TKK Reports in Information and Computer Science
Espoo 2010 TKK-ICS-R31

THE SMALLEST SET OF CONSTRAINTS THAT EXPLAINS THE DATA: A RANDOMIZATION APPROACH

Jefrey Lijffijt, Panagiotis Papapetrou, Niko Vuokko, and Kai Puolamäki

Aalto University
School of Science
and Technology

TKK Reports in Information and Computer Science
Espoo 2010 TKK-ICS-R31

THE SMALLEST SET OF CONSTRAINTS THAT EXPLAINS THE DATA: A RANDOMIZATION APPROACH

Jefrey Lijffijt, Panagiotis Papapetrou, Niko Vuokko, and Kai Puolamäki

Aalto University School of Science and Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science

Aalto-yliopiston teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietojenkäsittelytieteen laitos

Distribution:
Aalto University School of Science and Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science
PO Box 15400
FI-00076 AALTO
FINLAND
URL: http://ics.tkk.fi
Tel. +358 9 470 01
Fax +358 9 470 23369
E-mail: series@ics.tkk.fi

ISBN 978-952-60-3173-6 (Print)
ISBN 978-952-60-3174-3 (Online)
ISSN 1797-5034 (Print)
ISSN 1797-5042 (Online)
URL: http://lib.tkk.fi/Reports/2010/isbn9789526031743.pdf

AALTO ICS
Espoo 2010

Abstract: Randomization methods can be used to assess statistical significance of data mining results. A randomization method typically consists of a sampler which draws data sets from a null distribution, and a test statistic. If the value of the test statistic on the original data set is more extreme than the test statistic on randomized data sets we can reject the null hypothesis. It is often not immediately clear why the null hypothesis is rejected. For example, the cost of clustering can be significantly lower in the original data than in the randomized data, but usually we would also like to know *why* the cost is small. We introduce a methodology for finding the smallest possible set of constraints, or patterns, that explains the data. In principle any type of patterns can be used as long as there exists an appropriate randomization method. We show that the problem is, in its general form, NP-hard, but that in a special case an exact solution can be computed fast, and propose a greedy algorithm that solves the problem. The proposed approach is demonstrated on time series data as well as on frequent itemsets in 0–1 matrices, and validated theoretically and experimentally.

Keywords: Hypothesis testing, randomization, significant patterns, time series, frequent patterns.

CONTENTS

1 INTRODUCTION

A fundamental problem in data mining is how to identify significant and meaningful features from noisy data. We consider the novel approach of finding a minimal description of the data using randomization and statistical significance testing.

The key idea behind randomization methods is, given the original data set, to draw sample data sets from a null hypothesis, and then compute some predefined test statistic. If the test statistic in the original data set is more extreme than in the randomized samples, then we can claim that we have found a significant property of the data. However, this does not necessarily fully explain the data.

For example, suppose we are performing k-means clustering. A natural choice would be to use the k-means cost function as the test statistic. Non-random data is expected to have some structure, resulting in a lower clustering cost, as opposed to random data. Hence, it is highly likely to identify the clustering solution as significant (see, e.g., [22]). Therefore, the significance of the clustering solution does not really tell us anything new about the data—it would in fact be much more surprising, if the null hypothesis would not be rejected!

A better approach is to discover a set of patterns in the data, and then test for the significance of each individual pattern. For example, if we are dealing with 0–1 data and frequent itemsets, we can use the frequencies of the itemsets as the test statistic and compute the p-values for each itemset separately. Nonetheless, there are at least two problems in this approach: first is that of multiple hypothesis testing (see the discussion in [4, 12]), and second is that we may end up with a huge number of correlated patterns. We propose a solution for the latter problem.

The main contributions of this paper include:

- a framework that uses a global objective function to find a minimal set of constraints,
- a simple and efficient greedy algorithm to find the smallest set of constraints that explains the data statistically,
- a theoretical validation of the proposed framework and algorithm by the analysis and proof of non-trivial theoretical properties, such as NP-hardness and independence of constraints, and
- an experimental evaluation in two different domains using real data: time series analysis and frequent itemset mining.

The objective function is a p-value for the whole data set. The goal is to identify a minimal set of constraints to the null hypothesis such that the data is no longer significant. We can argue that this minimal set of constraints suffices to assess the statistical significance of data, since we can no longer reject the final constrained null hypothesis. The constraints correspond to interesting patterns. A constraint in 0–1 data can, for example, be that the frequency of a given itemset is fixed.

