thus, as the number of constraints grows, the effect of individual constraints to the test statistic becomes negligible. □

4 Application examples

The examples below show the applicability of our approach in practice. In the first example, given in Sect. 4.1, we show how to use the framework introduced in this paper to find a small set of itemsets that explains the data. The result is novel in the sense that we are not aware of any algorithm that would give (approximately) the same result. The second and third example show how to use the framework to derive solutions that are almost equivalent to those of well-known algorithms. In Sect. 4.2 we present a solution that is equivalent to segmentation of a time-series with quadratic cost function and in Sect. 4.3 we derive a solution that is equivalent to a form of agglomerative hierarchical clustering. In all three cases the greedy algorithm is used.

Our approach can provide insights into the known problems which are not apparent in the original formulation. For example, we can readily assess the statistical significance of the results.

4.1 Mining frequent itemsets

A major problem of frequent itemset mining is that the number of results is often very large. A number of approaches has been proposed to reduce the number of results presented to the user: presenting only maximal (Bayardo 1998), closed (Pasquier et al. 1999) or statistically significant itemsets (Webb 2007). Although these approaches can significantly reduce the number of presented itemsets, the problem remains that there are very many itemsets and these itemsets have significant correlations. The problem of redundancy can be solved easily in our approach. As an example, we study how to find a set of non-redundant itemsets using the lift of itemsets as a test statistic.

Let ω denote any $n \times m$ binary matrix and ω_0 the input data. Also, as before, $[n_C]$ indexes the set of constraints, which now correspond to itemsets. We write ω_j with $j \in \{1, \ldots, N\}$ to denote the N randomizations of ω_0. Finally, let X_i, $i \in [n_C]$ denote the i^{th} itemset. We define $support(X_i, \omega)$ as the relative number of transactions in ω that support itemset X_i and the lift of an itemset (Brin et al. 1997) is defined as usual:

$$lift(X_i, \omega) = \frac{support(X_i, \omega)}{\prod_{x \in X_i} support(x, \omega)}. \tag{22}$$

We take into account the redundancy between patterns by specifying a test statistic based on the sum of the lifts over all itemsets:

$$T(\omega) = \sum_{i=1}^{n_C} lift(X_i, \omega). \tag{23}$$

Using as test statistic the sum over the lifts allows for capturing complex dependencies that are present in the data. For example, if itemset $\{A, B, C\}$ is highly frequent, this may fully explain the frequency of some subset, such as $\{A, B\}$, but it may also explain the frequency of supersets, such as $\{A, B, C, D\}$. If such dependencies exist, then using $\{A, B, C\}$ as a constraint in the randomization will lead to other itemsets retaining their frequency and thus will have a great impact on the test statistic. That is, the impact will be much more than the lift of the itemset. Overall, this setting can be thought of as finding patterns that explain as much as possible of the total lift that is present in the data. Because lift is a good measure for finding surprising patterns, we could argue that these patterns explain as much as possible about the data. We assume that the frequencies of individual items, as well as the distribution over the number of items per transaction are known a priori, i.e., we are already aware of these properties and they are not what we what to find in the data.

We study the `paleo` data (Puolamäki et al. 2006), which contains binary information about the genus of fossils found at certain locations throughout Europe and Asia. It has been derived from the NOW database (Fortelius 2005). The data consists of 124 rows (sites) and 139 columns (genus) and about 11 % of the cells are ones. We use the itemset-swap algorithm from Hanhijärvi et al. (2009b) to generate randomized data with the same row and column margins and itemsets as constraints. Because generating datasets with the exact same frequencies is NP-Hard, the algorithm accepts frequency changes with a small probability, which is controlled using a parameter w. We choose $w = 4$, as recommended in Hanhijärvi et al. (2009b). We also used the method in Hanhijärvi et al. (2009b) for finding the number of swaps required to properly mix the matrix, and found 8000 swaps to be appropriate.

A common problem in permutation testing is that the real p value that is approximated can be very small. Using empirical p values (North et al. 2002), we often have that many p values are equal to $1/N$, where N is the number of randomizations. This is a problem in our approach, since the p values are used to discriminate between (sets of) constraints. Thus, if N is too low, many constraints can give the same p value and then we do not know which one to prefer, although the real p value associated with the constraints can in fact be very different. To overcome this problem we compute all p values using 100 randomized datasets and then estimate the mean $\hat{\mu}$ and the variance $\hat{\sigma}^2$ over the test statistics of the 100 datasets. Then, we assume that the test statistics follow a normal distribution and compute the p value of the input data using the cumulative distribution function of the normal distribution:

$$\hat{p} = 1 - \Phi\left(\frac{T(\omega_0) - \hat{\mu}}{\sqrt{\hat{\sigma}^2}}\right). \tag{24}$$

where

$$\Phi(x) = (2\pi)^{-1/2} \int_{-\infty}^{x} \exp(-t^2/2)dt \tag{25}$$

The quality of the approximation depends of course on the normality assumption being valid. An added advantage in this case is that it is very unlikely that two constraints give exactly the same p value, hence there will be few ties.

