Department of Information and Computer Science

Multiple Hypothesis Testing in Data Mining

Sami Hanhijärvi

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 51/2012

Multiple Hypothesis Testing in Data Mining

Sami Hanhijärvi

Doctoral dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the School of Science for public examination and debate in Auditorium AS1 at the Aalto University School of Science (Espoo, Finland) on the 11th of May 2012 at noon (at 12 o'clock).

Aalto University
School of Science
Department of Information and Computer Science

Supervisor
Prof Heikki Mannila and Prof Juho Rousu

Instructor
Dr Kai Puolamäki

Preliminary examiners
Prof Tapio Salakoski, University of Turku
Prof Martti Juhola, University of Tampere

Opponent
Prof Jaak Vilo, University of Tartu, Estonia

Aalto University publication series
DOCTORAL DISSERTATIONS 51/2012

ISBN 978-952-60-4604-4 (printed)
ISBN 978-952-60-4605-1 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Unigrafia Oy
Helsinki 2012

Finland

The dissertation can be read at http://lib.tkk.fi/Diss/

441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Sami Hanhijärvi

Name of the doctoral dissertation
Multiple Hypothesis Testing in Data Mining

Publisher School of Science

Unit Department of Information and Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 51/2012

Field of research Computer and Information Science

Manuscript submitted 15 June 2010 **Manuscript revised** 30 March 2012

Date of the defence 11 May 2012 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

Data mining methods seek to discover unexpected and interesting regularities, called patterns, in presented data sets. However, the methods often return a collection of patterns for any data set, even a random one. Statistical significance testing can be applied in these scenarios to select the surprising patterns that do not appear as clearly in random data. As each pattern is tested for significance, a set of statistical hypotheses are considered simultaneously. The multiple comparison of several hypotheses simultaneously is called multiple hypothesis testing, and special treatment is required to adequately control the probability of falsely declaring a pattern statistically significant. However, the traditional methods for multiple hypothesis testing can not be used in data mining scenarios, because these methods do not consider the problem of varying set of hypotheses, which is inherent in data mining.

This thesis provides an introduction to the problem and reviews some published work on the subject. The focus is in multiple hypothesis testing and specifically in data mining. The problems with traditional multiple hypothesis testing methods in data mining scenarios are discussed, and a solution to these problems is presented. The solution uses randomization, which involves drawing samples of random data sets and using the data mining algorithm with them. The results on the random data sets are then compared with the results on the original data set. Randomization is introduced and discussed in general, and possible randomization schemes in different data mining scenarios are presented. The solution is applied in iterative data mining and biclustering scenarios. Experiments are carried out to display the utility in these applications.

Keywords data mining, multiple hypothesis testing, statistical significance testing, biclustering

ISBN (printed) 978-952-60-4604-4 **ISBN (pdf)** 978-952-60-4605-1

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Espoo **Location of printing** Helsinki **Year** 2012

Pages 170 **The dissertation can be read at** http://lib.tkk.fi/Diss/

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Sami Hanhijärvi

Väitöskirjan nimi
Monen hypoteesin testaus tiedonlouhinnassa

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietojenkäsittelytieteen laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 51/2012

Tutkimusala Informaatiotekniikka

Käsikirjoituksen pvm 15.06.2010 **Korjatun käsikirjoituksen pvm** 30.03.2012

Väitöspäivä 11.05.2012 **Kieli** Englanti

☐ **Monografia** ☒ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

Tiivistelmä

Tiedonlouhinnan menetelmillä pyritään löytämään annetusta aineistosta yllättäviä ja mielenkiintoisia säännönmukaisuuksia, joita kutsutaan hahmoiksi. Useat menetelmät kuitenkin löytävät hahmoja kaikista aineistoista, jopa täysin satunnaisista. Näissä tilanteissa voidaan käyttää tilastollista testausta valitsemaan yllättävät hahmot, jotka eivät esiinny yhtä vahvasti satunnaisessa aineistossa. Monen hahmon tilastollista merkittävyyttä testatessa käsitellään samalla yhdenaikaisesti joukkoa tilastollisia hypoteesejä. Usean hypoteesin yhdenaikaista testausta kutsutaan monen hypoteesin testaamiseksi, joka vaatii erityistoimenpiteitä, jotta väärien johtopäätösten todennäköisyyttä voidaan hallita. Kuitenkaan tyypillisiä monen hypoteesin testausmenetelmiä ei voida käyttää tiedonlouhinnassa, koska ne eivät ota huomioon tiedonlouhinnassa tyypillistä vaihtelevan hypoteesijoukon ongelmaa.