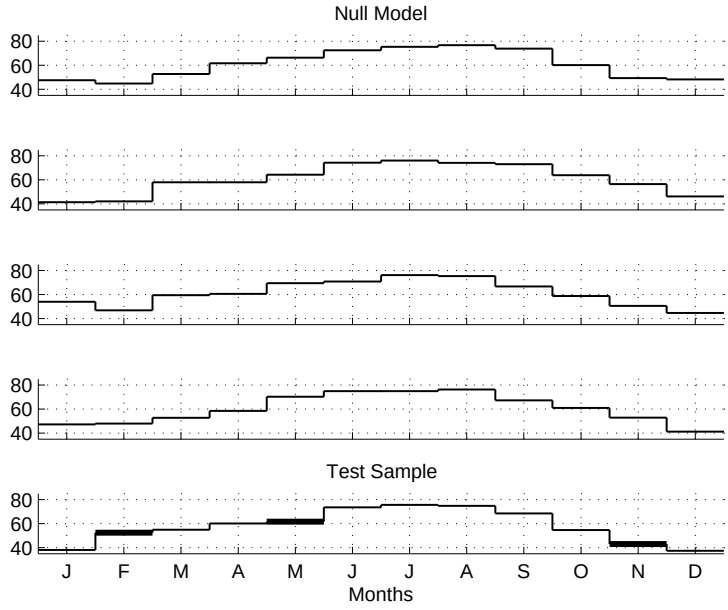

Figure 1: Five time series with monthly mean temperatures (in °F) from years 1972 to 1976. The first four time series are sampled from the null model and the one at the very bottom is the test sample, which is significantly different from the null model. This deviation can best be explained by looking at the following set of months: $\{February, May, November\}$.

This paper is structured as follows: in Section 2, we describe the framework proposed in this paper, as well as summarize the central results. In Section 3 we discuss the related work. Then, in Section 4, we present the formalism as well as derive the theorems presented in this paper. Our experimental evaluation on two application domains, one involving time series data and one involving binary matrices and frequent sets, is presented in Section 5.

2 FRAMEWORK

We develop a general framework for identifying the shortest description of a data set using a set of features. The proposed framework is general in the sense that it only needs the data features (e.g., frequent patterns, etc.), an appropriately defined test statistic, and a null hypothesis. We obtain a minimal set of features that contains a minimal set of patterns that explain the data.

In this paper, we propose a new framework that directly optimizes the global p-value of the data. This p-value corresponds to the test statistic computed from the whole data. At each iteration, a new pattern is identified, such that the global p-value is maximized. Notice that the global p-value is not for the individual patterns, but for the whole data when the patterns are used as constraints: the higher the p-value of the data the more important the constraint. To get a better insight, let us first consider an example where the data is represented by time series.

Figure 1 shows a set of four time series that have been sampled from a null model and one time series that is the test sample. Each time series contains the mean monthly temperature values (in °F) for a year. Assume we have a black box method to assess the p-value of a sample and let the significance threshold be $\alpha = 0.05$. Using the black box method, the p-value of the test sample is initially 0.017, hence we can reject the null hypothesis. Next we want to explain why the test sample (year) is significant.

Consider a set of constraints to the null hypothesis. There is a constraint for each month which requires that the respective monthly mean temperature is of the same value as in the test sample. We compute a p-value using a null hypothesis with each of these constraints. The highest p-value of 0.026 is obtained when we constrain *February*. We therefore add *February* to the result. Then we identify the next month, that together with *February*, gives the highest p-value when used as constraint. That month is *May* and in combination with *February*, it gives a p-value of 0.035. Finally, adding *November* to the result, yields a p-value of $p = 0.061 \geq \alpha$. We have reached the minimal set of months that, if used as constraints to the null model, render the test year insignificant. Hence, the set of constraints $\{February, May, November\}$ form an explanation of all significantly non-random behavior of the test sample.

With the above example we illustrated the main methodology of the proposed framework. The input of the framework therefore includes a null model, as set of predefined constraints, and a test statistic and the respective p-value that serves as the objective function. We propose exhaustive and approximate greedy algorithms to solve the optimization problem. In the exhaustive approach, all possible sets of given size are examined, and the one with the highest p-value is reported. In the greedy approach, we iteratively add constraints that cause the highest increase in the p-value of the test sample.

3 RELATED WORK

Several randomization methods exist in statistics [8, 25] that become handy when 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, e.g., ecology [29]. Several approaches have been proposed to randomize binary matrices. A Markov chain with local swaps that respect the marginal distribution is used by Gionis et al. [7], where the problem of randomizing binary matrices of fixed size while preserving row and column margins is studied. A similar idea has been discussed for graphs [10, 28], and for real matrices [22].

Iterative knowledge discovery has been the focus of interest for several researchers. Jensen et al. [14] proposes an iterative model selection approach that tests if a candidate model is better than the current model via randomization. Several approaches have been proposed (e.g., [20]) for ordering patterns based on how informative they are with respect to the data they are modeling. Similar methods for itemset mining (e.g., [6, 16, 17, 2]) and association

rules [1] have also been studied. However, all aforementioned approaches are only applicable to certain classes of patterns. Other approaches for finding significant patterns have been studied, including HMM-based methods [13] and others [26, 27], but are not directly related to the scope of this paper. Our approach and MDL [9] have similar objectives, but are based on different principles.