Having defined how to compute a p value for the data using constraints, we can use Algorithm 1 to find the k best constraints. The itemsets used in the test statistic are all closed frequent patterns with $lift \geq 1$ and minimum support 0.1. We exclude itemsets with only one item, because the column margins are fixed in swap randomization. We find that there are 118 such itemsets. As possible constraints we use also the same 118 itemsets. We have conducted the experiment with $k = 118$, that is, we run the greedy algorithm until all itemsets are used as constraints. Using a straightforward implementation in R, each step takes ~1.5 h using one core on a 2.8 GHz Intel Core 2 processor.

The results for $k = 20$ can be found in Table 3 and the p values up to $k = 118$ in Fig. 1. We find that the p value of the data is initially very low $\hat{p} = 10^{-299.94}$ and that after 17 itemsets the p value hardly increases. This can be due to the fact that there is a lot of structure in the data, but also because of the itemset frequencies being preserved only approximately in the randomization. As can be seen in Fig. 1, the p value rapidly increases with the first four constraints and increases only marginally after 17 itemsets. From Table 3 we find that the itemsets used as constraints are not the itemsets with highest lift, which shows that the correlations between the itemset frequencies are indeed taken into account in a meaningful way. Thus, we could argue that the first four itemsets are the most informative patterns and that the first 17 itemsets explain everything there is to find in the data (with respect to the test statistic).

To give some example of the genus involved in the itemsets, we list the first four itemsets:

1. Anchitherium, Aceratherium, Dryopithecus
2. Anchitherium, Dicerorhinus, Aceratherium, Parachleuastochoerus
3. Dicerorhinus, Miotragocerus
4. Dorcatherium, Eotragus

Table 3 The twenty most informative itemsets in the paleo data selected by Algorithm 1, with respect to the sum of the lift over all itemsets

For each itemset we give the p value after imposing it as a constraint, along with the support and lift of the itemset. The initial p value of the data is $10^{-299.94}$. We find that the first four itemsets are far most informative and that using 17 itemsets as constraints gives the highest p value. The p values are all quite low due to the randomization preserving the itemset frequencies only approximately

No.	Constraint	$\hat{p}(log_{10})$	Support	Lift
1	29 58 60	−109.38	0.19	5.57
2	29 57 58 59	−79.76	0.10	19.47
3	57 64	−56.46	0.10	2.48
4	16 19	−42.75	0.12	5.17
5	107 109	−43.81	0.11	3.62
6	19 22	−44.78	0.11	4.57
7	69 75	−41.27	0.19	2.17
8	87 95	−39.51	0.11	2.23
9	87 91	−37.28	0.10	3.44
10	47 60	−38.09	0.13	1.77
11	19 22 27	−34.81	0.10	16.44
12	60 69 75	−32.20	0.11	4.69
13	68 72	−28.97	0.11	3.05
14	47 58	−29.80	0.15	1.99
15	111 117	−30.05	0.11	3.40
16	60 70	−27.98	0.10	1.92
17	60 69 72	−26.14	0.10	5.95
18	60 75	−28.59	0.15	1.56
19	29 64	−27.24	0.15	1.82
20	57 60	−27.48	0.16	2.73

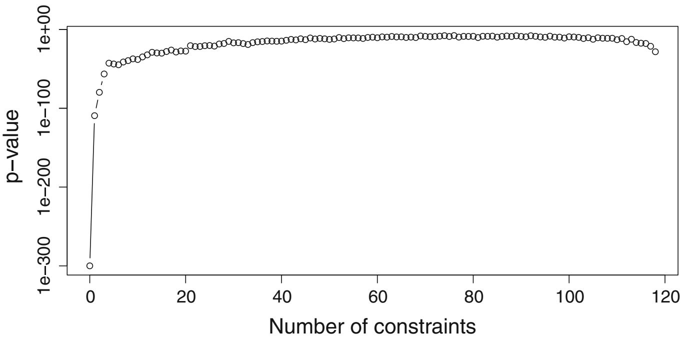

Fig. 1 P values for the paleo data when increasing the number of constraints. We observe that the p value increases rapdidly with the first four constraints, indicating that the four corresponding itemsets (sets of genus) explain most of the data. We also observe that the p value does not approach one even when all constraints are imposed, which is due to the itemset frequencies being retained only approximately in the randomization.

All of the species are long-lived genera that are associated to Western European closed habitats or forests. Careful analysis showed that the 118 itemsets included in the test statistic cover only part of the data, which suggests that a lower support threshold would perhaps lead to more interesting results for the domain experts. However, the first four itemsets seem be a good summary of the patterns included in the test statistic. The full list of genus corresponding to columns can be found in Puolamäki et al. (2006).