Tämä väitöskirja esittelee ongelman ja tarkastelee aiheeseen liittyviä julkaisuja. Kirja keskittyy monen hypoteesin testaamiseen erityisesti tiedonlouhinnan tilanteissa. Tyypillisten monen hypoteesin testaamiseen käytettävien menetelmien ongelmia tiedonlouhinnassa käsitellään, ja ongelmiin esitetään ratkaisu. Tämä perustuu satunnaistukseen, jossa luodaan satunnaisia aineistoja ja käytetään tiedonlouhinnan menetelmää näihin aineistoihin. Saatuja tuloksia verrataan alkuperäisestä aineistosta saatuihin tuloksiin. Satunnaistaminen esitellään yleisesti ja käsitellään mahdollisia satunnaistamismenetelmiä erilaisissa tiedonlouhinnan tilanteissa. Esitettyä ratkaisua käytetään iteratiivisessa tiedonlouhinnassa ja kaksoisryhmittelyssä, joissa kokeellisesti myös osoitetaan ratkaisun hyöty.

Avainsanat tiedonlouhinta, monen hypoteesin testaus, tilastollinen testaus, kaksoisryhmittely

ISBN (painettu) 978-952-60-4604-4 **ISBN (pdf)** 978-952-60-4605-1

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Espoo **Painopaikka** Helsinki **Vuosi** 2012

Sivumäärä 170 **Luettavissa verkossa osoitteessa** http://lib.tkk.fi/Diss/

Contents

List of publications

1. Sami Hanhijärvi. Multiple Hypothesis Testing in Pattern Discovery. *Lecture Notes in Computer Science*, 2011, Volume 6926/2011, 122–134.

2. Sami Hanhijärvi, Markus Ojala, Niko Vuokko, Kai Puolamäki, Nikolaj Tatti, Heikki Mannila. Tell Me Something I Don't Know: Randomization Strategies for Iterative Data Mining. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 379–388, New York, NY, USA, 2009. ACM.

3. Sami Hanhijärvi, Gemma C. Garriga, Kai Puolamäki. Randomization Techniques for Graphs. In *Proceedings of the 9th SIAM International Conference on Data Mining (SDM '09)*, pages 780–791, 2009.

4. Kai Puolamäki, Sami Hanhijärvi, Gemma C. Garriga. An Approximation Ratio for Biclustering. *Information Processing Letters*, 108(2):45–49, 2008.

5. Sami Hanhijärvi, Aleksi Kallio. Biclustering Gene Expression Data with Minimum Description Length. Technical Report Aalto-ST 9/2012, Aalto University School of Science, Espoo, Finland, 2012.

Author's contribution

Publication 1 was based on the context of previous work of Dr Puolamäki, Dr Garriga and the present author, published as a technical report [41]. Publication 1 presented a different method to what was presented in the technical report, simplifying the method and providing a proof of validity with only a general assumption. The publication was completely done by the present author.

The idea in Publication 2 was developed together with all the authors. The proof for NP-hardness was done by Dr Tatti, and the experiments were carried out by the present author and Dr Ojala. The paper was written mostly by the present author, Mr Ojala, Dr Tatti, and Prof Mannila, while other authors also contributed in the writing.

The idea in Publication 3 was initiated by the present author, who also wrote most of the paper and carried out all of the experiments. The other authors contributed in the development of the ideas and wrote parts of the paper.

The original idea for the problem in Publication 4 was made by Dr Puolamäki and it was developed by all of the authors. The proof in binary case was constructed by Dr Puolamäki together with the present author, while the final step of the proof was found by Dr Puolamäki. The real case was proven by the present author. All authors contributed in writing the paper.

The initial idea for the biclustering algorithm in Publication 5 was made by Prof Mannila. However, the initial idea was further developed mostly by the present author, in co-operation with Prof Mannila, Prof Taipale, Dr Kivioja and Mr Kallio. The mathematical details and approximations were completely developed by the present author. The paper is almost completely written by the present author, with parts written by Mr Kallio, and the experiments are completely carried out by the present author.