An iterative pattern selection approach by Hanhijärvi et al. [11] considers the problem of randomizing data by taking into account previously discovered patterns. The p-values are first computed for all patterns (e.g., all frequent itemsets of a certain size) using some null hypothesis that is then constrained by requiring that the test statistics related to the patterns with the smallest p-values are preserved. Next, the p-values for the patterns are computed using this constrained null hypothesis. This process is continued in an iterative fashion until some termination criterion is met, e.g., no patterns are significant, or when there are enough constraints; this can however take several iterations. We end up with a set of patterns that constrain the null hypothesis and explain some features of the data. There is however no global objective function. The major drawback of this approach is that we may end up with a larger than necessary set of patterns. Our framework differs from this approach in that we define and use an objective function for finding the set of significant patterns. Our objective function is chosen so that the minimal set of significant patterns is extracted. Notice this does not mean that we simply set a significance threshold, because the patterns or constraints often have interactions that are taken into account by our approach, while Hanhijärvi et al. looks only at the p-values of individual patterns.

Randomizing time series is a challenging task and several approaches have been proposed [3, 18, 23, 24]—any of these approaches can be used in our framework. There is naturally a huge literature of pattern finding methods, like SAX [15] for time series. We do not however consider them in this paper, because they are not based on statistical significance and randomization.

4 THEORY

In this section, we first provide the necessary definitions and formulations (Section 4.1), illustrate them with a simple example (Section 4.2), and present the theoretical results.

4.1 Formal Definitions

Let Ω denote the set of all possible data samples (i.e., our sample space) and $\omega_0 \in \Omega$ denote our original test sample for which the p-values will be computed. The null hypothesis is defined by a probability function Pr over the sample space Ω. We use $Pr(\omega)$, where $\omega \in \Omega$, 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 (or patterns). Each constraint is indexed in $[n_C]$ [1]. Also, let $C_i \subseteq \Omega$ (with $i \in [n_C]$) denote the set

[1] Notice the shorthand notation $[t] = \{1, \ldots, t\}$, where $t \in \mathbb{N}$.

of samples in Ω that satisfy constraint i. We require that ω_0 is also in C_i, i.e., $\omega_0 \in C_i \subseteq \Omega$. A set of constraint indices will be denoted by $I \subseteq [n_C]$. Since the proposed framework and formulation is general and can be used in different application areas and for various types of constraints, for the remainder of this section, we will only use I to refer to the indices of the constraints. We are going to actually describe these constraints in Section 5, where we demonstrate our framework for two different applications.

We assume that each data sample $\omega \in \Omega$ has a test statistic, denoted by $T(\omega) \in \mathbb{R}$. The test statistic can either be independent of the set of constraints, or it can change each time a new constraint is added.

Further, we define

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

Given a set of constraints, indexed by I, and based on the conventional definition of the p-value, the p-value $p(I)$ is defined as

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

where we have used the definition of conditional probability, and $\Omega_I = \cap_{i \in I} C_i$, with $\Omega_\emptyset = \Omega$.

4.2 Example

Consider the problem of randomizing $m \times n$ binary matrices while preserving some of the statistics of the original matrix, such as row and column margins (see, e.g., [7]). 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 $Pr(\omega) = 1/|\Omega| = 2^{-mn}, \forall \omega \in \Omega$. 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 include all $m \times n$ binary matrices for which the margins of row i are equal to the margin of row i in the test 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.

4.3 Problem Definitions

Our problem can be formally defined in two equivalent ways:

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

Problem 2 Minimal Set Problem. *For a given α, find a minimal set $I \subseteq [n_C]$ such that $p(I)$ is at least α.*

The above problem definitions correspond to the following decision problem:

Problem 3 Decision Problem. *For a given k and α, does there exist a set $I \subseteq [n_C]$ of size of at most k such that $p(I)$ is at least α?*

4.4 NP-hardness

In this section we show that our problem is NP-hard.

Theorem 4 *The Maximization Problem (Problem 1), the Minimal Set Problem (Problem 2), as well as the Decision Problem (Problem 3) are NP-hard.*

Proof It is sufficient to show that the decision problem is NP-hard. A special case of the decision problem considered here 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 $p(I) \geq 1$? We can have such a solution only if there exists a set of k constraints $I \subseteq [n_C]$, such that the intersection of C_i, for each $i \in I$, with Ω_- is the empty set. Formally, we require that

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

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

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

We take the intersection of both sides of this equation with Ω_-, resulting to

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

Denoting $T_i = \Omega_- \cap (\Omega \setminus C_i)$ we finally obtain $\cup_{i \in I} T_i = \Omega_-$. Our problem is therefore equivalent to the set cover problem over T_i where the universe is Ω_-: does there exist a set of k sets T_i such that their union is Ω_-. Problem 3 is therefore NP-hard. ∎

4.5 Algorithms to Solve Problems 1 and 2

We propose two algorithms to solve the Maximization Problem (Problem 1) and the Minimal Set Problem (Problem 2). The first two implement a straightforward exhaustive search for Problems 1 and 2, respectively, which always outputs the optimal solution. The running time of these exhaustive algorithms is $O(n_C^{|I|})$, where I is the set of constraint indices output by the algorithm. Algorithms 1 and 2 are the respective greedy algorithms.

For a fixed size k, the exhaustive algorithm solves Problem 1 by performing an exhaustive search over all sets of constraints (for which $|I| = k$) and selecting the subset with the maximal global p-value. Problem 2 can be solved similarly.