4.2 Time series segmentation

One very typical task in time series analysis is segmentation (Bellman 1961). Let $\omega = \omega_1 \ldots \omega_n, \omega_i \in \mathbb{R}$, be a time series. A k-segmentation S_k of ω is a partition of ω into k non-overlapping contiguous segments $S_k(\omega) = s_1 \ldots s_k$ defined by a set of splitpoints $B = \{\beta_0, \beta_1, \ldots, \beta_k\}$, with $\beta_0 = 0, \beta_1, \ldots, \beta_{k-1} \in [1, \ldots, n-1]$, $\beta_k = n$, and $\beta_i < \beta_j$ if $i < j$. Each s_i is a segment, i.e., a subsequence of ω, with $s_i = \omega_{\beta_{i-1}+1} \ldots \omega_{\beta_i}$, for $i \in \{1, \ldots, k\}$. Each point in s_i is collapsed to a representative value μ_{s_i}. In the remainder of this section μ_{s_i} denotes the mean value of the points in segment s_i.

Then, given a time series ω, the k-segmentation problem is to minimize the loss in accuracy induced by collapsing the original points in ω into k representatives,

$$L(\omega, S_k) = \sum_{s_i \in S_k} \sum_{j \in s_i} \left(\omega_j - \mu_{s_i}\right)^2 .$$

Given the set $\mathbb{S}_k(\omega)$ of all possible k-segmentations of ω, we define the optimal k-segmentation as

$$S_{OPT} = \arg\min_{S_k \in \mathbb{S}_k(\omega)} L(\omega, S_k). \tag{26}$$

We claim that the k-segmentation problem can be solved using our approach. It suffices to describe a time series randomization method, the set of constraints that will be used, and the test statistic.

The sample space Ω consists of all permutations of the original time series. The randomization can be performed efficiently by permuting the original time series uniformly at random, which defines the null probability function Pr. We impose $k-1$ constraints $C_i, i \in [k-1]$, by introducing $k-1$ splitpoints $\{\beta_1, \ldots, \beta_{k-1}\}$. Note that β_0 and β_k are by definition fixed to 0 and last position of ω respectively. The imposed constraints define segments in ω and allow only for permutations within the same segment. For example, given the original time series ω, a constraint is imposed by defining a split point $s \in [n-1]$ such that the randomization method is not allowed to swap points between segments $\omega_1 \ldots \omega_s$ and $\omega_{s+1} \ldots \omega_n$.

The test statistic—which essentially measures autocorrelation—is defined as follows:

$$T(\omega) = -\frac{1}{2} \sum_{i=1}^{n-1} (\omega_{i+1} - \omega_i)^2. \tag{27}$$

Lemma 4 *Let $X_i, i \in [2n]$, be i.i.d. random variables with zero expectation $E(X_i) = 0$. Define a random variable $S = \sum_{i=1}^{n} X_i X_{n+i}$. At the limit $n \to \infty$, S follows a Gaussian distribution with mean zero and variance $\sigma^2 = \sum_{i \in [n]} E(X_i^2) E(X_{n+i}^2)$*

Proof Let $Z_i = X_i X_{n+i}$ be a random variable. Then, S can be expressed as $S = \sum_{i=1}^{n} Z_i$. Since all X_i are independent, we have $E(Z_i) = 0$ and $E(S) = 0$.

The variance of Z_i is given by $E(Z_i^2) - E(Z_i)^2 = E(Z_i^2) = E(X_i^2)E(X_{n+i}^2)$. By the central limit theorem, at the limit $n \to \infty$, S follows a Gaussian distribution with variance $\sigma^2 = \sum_{i=1}^{n} E(X_i^2)E(X_{n+i}^2)$. □

Next we show that our approach solves the k-segmentation problem at the limit of a long time series.

Theorem 6 *The segmentation problem of Eq.* (26) *is equivalent to Problem 1 when the sample space* Ω, *the null probability function* Pr, *and the constraints are defined as above, at the limit of a long time series* $n \to \infty$, *with fixed* k, *and when the segments have equal variance.*

Proof Consider the limit $n \to \infty$ and let $s_i = \omega_l \ldots \omega_{l+|s_i|-1}$. Then the test statistic within segment s_i is given by

$$T(s_i) = -\frac{1}{2} \sum_{\omega_j}^{i^l + |s_i| - 1} (\omega_{j+1} - \omega_j)^2. \tag{28}$$

Next, we add and subtract the mean value μ_{s_i} of segment s_i, and hence the test statistic now becomes

$$T(s_i) = -\sum_{\omega_j \in s_i} (\omega_j - \mu_{s_i})^2 + \sum_{\omega_j \in s_i^*} (\omega_j - \mu_{s_i})(\omega_{j+1} - \mu_{s_i}) + \epsilon, \tag{29}$$

where s_i^* denotes segment s_i with the last element removed. The additional factor ϵ contains terms related to the correlation between points within the segments and effects due to segment boundaries. The relative contribution of ϵ vanishes as the number of points becomes very large, i.e., $n \to \infty$, and hence it is omitted in the remainder of this proof.