Acknowledgments

This thesis was carried out in the Department of Information and Computer Science (ICS) at Aalto University School of Science. My work was financially supported by the Academy of Finland (National Centers of Excellence program), Finnish Doctoral Programme in Computational Sciences (FICS), and Emil Aaltonen Foundation.

It has been a privilege to work in such a distinguished and inspiring department, led by Prof Pekka Orponen. I want to thank the supervisor of this thesis, Prof Juho Rousu, for his support and contribution in finalizing this thesis. My sincere thanks also go to the reviewers of this thesis Prof Antti Räisänen and Prof Tapio Salakoski.

I want to express my gratitude to Prof Heikki Mannila for supervising most of this thesis. My sincere thanks belong also to my instructor Dr Kai Puolamäki for his contribution and guidance throughout the thesis work. I also want to thank my collaborators Dr Gemma C. Garriga, Aleksi Kallio, Dr Teemu Kivioja, Dr Markus Ojala, Prof Jussi Taipale, Dr Nikolaj Tatti and Dr Niko Vuokko. It has been a pleasure working with all of you.

I want to thank my former colleagues at the ICS, Dr Hannes Heikinheimo, Dr Jaakko Hollmén, Mr Jefrey Lijffijt, Dr Panagiotis Papapetrou and Dr Antti Ukkonen for their great and inspiring company.

My warm thanks belong go to my friends with whom I've spent countless unforgettable moments of sheer fun. I am also deeply grateful to my family for their support and encouragement throughout my life.

1. Introduction

The rapid development of computing machinery has enabled the acquisition of vast amounts of data. Measuring devices have become more accurate and able to gather numerous results simultaneously. The data storage capability has also increased exponentially, making it possible to collect and store vast amounts of data for future analysis. These aspects have motivated the development of machine learning and data mining methods. Users are no longer able to inspect and analyze the data by hand but they need computer aided tools to present them with various aspects of the data. Computers allow innumerable amount of calculations to be carried out in no time at all, previously requiring years and years of manual labor. With this development, the analysis methods have been transferred to computers, and new elaborate methods have been constructed to utilize the vast computer power available to extract all the interesting information in a data set.

Data mining methods [40, 107] include a multitude of different patterns to discover, among which are itemset mining [17, 39, 99], frequent subgraph mining [38, 49, 101], biclustering [8, 10, 19, 57, 70], to name a few. All of these methods take as input a data set and some parameters, and produce a collection of results that contain patterns found in the data. Often the patterns are somehow optimal or at least good according to some criteria. Such is the case in frequent itemset mining, where all the itemsets have a frequency of at least the predefined threshold. Frequency can be seen as a goodness measure and the algorithm finds all the good patterns according to the measure. All of the mentioned data mining methods, as well as many other methods existing today [40], search and return patterns that are good according to some measure.

However, a common and expected pattern may have a high value in the goodness measure, and therefore, a high value in the measure may

mislead the user to think the corresponding pattern is necessarily exceptional. For example in frequent itemset mining, the frequency of an itemset alone might not express how interesting the itemset is. The frequency of a large itemset is always at most the frequency of a smaller itemset that is a subset of the larger one. Even though the larger itemset may express meaningful regularities in the data, all possible subsets of the larger itemset would be ranked more interesting according to frequency, irrespective of how interesting the itemsets truly are to the user. This problem arises in many scenarios of data mining, where patterns are mined according to some measure. The measure may not convey meaningful information about the exceptionality of the pattern, but merely the fit to the defined criteria. In an application perspective of data mining methods, the user may want to have some guarantee that the patterns returned by the algorithm are somehow surprising and exceptional, before the user commits time and money to further analyze or use the results.

Statistical significance testing has long been used to find if the observed result is surprising or only a consequence of mere chance [18, 30]. The idea is to assume a distribution for the goodness measure when the result is caused by chance alone, *i.e.*, the result is not significant, and find how unexpected the observed value of the goodness measure is in that distribution. A result is assigned with a p-value, which is the probability of obtaining equal or more extreme value in the goodness measure in the assumed distribution. It is calculated from the tails of the distribution. If the p-value is large, the observed value for the goodness measure is likely according to the distribution. In that case, it is evident that the found result is not statistically significant, and is likely caused by random effects. Commonly, the p-value is compared to a user defined threshold, a confidence level. If the probability is less than that level, the result is declared statistically significant with the confidence determined by the set level. For example, if the confidence level is 5%, then a result with at most 5% p-value is considered statistically significant. The confidence level gives the largest accepted probability that the result is falsely declared statistically significant.