The greedy algorithm (Algorithm 1) solves Problem 1 in a greedy fashion. At each iteration, the algorithm selects the next constraint that maximizes the p-value and terminates when $|I| = k$. Finally, Algorithm 2 solves Problem 2 in a similar manner. At each iteration, the constraint that maximizes the p-value is recorded in I, until a p-value of α is reached.

Algorithm 1 The greedy algorithm for Problem 1.

GREEDY1(k) {Input: k, number of constraints. Output: I, the set of k constraint indices.}
Let $I \leftarrow \emptyset$.
while $|I| < k$ **do**
  Find $i \in [n_C] \setminus I$ such that $p(I \cup \{i\})$ is maximal.
  Let $I \leftarrow I \cup \{i\}$.
end while
return I

Algorithm 2 The greedy algorithm for Problem 2.

GREEDY2(α) {Input: α, significance threshold. Output: I, the set of constraint indices.}
Let $I \leftarrow \emptyset$.
while $p(I) < \alpha$ and $|I| < n_C$ **do**
  Find $i \in [n_C] \setminus I$ such that $p(I \cup \{i\})$ is maximal.
  Let $I \leftarrow I \cup \{i\}$.
end while
return I

Notice that all algorithms use the global p-value $p(I)$ (which is the objective function) that is defined appropriately depending on the application area. Two specific applications are described in detail in Section 5.

4.6 Independence of Constraints

In this section we define the independence of constraints, and show that both greedy algorithms produce optimal results when this independence holds. If the constraints are independent, it is actually sufficient to compute the p-values $p(\{i\})$ for each constraint and pick those k constraints with the highest p-values. This is in fact a very interesting finding, as it shows that if for some application area of interest, we can define a test statistic such that independence of constraints holds, then the greedy approach is not just a heuristic, but it can produce optimal results. Even if there is only a weak dependence between the constraints, we expect the greedy approach to produce good results.

Given Ω_+, constraints i and j are *conditionally independent*, if

$$Pr(C_i \mid \Omega_+)Pr(C_j \mid \Omega_+) = Pr(C_i \cap C_j \mid \Omega_+). \tag{2}$$

Now, assuming that constraints i and j are independent also if $\Omega_+ = \Omega$, i.e., $Pr(C_i \cap C_j) = Pr(C_i)Pr(C_j)$, and by using Bayes rule, Equation (2) can be written as follows:

$$Pr(\Omega_+ \mid C_i \cap C_j) = \frac{Pr(\Omega_+ \mid C_i)Pr(\Omega_+ \mid C_j)}{Pr(\Omega_+)}. \tag{3}$$

Expressing the result as p-values using Equation (1), we can rewrite Equation (3) as

$$p(\{i,j\}) = p(\emptyset)^{-1}p(\{i\})p(\{j\}). \tag{4}$$

Finally, we have arrived to the following definition:

Definition 5 *We call constraints i and j* independent *if the p-values satisfy $p(\{i,j\}) = p(\emptyset)^{-1}p(\{i\})p(\{j\})$, or equivalently, if $Pr(C_i \cap C_j \mid \Omega_+) = Pr(C_i \mid \Omega_+)Pr(C_j \mid \Omega_+)$, for all choices of $\omega_0 \in \Omega$. We call a set of constraints independent if all pairs of constraints in that set are independent.*

Lemma 6 *If the constraint indices in $[n_C]$ are independent, the p-values satisfy $p(I \cup J) = p(\emptyset)^{-1}p(I)p(J)$, for all sets $I \subseteq [n_C]$ and $J \subseteq [n_C]$, such that $I \cap J = \emptyset$.*

Proof Follows directly from Definition 5 and Equation (1). ∎

Theorem 7 *If the constraints are independent the greedy algorithms (Algorithms 1 and 2) give an optimal solution.*

Proof Consider the Minimal Set Problem (Problem 2), with k given as input, and the respective greedy algorithm (Algorithm 1) and assume that the constraints are independent. It follows from Lemma 6 that

$$p(I) = p(\emptyset)^{-|I|+1} \prod_{i \in I} p(\{i\}). \tag{5}$$

To get a maximal $p(I)$ we must therefore pick k constraints that have maximal p-values $p(\{i\})$. Lemma 6 can be re-written as

$$p(I \cup \{i\}) = p(\emptyset)^{-1}p(I)p(\{i\}). \tag{6}$$

Consider an iteration of the greedy algorithm (Algorithm 1), where we have l entries in I, with $l < k$. At the next iteration, we will add to I the constraint with index $i \in [n_C] \setminus I$ that maximizes $p(I \cup \{i\})$. We notice from Equation (6) that the algorithm always picks the constraint index i with the largest $p(\{i\})$. Hence, during the k iterations of adding constraints to I Algorithm 1 selects those k constraints that have the largest p-values, which is the optimal solution. The generalization to Algorithm 2 is straightforward. ∎

4.7 Relation to Sampling

A randomization method produces samples from the possibly constrained null distribution. Typically, the p-value in Equation (1) cannot be solved analytically.