We also assume that variables ω_j within a segment are sampled i.i.d. from the empirical value distribution within a segment, which is asymptotically true for very large segments, since for very large segments the permutation and i.i.d. sampling are equal operations.

By Lemma 4, the expected value of the second term of Eq. (29) is zero. Hence, the expected value of the test statistic under randomization within segment s_i is given by the first term which is constant under the randomization and hence has zero variance,

$$E\left(T(s_i)\right) = -\sum_{\omega_j \in s_i} \left(\omega_j - \mu_{s_i}\right)^2. \tag{30}$$

Now, assuming a k-segmentation on ω, $S_k(\omega)$, the full test statistic for ω is given by

$$T(\omega) = \sum_{s_i \in S_k} T(s_i). \tag{31}$$

Using Eqs. 30 and 31, the expected value of the full test statistic is therefore

$$E\left(T(\omega)\right) = -L(\omega, S_k), \tag{32}$$

where S_k is the segmentation that corresponds to the chosen constraints.

The variance of the test statistic given by Eq. (29) is due to the second term of the equation, since the first term is invariant under randomization where the values of ω_j are permuted within a segment. The variance of a segment is given by

$$\sigma_i^2 = \sum_{\omega_j \in s_i} (\omega_j - \mu_{s_i})^2/|s_i|. \tag{33}$$

Since we assume the ω_j's are independent of each other, we also have that the terms of the form $X_j = \omega_j - \mu_{s_i}$ are independent of each other, thus, by Lemma 4, the variance of the second term of Eq. (29) is equal to $|s_i|(\sigma_i^2)^2$. Hence, the test statistic under a segmentation S_k follows a Gaussian distribution (at the limit $n \to \infty$) as follows:

$$T(\omega) \sim N\left(\mu = -L(\omega, S_k) = -\sum_{s_i \in S_k} |s_i|\sigma_i^2, \sigma^2 = \sum_{s_i \in S_k} |s_i|(\sigma_i^2)^2\right). \tag{34}$$

Assuming that the variances of all segments are constant, i.e., $\sigma_i^2 = \sigma_0^2, \forall s_i$, then we have

$$L(\omega, S_k) = \sum_{s_i \in S_k} |s_i|\sigma_i^2 = \sigma_0^2 \sum_{s_i \in S_k} |s_i| = n\sigma_0^2. \tag{35}$$

and

$$\sigma = \sqrt{\sum_{s_i \in S_k} |s_i|(\sigma_i^2)^2} = \sqrt{\sigma_i^4 \sum_{s_i \in S_k} |s_i|} = \sqrt{n}\sigma_0^2. \tag{36}$$

In other words, the expected value of the test statistic is proportional to the length of the time series $\propto n$, while the standard deviation is proportional to its square root $\propto \sqrt{n}$. Therefore, as $n \to \infty$ the standard deviation σ of the Gaussian distribution becomes negligible compared to the mean μ of the Gaussian distribution; hence, smaller μ results to a larger p value for the original time series.

Our approach will find a solution that maximizes Eq. (32) or minimizes $L(\omega, S_k)$. It follows that the solution to Problem 1 is equivalent to the solution of Eq. (26). □

The significance testing formulation of the classic segmentation problem gives insight that is not provided by just using the error function. As shown in Eq. (34), the distribution of the test statistic follows a Gaussian distribution at the limit of a long time series.

If the value of the loss function is given by $L(\omega, S)$, then by Eq. (34) the p value of the solution is given by

$$p_N(S) = \Phi\left(-\frac{T(\omega) + L(\omega, S)}{\sqrt{\sum_{s_i \in S} |s_i|(\sigma_i^2)^2}}\right), \tag{37}$$

Table 4 For each possible splitpoint of the time series (column 1), we can see the loss of Eq. (26) (column 2), the square root of the loss (column 3), the expected value of the test statistic (column 4), the standard deviation of the test statistic (column 5), and the numerical and analytic p value of Eq. (37) (columns 6 and 7)