The use of significance testing to assess the results of data mining algorithms has been studied extensively [7, 9, 14, 50, 54, 65, 68, 95, 105, 112]. There are two main approaches: analytical definition of the distribution for the goodness of a pattern if the data were random, and sampling of random data sets and testing how good the patterns are in random data.

Analytical methods admit fast calculations but often require assumptions about the distribution for the goodness measure. Randomization methods make assumptions about the distribution of the data set instead, which may be easier to define in some cases. However, randomization often requires extensive calculations, as a large number of random data sets are used with the data mining algorithm. Efforts have been made to construct meaningful definitions of random data suitable for applied scenarios [46, 48, 77, 89, 100, 110]. For example, Gionis *et al.* [32] considered randomizing binary data matrices that had the same column and row sums as the original matrix. Such a randomization procedure creates data sets that, for example, have the same number of purchases for each client and of each product as in the original data, but the knowledge of which products clients have bought is purposefully lost.

While many methods for statistical significance testing have been proposed in the data mining literature, a very important question has been largely ignored. Data mining methods very often return a large collection of results, and if all results are tested for statistical significance simultaneously, the user needs to correct for the multiple comparisons [9, 65, 104, 105]. With increasing number of simultaneous tests, the probability of falsely declaring a result statistically significant will most likely increase. For example, if a 6-sided die is thrown repeatedly, the probability of at least one 6 among all thrown numbers increases with every throw. The same applies in statistical significance testing: if the number of statistical tests, results or likewise hypotheses, increases, the probability that at least one has a p-value less than the confidence threshold increases. This is called the *multiple hypothesis testing* problem, which requires careful consideration to limit the number of results that are falsely declared statistically significant.

Multiple hypothesis testing has been long studied in the statistics literature [11, 12, 25, 29, 45, 88, 93, 94]. A collection of methods to overcome the problem have been proposed. Many of the methods are very good, if not optimal, when the p-values can be calculated by either analytical or randomization methods. These methods alter the p-values to obtain adjusted p-values. The adjusted values are often thresholded with a given level, declaring all results statistically significant that have an adjusted p-value of at most the level. The adjustment method controls the probability or expected proportion of false declarations among all the results declared statistically significant. Some methods do not adjust the p-values but di-

rectly calculate them by means of randomization. Such procedures obtain the flexibility of randomization but also suffers from the computational cost.

Data mining scenarios have specific problems when each returned pattern is tested for statistical significance. The methods often return a varying number of patterns, depending on the data set and parameter values. For example, in frequent itemset mining, only the itemsets with high enough frequency will be returned. However, the traditional multiple hypothesis testing methods require that the set of hypotheses is fixed before the data is examined. Therefore, all the possible patterns the data mining algorithm can return need to be considered as hypotheses. The number of such pattern is often orders of magnitude larger than the number of patterns actually returned. If all of the possible patterns are assigned hypotheses and the corresponding p-values are corrected for multiple hypotheses, it is very likely that no pattern is declared statistically significant. This is because the methods lose power when the number of hypotheses increases as they try to limit the probability or expected proportion of false declarations among patterns declared significant. Therefore, data mining scenarios require special consideration when testing the statistical significance of the patterns output by a data mining algorithm.

This thesis discusses the problem of multiple hypothesis testing in data mining. The basic principles of data mining and statistical significance testing are introduced next. Chapter 2 introduces the multiple hypothesis testing problem and presents existing solutions to it. A randomization framework is reviewed in Chapter 3, which is an important part of the solution. Chapters 4 and 5 introduce two data mining scenarios that benefit from multiple hypothesis testing methods in data mining. Finally, Chapter 6 concludes the thesis with discussion.