In such case, the p-value of Equation (1) can be approximated with the empirical p-value [21]

$$\hat{p}(I) = \frac{1 + \sum_{i=1}^{n} H(T(\omega_i) - T(\omega_0))}{1 + n}, \tag{7}$$

where ω_i, $i \in [n]$, denotes a sample drawn from $Pr(\omega \mid \Omega \cap_{i \in I} C_i)$, and H is a step function defined by $H(t) = 1$ if $t \geq 0$, $H(t) = 0$ otherwise.

5 APPLICATIONS AND EXPERIMENTS

The framework proposed in this paper is studied and experimentally evaluated in two different application areas. The first concerns real valued observations modeled by time series and the second consists of transactional data modeled by binary matrices.

5.1 Time Series

The samples are sequences of mean monthly temperatures, each sample covering exactly one year. Our goal is to identify years that are significantly different from others and find the smallest set of months that explains this.

Experimental Setup
We study the USHCN Monthly Temperature Data [19] containing temperature measurements of 1,218 base stations all over the U.S. over the period 1895 until 2008. The maximum number of years covered by any base station is 114 and the minimum is 105, with an average of 113.9. We use only the data of the 1,139 base stations with no missing values. The mean monthly temperatures are used without any preprocessing.

Problem Setting
We are given a test sample $S_0 \in \Omega$ containing the monthly mean temperatures for one year and a station, and we want to assess how surprising this sample is. We have to specify a null model, a test statistic and a type of constraints. We define the *test statistic* as

$$T(S_i) = \sum_{j=1}^{m} |S_i(j) - \mu(j)|, \tag{8}$$

where $\mu(j)$ is the mean temperature of the jth month of all years for the station. The test statistic is simply the L_1-norm distance to the expectation.

We are interested in finding the smallest set of mean monthly temperatures that explains the deviation from the null model. Therefore, our constraints correspond to months. The jth constraint defines C_j to be set of samples for which the temperature of the jth month is $S_0(j)$, i.e., $C_j \subseteq \Omega$ contains all samples with the j^{th} month fixed to the value of the test sample. The effect of this constraint is to essentially remove the influence of that month on the test statistic.

We interpret the historical data as samples from some unknown null distribution, because it would be difficult to come up with a reasonable null model. We obtain empirical p-values by comparing one year to all others. In principle, we could obtain an empirical sample from the constrained null hypothesis by considering only those samples that satisfy the constraints. This would however be too restrictive, since the data is real valued. To remedy this, we further assume that the months in the null model are independent. To obtain samples from the constrained sample space we simply fix the temperatures of the constrained months to the respective values in the test sample. This is equivalent to assuming that all monthly temperature measurements in the null model are independent.

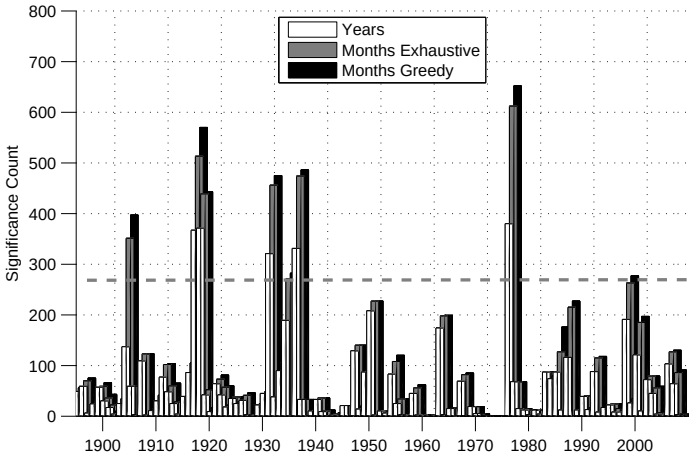

Figure 2: For each year, the number of times the year is reported as significant, as well as the number of months reported as significant, aggregated over all 1,139 base stations.

We assess the results for each base station separately and present aggregated results. We use leave-one-out for each year and try finding significant years and months in comparison to all other data of the same base station. We use both greedy and exhaustive algorithms with a fixed confidence threshold of $\alpha = 0.05$, and repeat this for each base station. We should keep in mind that, because we repeat this for all years, we expect to find approximately 5% of the years significant for each base station, even if the data is all random.

Results

We study which years and months in the USHCN data set are significantly different from normal. We also present a comparison between the greedy and exhaustive algorithm to solve this problem.

An overview of the results is given in Figure 2 which shows the number of times a year is reported as significant, aggregated over all 1,139 base stations. For each significant year and base station, we computed the smallest set of months that explains the significance. Table 1 contains the frequency distribution of the number of months needed to explain a significant year. We notice that there can often be more than one significant month, but rarely more than seven. The difference between the greedy algorithm and the exhaustive algorithm is small, because the constraints are approximately independent.

Table 1: The percentage of samples for a given number of significant months.

	Number of significant months							
Algorithm	1	2	3	4	5	6	7	8+
greedy	76.8	15.4	4.8	1.6	0.5	0.3	0.1	0.2
exhaustive	76.8	17.1	4.8	1.0	0.2	0.1	0.1	0.0

In Table 2 we find a frequency breakdown of all months in the 5 years with distinctively high counts (see Fig. 2). We see that for some years (1918, 1936) one month is reported much more frequently than all others, and for

each of the years many months are not very significant. The winter of 1917–1918, preceding the Spanish Flu epidemic, is well known to be extremely cold.