β	$L(\omega, S_k)$	$\sqrt{L(\omega, S_k)}$	$E(T(\omega))$	$\mathrm{sd}(T(\omega))$	$p(S_k)$	$p_N(S_k)$
1	47.451	6.888	−54.457	13.001	0.004	0.000
2	47.882	6.920	−52.330	12.398	0.006	0.000
3	42.763	6.539	−47.312	11.741	0.009	0.001
4	35.290	5.941	−40.973	10.374	0.019	0.009
5	27.654	5.259	−33.393	9.019	0.072	0.102
6	32.894	5.735	−38.611	9.541	0.019	0.022
7	28.390	5.328	−34.096	8.719	0.057	0.086
8	31.470	5.610	−37.050	9.003	0.022	0.036
9	21.675	4.656	−28.203	7.274	0.154	0.413
10	22.477	4.741	−28.509	7.617	**0.157**	0.355
11	25.927	5.092	−32.833	8.565	0.078	0.165
12	**20.700**	4.550	−29.318	7.771	0.138	**0.490**
13	39.677	6.299	−45.758	11.437	0.012	0.002
14	49.039	7.003	−54.419	13.385	0.004	0.000

In both the original setting (given by the loss function) and our approach (given by $p_N(S_k)$) the optimal solution is splitting the time series at point 12. It can be seen that splitting the time series at point 9 is equally good, according to our approach (given by $p(S_k)$). An advantage of our approach in this particular application example is that we can readily find the p value associated with the solution since there exists an analytical form ($p_N(S_k)$) to compute it

The smallest loss and the largest p-values are shown in bold font

where Φ is the cumulative distribution function of the Gaussian distribution (see Eq. 25), ω is the original time series, and $\sigma_i^2 = \sum_{\omega_j} \in s_i(\omega_j - \mu_{s_i})^2/|s_i|$. This can be used in regularization to pick a suitable value for k.

As a toy example, we generated a time series of length 15 where the values were drawn from a Gaussian distribution with unit variance. The generative process was defined such that the resulting time series would consist of three segments of equal length. Specifically, for the first 5 points in the time series the mean was set to 2, for the next 5 points it was set to 0, and for the last 5 points it was set to −2. Hence, ω was synthetically generated so as to contain three segments: $s_1 = \omega_1 \ldots \omega_5$, $s_2 = \omega_6 \ldots \omega_{10}$, and $s_3 = \omega_{11} \ldots \omega_{15}$. The generated time series was the following:

$$\omega = (3.4, 1.4, 2.4, 2.6, 2.4, -0.1, 1.5, -0.1, 2.0, -0.1, -0.7, 0.3, -3.4, -2.3, -2.1).$$

We studied the segmentation problem for this time series using $k = 2, 3$ with exhaustive search.

Due to the generative process, we expect that the optimal 3-segmentation will result to the following set of splitpoints: $\{0, 5, 10, 15\}$. Table 4 shows the resulting 2-segmentation. We compare the optimization of the cost function given by Eq. (26) (column 2) with our approach (columns 6 and 7). Note that for our approach we report two p values: $p(S_k)$ corresponds to the numerical solution using randomization and $p_N(S_k)$ corresponds to the analytical solution given by Eq. (37). Specifically, for

$k = 2$, the classic time series approach and analytic approach of Eq. (37) produce a segmentation with splitpoints {0, 12, 15} while our approach produces the following splitpoints {0, 10, 15}. However, both approaches produce the same 3-segmentation {0, 5, 12, 15}. The p values reported by our approach are 0.16 and 0.79, respectively. In addition, our approach reports $p(\emptyset) = 0.002$, which is the p value of the data without imposing any constraint. From the p values we can conclude that already the 2-segmentation lifts the p value over the significance threshold and therefore a 3-segmentation may already result to overlearning. Also, Theorem 6 merely states that the methods are equivalent at the limit $n \to \infty$, so for short sequences we expect that there would be some differences in the results. Still, Table 4 shows that the numerically computed p value $p(S)$ is in rough agreement with the p value $p_N(S)$ computed analytically using Eq. (37).

The experiment was repeated using the same generative distribution but now the resulting time series was four times as long ($n = 60$). Again, three segments of equal length were "planted" into ω with segment boundaries at positions 20, 40, and 60, respectively. Both methods produced the same 2-segmentation, with splitpoints {0, 20, 60}, and a p value equal to 0.015. The optimal 3-segmentation is {0, 17, 35, 60} for both methods, with a p value of 0.78. For this time series, using a 3-segmentation makes sense, because according to the p value the 3-segmentation explains the data while for the 2-segmentation the p value is below the 0.05 confidence threshold.

Notice that in practice, when the time series is long enough, the segmentation problem of Eq. (26) can be solved using standard methods such as dynamic programming and the p value that can be used to control against overlearning can be computed using Eq. (37); hence, no randomizations are necessary.

4.3 Agglomerative hierarchical clustering

Finally, we consider the problem of agglomerative hierarchical clustering. We show that the solution to Problem 1 using the definitions above and the greedy algorithm (Algorithm 1) is equivalent to a form of agglomerative clustering.

Let the data set be $\omega \in \mathbb{R}^{n\times m}$ and consider clustering the rows of the matrix. We assume for simplicity of notation that each row of ω has zero mean and unit variance.