1.1 Data mining

Data mining is the process of discovering reoccurring events, or patterns, in a given data set . A pattern may be, for instance, an association rule for Boolean data explaining that if all the variables in the antecedent are true, then the variable in the consequent is often true as well [1, 2, 59, 60]. An example of such a case is the market basket data, where buying milk and bread often implies that the customer will buy cheese. Another example is a reoccurring episode in an event sequence, which

expresses that a group of events often occur close to one other [22, 61, 62]. Finding such patterns reveals interesting structures and laws behind the data generation process.

Consider the problem of frequent itemset mining, which is used here as an example to introduce the notation. A data set given to be mined is denoted by D, which can be, for instance, a matrix, a graph or a time series. In frequent itemset mining, D is a binary matrix of size $N \times M$, where the $[M]$ columns are items and the $[N]$ rows are instances, and $[M] = \{1, \ldots, M\}$. A 1 in the matrix represents that the item is present for the respective instance. In market basket data, an item is a product, or a product class, and a row is a customer. A 1 represents that the customer has bought the respective product.

A pattern is denoted by x. In frequent itemset mining, a pattern is an itemset $x \subseteq [M]$, a collection of items. A frequent itemset is an itemset for which sufficiently many rows have 1's on all of the itemset's items. In other words, the frequency of an itemset satisfies $\mathrm{freq}(x, D) \geq \sigma$, where

$$\mathrm{freq}(x, D) = |\{i \in [N] | \forall j \in x : D_{ij} = 1\}|$$

is the number of rows where all the items of the itemset x have 1's, and σ is the minimum frequency threshold. The term D_{ij} returns the value of D on the ith row and jth column.

A data mining algorithm is denoted by a function A, which takes a data set as an input. All other inputs, such as parameters, are considered invariant and contained in the definition of A. They are later assumed to be constant between repeated calls to the algorithm. In the example, A is the algorithm that finds all itemsets that are frequent,

$$A(D) = \{x \subseteq [M] \mid \mathrm{freq}(x, D) \geq \sigma\}$$

The minimum frequency threshold σ is considered to be hard coded to A, and is therefore not an input parameter of A in this setting.

An algorithm A outputs a set of patterns $A(D) = P$, which is a subset of all the possible patterns the algorithm can output, $P \in \mathcal{P}$. The cardinality of the output set is denoted by $m = |P| = |A(D)|$. In the example, P is the collection of frequent itemsets that are found from the data set D. The space of all possible patterns $\mathcal{P}$ contains all the subsets of the items $[M]$, including $[M]$ but excluding the empty set.

1.2 Statistical significance testing

Statistical significance testing is the foundation of experimental science. The idea is to formulate a theory and experimentally gather statistical evidence that will suggest if theory is true or not. For example in medicine, an experimenter tests a drug on multiple subjects to find out if the drug is effective. Statistical tests are structured in the form of hypothesis tests, where there are two hypotheses: null and alternative [24, 73, 30]. The *null hypothesis*, denoted by H_0, represents the case when the theory does not hold and which is the assumed outcome of the experiments if the tested theory is false. If H_0 is true, no new information has been gained, other than the negative result of the theory being false. The *alternative hypothesis*, denoted by H_1, is the opposite, complementing H_0. Thus the statistical test is to gather evidence to find out if the null hypothesis H_0 should be rejected and the alternative hypothesis H_1 accepted, or not. As an example, consider the very simple case of testing whether a coin is biased or not. Each coin toss is assumed to be independent and Bernoulli distributed with the common parameter θ, which states, for instance, the probability of heads. The theory is that the coin is biased, and thus a null hypothesis corresponds to the case that both outcomes of a coin toss are equally probable, $\theta = ½$, landing heads or tails with equal probability. The alternative hypothesis is then the complement of this, stating that the coin is biased and either heads or tails is favored. The alternative hypothesis does not state which side is favored and how much. It only states that the coin is biased.

Statistical evidence is gathered to be able to decide between the two hypotheses. This is done by defining a *test statistic* that can show the difference between these two cases. In the coin example, a proper test statistic is the number of heads (or tails) z after k tosses. If the coin is fair, the test statistic follows the binomial distribution $B(k, \theta)$ with the success probability $\theta = ½$. If it is not, the test statistic still follows the binomial distribution, but with some other success probability, $\theta \neq ½$. The null hypotheses can therefore be explicitly defined as $H_0 : \theta = ½$, meaning that the alternative hypothesis is $H_1 : \theta \neq ½$. In this example, the binomial distribution is the *null distribution,* which is the distribution the test statistic follows if the null hypothesis is true. The next thing to do is to throw the coin k times and calculate the number of heads z.