Table 2: Months reported as significant by the exhaustive algorithm for all 1,139 base stations and top 5 years.

	Year				
Month	1917	1918	1931	1936	1976
January	31	340	31	14	20
February	39	4	64	277	141
March	49	31	13	5	15
April	7	0	7	2	3
May	66	6	10	0	31
June	10	5	5	5	10
July	2	4	5	105	8
August	3	2	5	60	5
September	4	29	96	3	5
October	106	3	9	1	114
November	13	2	89	2	231
December	183	13	122	0	29
Total Months	513	439	456	474	612
Year Count	367	371	321	331	380

5.2 Binary Matrices

The second application area studied in this paper concerns binary matrices used to model transactional data. Our sample space includes only those binary matrices having the row and column margins fixed to those of the test sample. The itemset frequencies are used as constraints.

Discovering the set of frequent itemsets in the test data sample can be easily solved by traditional data mining methods. Our main target here is to identify minimal set of itemsets that describes the test sample.

Preliminary Definitions

Let D be a 0–1 matrix with m rows, corresponding to transactions, and n columns, corresponding to attributes. D_{rc} refers to the element at row r and column c of D. An itemset $X \subseteq \{1, \ldots, n\}$ indicates a set of attributes in D. A row r covers an itemset X, if $D_{rx} = 1$, for all $x \in X$. The frequency $fr(X, D)$ of an itemset X in D is the number of rows in D that cover X. Finally, the row and column margins of D, are the row and column sums of D, respectively. Given a binary matrix D, for each row $i \in [m]$ and column $j \in [n]$ in D, the margins are fixed to M_i^{row} and M_j^{col}, i.e.:

$$\forall i \in [m], \sum_{j \in [n]} D_{ij} = M_i^{row}, \quad \forall j \in [n], \sum_{j \in [m]} D_{ij} = M_j^{col}.$$

Let $\mathcal{F} = \{X_1, \ldots, X_{|\mathcal{F}|}\}$ be the set of itemsets in D with frequency above some predefined threshold, i.e., $fr(X_i, D) \geq min\_sup, \forall X_i \in \mathcal{F}$.

Problem Setting

Following the notation introduced in Section 4, the sample space Ω is the set of all binary matrices of m rows and n columns with row and column margins fixed to those of the test sample ω_0, corresponding to the binary matrix D.

The constraints for this specific application area correspond to itemset frequencies. Let $\mathcal{F}$ be the set of itemsets that are frequent in the test sample. Based again on the formulation in Section 4, each itemset $X_i \in \mathcal{F}$ constrains the original space to $C_i \subseteq \Omega$ that contains all binary matrices $\omega \in \Omega$ in which the frequency of itemset X_i equals the frequency of that itemset in the test sample.

Finally, given a binary matrix D and a set of constraints (i.e., set of frequent itemsets in $\mathcal{F}$), how to obtain the test statistic used to obtain one global p-value is explained next.

Global p-value from p-values of Patterns

We can easily compute empirical p-values for individual itemsets,

$$p_i(D) = \frac{1 + \sum_{j=1}^{n} H(fr(X_i, D_j) - fr(X_i, D))}{1 + n},$$

where D_j, $j \in [n]$, denotes the jth sample from the null distribution.

The p-values of the individual itemsets can be used to construct a global test statistic using

$$T(D) = - \min_{i \in [n_C]} p_i(D). \tag{9}$$

The global p-value can be computed by using Equations (9) and (7).

Experimental Setup

The PALEO data set [5] has been used for testing both greedy and exhaustive algorithms. The data set contains paleontological information about presence of mammals in fossil sites, represented by a binary matrix D of $m = 124$ rows (fossil sites) and $n = 139$ columns (species of mammals). The density of 1's in D is 11.5%.

The samples were generated via randomization of binary matrix D. As our constraints are itemset frequencies, we used algorithm *Itemset-Swap* described by Hanhijärvi et al. [11]. *Itemset-Swap* randomizes a binary matrix while preserving the frequencies of a given set of itemsets. Parameter w was set to 4 as suggested by [11]. The empirical p-values and test statistic T were calculated as described in Section 5.2. We generated 1000 randomized versions of D for each p-value calculation. The number of swaps K used by the *Itemset-Swap* algorithm was not fixed. At each randomization we computed the Frobenius norm [22] between the original and the swapped version of D. This norm corresponds to the number of cells in which the two matrix differ. If the difference between the randomized matrices produced by two consecutive iterations was less than 1%, swapping was terminated, as in [11].

For mining frequent itemsets, we varied the minimum support threshold and chose 8%, because it gave a reasonable number of 439 frequent itemsets. We only considered itemsets of size 2 and 3, as, in such data, it appears to be fairly easy to understand the co-occurrence of 2 or 3 variables and increasingly difficult to interpret itemsets of larger sizes.

Results

In this section, we study the performance of the greedy algorithm (Algorithm 2) and compare it with the competitor method described by Hanhijärvi et al. [11]. For the purposes of this paper, we ran Hanhijärvi et al. for the PALEO data and recorded the most significant constraint (itemset frequencies) chosen at each iteration. We evaluated the results using the global p-value used by our greedy algorithm.