In agglomerative clustering each row is initially assigned to its own cluster. At each iteration the two clusters that have highest similarity are merged. The algorithm continues until some convergence criterion, such as a given number of clusters, is met.

The agglomerative clustering problem can be formulated and solved using our approach as follows. The randomization is performed by permuting each row of the data matrix uniformly at random. This defines the null probability function Pr and the sample space Ω. Let the constraints C_{ij} correspond to the $n(n-1)/2$ pairs of rows. Imposing a constraint C_{ij} means that rows i and j are permuted with the same permutation in the randomization. Initially each row is in its own cluster. Two clusters are merged if a constraint is imposed that involves rows in both clusters.

Let us denote by P the partition of rows into clusters. Initially, all rows are assigned to individual clusters, i.e., $P = \{\{1\}, \ldots, \{n\}\}$. At each iteration of the greedy algorithm we merge two clusters that increase the p value most.

We observe that each of the elements of the randomized matrix is sampled from a distribution with zero mean and unit variance, because the elements of any row have a zero mean and unit variance.

The test statistic is given by

$$T(\omega) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{m} \omega_{ik}\omega_{jk}. \tag{38}$$

Let us assume, for simplicity, that m is very large in which case we can use the central limit theorem as before, and the test statistic of Eq. (38) obeys under null hypothesis a Gaussian distribution to a good accuracy. The test statistic is a sum of terms, and to find the distribution of the test statistic under null hypothesis it is sufficient to count the contribution of the terms into the mean and variance of the distribution.

The contribution of the cross-correlation term in Eq. (38) involving rows i and j depends on whether or not the rows are in the same cluster. The cross-correlation term reads

$$T_{CC}^{ij}(\omega) = \sum_{k=1}^{m} \omega_{ik}\omega_{jk}.$$

If the rows i and j are in different clusters then the expectation of the cross-correlation term is zero, $E(T_{CC}^{ij}(\omega)) = 0$, and by Lemma 4 the variance is $VAR(T_{CC}^{ij}(\omega)) \approx m$. If the rows i and j are in the same clusters then the expectation of the cross-correlation term is given by

$$E(T_{CC}^{ij}(\omega)) = \sum_{k=1}^{m} \omega_{ik}\omega_{jk},$$

but the variance is zero $VAR(T_{CC}^{ij}(\omega)) = 0$.

Summarizing, the expectation of the test statistic reads,

$$E(T(\omega) = \frac{1}{2} \sum_{R\in P} \sum_{i\in R} \sum_{j\in R\setminus\{i\}} \sum_{k=1}^{m} \omega_{ik}\omega_{jk}, \tag{39}$$

which has a contribution from all pairs of rows that are in the same cluster. The variance of the test statistic under the null hypothesis reads

$$VAR(T(\omega)) = \frac{1}{2} \sum_{R\in P} \sum_{i\in R} \sum_{j\in[n]\setminus R} m = \frac{1}{2} m \sum_{R\in P} |R|(n-|R|), \tag{40}$$

i.e., there is a contribution of all pairs of rows that are not in the same cluster

The Eqs. (39) and (40) determine the distribution of the test statistic at the limit of large m. Finding two clusters whose merger would increase the p value most is equivalent to finding a pair of clusters R and S in P such that

$$\frac{E(T(\omega))}{\sqrt{VAR(T(\omega))}}$$

is as large as possible after merging the clusters R and S. Because initially the variance changes relatively little when merging the clusters it is sufficient to find a merger that maximizes the increase in the expectation of Eq. (39).

In other words, under the formulation in our framework, the greedy algorithm merges clusters R and S that have highest similarity as defined by

$$\text{sim}(R, S) = \sum_{i \in R} \sum_{j \in S} \sum_{k=1}^{m} \omega_{ik}\omega_{jk}. \tag{41}$$

In other words, our formulation of is equivalent to agglomerative clustering with a cluster similarity defined by Eq. (41).

4.4 Application of the framework

The framework presented in this paper includes three choices for a user to make:

1. Define the null hypothesis
2. Define the test statistic
3. Define the constraints

We argue that each of these choices is straightforward to make for most end-users. Next, we shortly discuss each of them.

The null hypothesis expresses the background knowledge that the user has about the data, i.e., it expresses what is *not* interesting to find or explain about the data. A wide variety of null hypothesis have been discussed in the literature, for various types of data. Although in principle a null hypothesis could express any type of knowledge and our framework does not impose any restrictions, we expect that in practice it would be easier to use a null hypothesis proposed in the literature. For some examples see the discussion on the related work below.

The test statistic should quantify in a single number all the properties of the data that a user wants to have explained. For example, the sum of a statistic over a collection of itemsets (Sect. 4.1), the clustering or segmentation error, the descriptions length of the data, etc. In a straightforward scenario where a user is interested only in one type of pattern, such as itemsets, the test statistic and the constraints are closely related to each other. The test statistic and the null hypothesis together define the p value for a data set.

The constraints are perhaps most straightforward to choose, they are the patterns that we allow the algorithm to give as output. In the previous sections, we have given several examples of constraints in the settings of mining itemsets, clustering and segmentation of data.

4.5 Computing *p* values for complex null hypotheses

As explained in the previous section, there are no restrictions on the null hypothesis from the perspective of the framework. However, it is required that we can estimate the *p* value for the data under the null hypothesis. For many null hypotheses, it is difficult to compute such *p* values analytically and it can be more practical to devise a method for randomizing the data and estimating the *p* values empirically. This is the case, for example, when we are studying a binary matrix and want to assume fixed row and column margins (Gionis et al. 2007).

It should be noted that using randomization brings additional computational complexity. To be precise, the computational complexity of the whole framework is multiplied by a constant factor. The factor may, however, be large and depends on the size of the data and possibly on the number of constraints imposed. See for example Hanhijärvi et al. (2009b) for further discussion. There also exist null models that can take into account a wide variety of constraints and still allow for *p* values to be computed analytically, see, e.g., De Bie (2011b). With respect to this, our framework is general and allows the user to either choose simpler and faster null models, or more complex null models when required.

5 Related work

Permutation testing in statistics is not new. For an overview see, e.g., Good (2000) or Westfall and Young (1993). Randomization methods are practical if it is easier to devise a way of sampling from a null hypothesis rather than defining it analytically. Binary matrices have attracted great attention in the data mining community and they have been used to represent knowledge from various fields such as ecology (Zaman and Simberloff 2002). Several methods have been proposed to randomize binary matrices. The problem of randomizing binary matrices of fixed size while preserving row and column margins is studied in Gionis et al. (2007). They introduce a method based on a Markov chain with local swaps that respect the marginal distribution. Similar ideas have been discussed for graphs (Hanhijärvi et al. 2009a; Ying and Wu 2009) and for real matrices (Ojala et al. 2009), while other approaches have been studied for time series randomization (Bullmore et al. 2001; Schreiber and Schmitz 1999; Vuokko and Kaski 2011). Any randomization method that can incorporate constraints can be used in our approach.

Several approaches have been proposed to reduce the number of extracted patterns. For example, by using condensed representations, such as maximal (Bayardo 1998), closed (Pasquier et al. 1999), or non-derivable itemsets (Calders and Goethals 2007) or by mining only significant itemsets (Webb 2007). Other works have targeted further reduction of the set of patterns and accounting for redundancy, for example by ranking patterns (Mielikäinen and Mannila 2003), or using heuristics and local objective functions to mine pattern sets directly (Bringmann and Zimmermann 2009; Gallo et al. 2007), or by mining patterns iteratively (Hanhijärvi et al. 2009b).

Knobbe and Ho (2006) discuss various criteria for good pattern sets, while Vreeken et al. (2011) employ the minimum description length principle to find a good set of patterns that describe the whole data. However, the method has a very different objective: description of the whole data, instead of finding *interesting* patterns. Most related to our approach is the work by De Bie (2011a), which utilizes the Maximum Entropy principle to find a set of *interesting* and *non-redundant* patterns. Like our approach, the focus is on finding subjectively interesting patterns (De Bie 2011b). Our approach is more general in the sense that it is not restricted to the Maximum Entropy model and thus allows in principle more complex null models and patterns to be taken into account as background knowledge.

Hanhijärvi et al. (2009b) also considers the problem of taking into account previous knowledge, in the form of itemset frequencies or clusters, in assessing the significance of data mining results. We extend this work by considering how to efficiently mine a small set of results that fully explains the data, considering what we already know about the data, by defining a global *p* value on the data and using hypothesis testing on the basis of constrained null models in a more general way.

The problem definition and some of the theorems were initially published as a technical report (Lijffijt et al. 2010).

6 Conclusions

We have presented a statistical significance testing approach for mining the most informative set of patterns that describe the data. Our approach optimizes a global objective function based on a *p* value defined over the data. We presented a greedy algorithm to solve the problem and showed that the problem, in its general form, is NP-hard and cannot be approximated in polynomial time by using only the *p* values. An optimal solution can be found efficiently when the constraints are non-discriminant. We also showed that there exists an approximation algorithm if the constraints are only weakly dependent. Our contribution is not specific to any type of data, patterns, or constraints. We demonstrated the applicability and usefulness of the proposed approach in three real applications: frequent itemset mining, time series segmentation, and agglomerative clustering. Using our framework we can compute a statistical significance of a data mining results, in some cases even without randomizations. Interesting directions for future work include studying existing randomization approaches, various test statistics, and classes of constraints that would make sense in this domain.