Figure 3 shows a comparison of the two algorithms. The x axis corresponds to the number of constraints, i.e., number of itemset frequencies that are fixed, and the y axis shows the corresponding p-value for each set of constraints. For $\alpha = 0.05$ the greedy algorithm halts after 31 constraints. The performance for different values of α can be seen from Figure 3; notice that the greedy algorithm always outperforms the competitor method. Hanhijärvi et al. halts after 38 iterations, which means that it needs 7 more itemsets to describe the data. This is expected since the objective function used in this experiment is optimized for the greedy algorithm. Notice also that the performance of the greedy algorithm differs significantly from the method by Hanhijärvi et al. in terms of the set of constraints that are selected at each iteration. In Figure 3 it can be seen that even when the second constraint is selected, the greedy algorithm achieves a p-value of 0.0245 as opposed to 0.0113 of Hanhijärvi et al. This indicates that a different constraint has been chosen by the two algorithms: the constraint chosen by the greedy algorithm manages to increase the global p-value more than twice as much as Hanhijärvi et al. The running time of the greedy algorithm scales roughly linearly with the size of the number of constraints n_C.

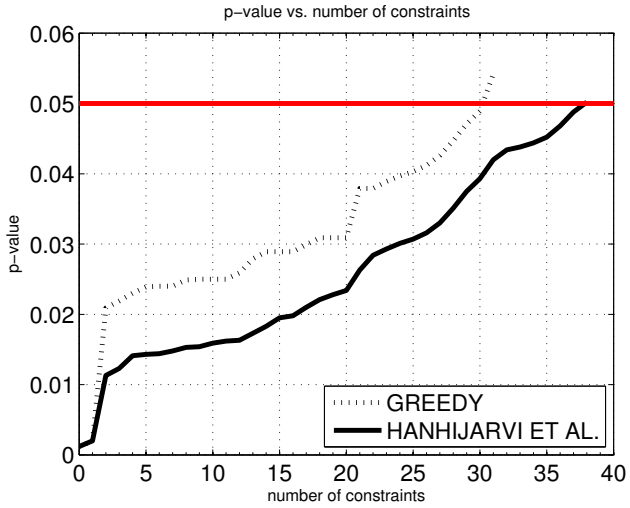

Figure 3: Global p-values for each set of constraints for a support threshold of 8%. Our greedy algorithm and Hanhijärvi et al. are compared for the PALEO data set. The greedy algorithm performs better as the p-value reaches threshold $\alpha = 0.05$ after only 31 iterations. Higher global p-value is better.

Similar results were obtained for different values of min_sup, but due to space limitations we do not include them in this section.

6 CONCLUSIONS AND FUTURE WORK

We have presented a generic framework that can be used to find a minimal description of the data, if we have a null hypothesis and a global test statistic for the full data set. We have shown that the problem is NP-hard, but that an approximate solution can be found efficiently. We have applied our framework to two distinct scenarios and have validated it experimentally, the first scenario being finding significant values in time series data, and the second finding itemsets in 0–1 data.

Our contribution is not specific to any type of data, constraints, or patterns. Our framework can be applied to scenarios where a randomization method exists but it is yet unclear how to utilize it. An interesting direction for future work would be to extend our work on time series, and study existing randomization approaches, various test statistics, and classes of constraints that would make sense in this domain.

REFERENCES

[1] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal. Mining Minimal Non-Redundant Association Rules Using Frequent Closed Itemsets. In *Computational Logic*, volume 1861. Springer, 2000.

[2] Bjorn Bringmann and Albrecht Zimmermann. The chosen few: On identifying valuable patterns. In *IEEE International Conference on Data Mining*, pages 63–72, 2007.

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

[4] Sandrine Dudoit, Juliet Popper Shaffer, and Jennifer C. Boldrick. Multiple hypothesis testing in microarray experiments. *Statistical Science*, 18(1):71–103, 2002.

[5] M. Fortelius. Neogene of the Old World Database of Fossil Mammals, 2005.

[6] Arianna Gallo, Tijl Nie, and Nello Cristianini. MINI: Mining informative non-redundant itemsets. In *European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2007.

[7] Aristides Gionis, Heikki Mannila, Taneli Mielikäinen, and Panayiotis Tsaparas. Assessing data mining results via swap randomization. *ACM Transactions Knowledge Discovery from Data*, 1(3):14, 2007.

[8] Phillip Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer-Verlag, 2nd edition, 2000.

[9] P.D. Grünwald. *The minimum description length principle*. The MIT Press, 2007.

[10] Sami Hanhijärvi, Gemma C. Garriga, and Kai Puolamäki. Randomization techniques for graphs. In *Proceedings of the 9th SIAM International Conference on Data Mining (SDM '09)*, pages 780–791, 2009.

[11] Sami Hanhijärvi, Markus Ojala, Niko Vuokko, Kai Puolamäki, Nikolaj Tatti, and Heikki Mannila. Tell Me Something I Don't Know: randomization strategies for iterative data mining. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 379–388, 2009.