Acknowledgments We thank Mikael Fortelius and Jussi Eronen for useful discussions and the anonymous reviewers for their helpful feedback. This work was supported by the Finnish Centre of Excellence for Algorithmic Data Analysis Research (ALGODAN) and the Academy of Finland (Project 129282)

References

Bayardo RJ Jr (1998) Efficiently mining long patterns from databases. In: Proceedings of the ACM SIGMOD international conference on management of data, pp 85–93

Bellman R (1961) On the approximation of curves by line segments using dynamic programming. Communications of the ACM 4(6): 284

Brin S, Motwani R, Ullman JD, Tsur S (1997) Dynamic itemset counting and implication rules for market basket data. In: Proceedings of the ACM SIGMOD international conference on management of data, pp 255–264

Bringmann B, Zimmermann A (2009) One in a million: picking the right patterns. Knowl Inform Syst 18(1):61–81

Bullmore E, Long C, Suckling J, Fadili J, Calvert G, Zelaya F, Carpenter A, Brammer M (2001) Colored noise and computational inference in neurophysiological (FMRI) time series analysis: resampling methods in time and wavelet domains. Human Brain Mapp 12:61–78

Calders T, Goethals B (2007) Non-derivable itemset mining. Data Min Knowl Discov 14(1):171–206

De Bie T (2011a) An information theoretic framework for data mining. In: Proceedings of the 17th ACM SIGKDD conference on knowledge discovery and data mining (KDD)

De Bie T (2011b) Maximum entropy models and subjective interestingness: an application to tiles in binary databases. Data Min Knowl Discov 23(3):407–446

Fortelius M (2005) New and old worlds database of fossil mammals (NOW). University of Helsinki. http://www.helsinki.fi/science/now/. Accessed 13 Sep 2012

Gallo A, De Bie T, Cristianini N (2007) MINI: mining informative non-redundant itemsets. In: Proceedings of the European conference on principles and practice of knowledge discovery in databases (PKDD), pp 438–445

Gionis A, Mannila H, Mielikäinen T, Tsaparas P (2007) Assessing data mining results via swap randomization. ACM Trans Knowl Discov Data 1(3):14

Good P (2000) Permutation tests: a practical guide to resampling methods for testing hypotheses. Springer, Berlin

Hanhijärvi S, Garriga GC, Puolamäki K (2009a) Randomization techniques for graphs. In: Proceedings of the SIAM international conference on data mining (SDM), pp 780–791

Hanhijärvi S, Ojala M, Vuokko N, Puolamäki K, Tatti N, Mannila H (2009b) Tell me something i don't know: randomization strategies for iterative data mining. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 379–388

Knobbe AJ, Ho EKY (2006) Pattern teams. In: Proceedings of the European conference on principles and practice of knowledge discovery in databases (PKDD), pp 577–584

Lijffijt J, Papapetrou P, Vuokko N, Puolamäki K (2010) The smallest set of constraints that explains the data: a randomization approach. Technical Report TKK-ICS-R31, Aalto University School of Science and Technology, Department of Information and Computer Science

Mielikäinen T, Mannila H (2003) The pattern ordering problem. In: Proceedings of the European conference on principles and practice of knowledge discovery in databases (PKDD), pp 327–338

North BV, Curtis D, Sham PC (2002) A note on the calculation of empirical p-values from Monte Carlo procedures. Am J Hum Genet 71(2):439–441

Ojala M, Vuokko N, Kallio A, Haiminen N, Mannila H (2009) Randomization methods for assessing data analysis results on real-valued matrices. Stat Anal Data Min 2(4):209–230

Pasquier N, Bastide Y, Taouil R, Lakhal L (1999) Efficient mining of association rules using closed itemset lattices. Inform Syst 24:25–46

Puolamäki K, Fortelius M, Mannila H (2006) Seriation in paleontological data using markov chain monte carlo methods. PLoS Comput Biol 2(2):e6

Schreiber T, Schmitz A (1999) Surrogate time series. Phys D 142:346–382

Vreeken J, van Leeuwen M, Siebes APJM (2011) Krimp: mining itemsets that compress. Data min Knowl Discov 23(1): 169–214

Vuokko N, Kaski P (2011) Significance of patterns in time series collections. In: Proceedings of the SIAM international conference on data mining (SDM), pp 676–686

Webb GI (2007) Discovering significant patterns. Mach Learn 68(1):1–33

Westfall PH, Young SS (1993) Resampling-based multiple testing: examples and methods for p-value adjustment. Wiley, New York

Ying X, Wu X (2009) Graph generation with prescribed feature constraints. In: Proceedings of the SIAM conference on data mining (SDM), pp 966–977

Zaman A, Simberloff D (2002) Random binary matrices in biogeographical ecology-instituting a good neighbor policy. Environ Ecol Stat 9(4):405–421