[12] Sami Hanhijärvi, Kai Puolamäki, and Gemma C. Garriga. Multiple hypothesis testing in pattern discovery, 2009. arXiv:0906.5263v1 [stat.ML].

[13] S. Jaroszewicz. Interactive HMM construction based on interesting sequences. In *Proc. of Local Patterns to Global Models (LeGo'08) Workshop at the 12th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'08)*, pages 82–91, Antwerp, Belgium, 2008.

[14] David Jensen. Knowledge discovery through induction with randomization testing. In *Knowledge Discovery in Databases Workshop*, pages 148–159, 1991.

[15] E. Keogh, J. Lin, and A. Fu. HOT SAX: Efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining*, pages 226–233, 2005.

[16] Arno J. Knobbe and Eric K. Y. Ho. Maximally informative k-itemsets and their efficient discovery. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 237–244, 2006.

[17] Arno J. Knobbe and Eric K. Y. Ho. Pattern teams. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases 2006*, pages 577–584, 2006.

[18] Joseph J. Locascio, Peggy J. Jennings, Christopher I. Moore, and Suzanne Corkin. Time series analysis in the time domain and resamplingmethods for studies of functional magnetic resonance brain imaging. *Human Brain Mapping*, 5:168–193, 1997.

[19] M. J. Menne, Jr. C. N. Williams, and R. S. Vose. The united states historical climatology network monthly temperature data – version 2. *Bulletin of the American Meteorological Society*, 90:993–1107, 2009.

[20] Taneli Mielikäinen and Heikki Mannila. The pattern ordering problem. In *European Conference on Principles of Data Mining and Knowledge Discovery, Lecture Notes in Artificial Intelligence*, pages 327–338. Springer-Verlag, 2003.

[21] B. V. North, D. Curtis, and P. C. Sham. A note on the calculation of empirical p-values from Monte Carlo procedures. *The American Journal of Human Genetics*, 71(2):439–441, 2002.

[22] Markus Ojala, Niko Vuokko, Aleksi Kallio, Niina Haiminen, and Heikki Mannila. Randomization methods for assessing data analysis results on real-valued matrices. *Statistical Analysis and Data Mining*, 2(4):209–230, 2009.

[23] Thomas Schreiber. Constrained randomization of time series data. *Physical Review Letters*, 80:2105, 1998.

[24] Thomas Schreiber and Andreas Schmitz. Surrogate time series. *Physica D*, 142:346–382, 1999.

[25] Peter H. Waterfall and S. Stanley Young. *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. Wiley, 1993.

[26] Geoffrey I. Webb. Layered critical values: a powerful direct-adjustment approach to discovering significant patterns. *Machine Learning*, 71(2-3):307–323, June 2008.

[27] Goeffrey J. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.

[28] X. Ying and X. Wu. Graph generation with prescribed feature constraints. In *Proc. of the 9th SIAM Conference on Data Mining*, 2009.

[29] A. Zaman and D. Simberloff. Random binary matrices in biogeographical ecology–instituting a good neighbor policy. *Environmental and Ecological Statistics*, 9(4):405–421, 2002.

TKK REPORTS IN INFORMATION AND COMPUTER SCIENCE

TKK-ICS-R21 Sami Hanhijärvi, Kai Puolamäki, Gemma C. Garriga
Multiple Hypothesis Testing in Pattern Discovery. November 2009.

TKK-ICS-R22 Antti E. J. Hyvärinen, Tommi Junttila, Ilkka Niemelä
Partitioning Search Spaces of a Randomized Search. November 2009.

TKK-ICS-R23 Matti Pöllä, Timo Honkela, Teuvo Kohonen
Bibliography of Self-Organizing Map (SOM) Papers: 2002–2005 Addendum.
December 2009.

TKK-ICS-R24 Timo Honkela, Nina Janasik, Krista Lagus, Tiina Lindh-Knuutila, Mika Pantzar, Juha Raitio
Modeling communities of experts. December 2009.

TKK-ICS-R25 Jani Lampinen, Sami Liedes, Kari Kähkönen, Janne Kauttio, Keijo Heljanko
Interface Specification Methods for Software Components. December 2009.

TKK-ICS-R26 Kari Kähkönen
Automated Test Generation for Software Components. December 2009.

TKK-ICS-R27 Antti Ajanki, Mark Billinghurst, Melih Kandemir, Samuel Kaski, Markus Koskela, Mikko Kurimo, Jorma Laaksonen, Kai Puolamäki, Timo Tossavainen
Ubiquitous Contextual Information Access with Proactive Retrieval and Augmentation.
December 2009.

TKK-ICS-R28 Juho Frits
Model Checking Embedded Control Software. March 2010.

TKK-ICS-R29 Miki Sirola, Jaakko Talonen, Jukka Parviainen, Golan Lampi
Decision Support with Data-Analysis Methods in a Nuclear Power Plant. March 2010.

TKK-ICS-R30 Teuvo Kohonen
Contextually Self-Organized Maps of Chinese Words. April 2010.

ISBN 978-952-60-3173-6 (Print)
ISBN 978-952-60-3174-3 (Online)
ISSN 1797-5034 (Print)
ISSN 1797-5042 (Online)