The statistical evidence is represented by a p-value, which expresses

the probability of obtaining the observed or more extreme result if the null hypothesis is true,

$$p = Pr(T \leq t|H_0),$$

where T represents the random variable of the test statistic and t is its realization. In the coin example,

$$p = Pr\left(|Z - ½\mathbf{k}| \geq |z - ½\mathbf{k}| \mid H_0\right),$$

where $Z \sim B(k, ½)$ and the test statistic is centered around the expected number ½k. The actual number for the p-value is calculated from the null distribution. In the coin example, p is the two-tailed cumulative probability of the binomial distribution.

In some cases, the null distribution may be unknown or it can be impossible to integrate over it, but samples can be drawn from it. Then an empirical p-value can be calculated by drawing samples of test statistic and comparing them to the original test statistic value. Assume that there are n samples of the test statistic t^j, $j \in [n]$, from the null distribution. Given that a smaller value is more interesting, the conventional empirical p-value calculation method, as discussed in [74], is defined as

$$p = \frac{|\{j \in [n]|t^j \leq t\}| + 1}{n + 1}. \tag{1.1}$$

The added ones ensure a p-value is always larger than zero, which results in a conservative hypothesis test. A conservative test is less likely to reject the null hypothesis than a test with the true p-value, and thus is less likely to falsely declare a result statistically significant. The added ones can also be viewed as including the original test statistic value in the set of random samples of test statistic values. As the null hypothesis is assumed to be true when calculating the p-value, the original test statistic value is correspondingly considered to be drawn from the null distribution, and therefore it is reasonable to add it to the set of random samples [74].

The p-value is either returned as is or thresholded with a confidence threshold α, where the null hypothesis H_0 is rejected if the p-value is equal or less than the α threshold. If H_0 is rejected, the result is said to be statistically significant with the confidence level α. Typical values for α are $\{0.001, 0.01, 0.05\}$.

This thesis introduces statistical significance testing in data mining scenarios. Therefore, the basic notation is defined by using the example of frequent itemset mining from the previous section.

For the purposes of statistical significance testing, the patterns are assigned with goodness measures $f(x, D)$, where $x \in A(D)$ and f is a the function that measures the fit of pattern x in data set D. A smaller value is assumed to be more interesting to the user. In the frequent itemset mining example, $f(x, D)$ could be defined as negative of the number rows the pattern x fits in the data set D, or $-\mathrm{freq}(x)$. The measures of fit are used as test statistics, and therefore, the function $f(x, D)$ is later called the test statistic function. Each pattern is tested for statistical significance, and a p-value $p_x(D)$ is calculated for each pattern. If the p-values $p_x(D), \forall x \in A(D)$, are sorted, they are denoted by $p_i, i \in [m]$, where $p_1 \leq p_2 \leq \cdots \leq p_m$.

2. Multiple hypothesis testing

The problem of multiple hypothesis testing arises in situations where a collection of hypotheses are tested simultaneously. Traditional methods designed for a single hypothesis can not be used as is, since the probability of false positives is likely to increase with increasing number of hypotheses. This situation is natural in data mining scenarios, where the statistical significance of a collection of patterns is assessed simultaneously. The chapter discusses the problem of multiple hypothesis testing (MHT) in data mining scenarios.

2.1 Introduction

Consider a fixed set of m null hypotheses H_{0i} to be tested simultaneously, $i \in [m]$. The hypotheses are coupled with their respective p-values, which are calculated the same way as for a single null hypothesis. These p-values are called the *unadjusted* or *raw* p-values. If all the null hypotheses are true and independent, and the p-values are simply thresholded with α, as was done in the case of only one null hypothesis, the expected number of false positives is αm. This is because each test can be seen as a Bernoulli trial with success probability α, and the expected number of successes among m tests is αm. The α confidence level no longer expresses the maximum allowed probability of a false positive, and is less than the true probability of falsely rejecting any true null hypothesis H_{0i}. This is the problem of multiple hypothesis testing.

In the traditional setting of multiple hypothesis testing, the number of null hypotheses, m, is known in advance. However, the number of true and false null hypotheses, m_0 and m_1 respectively, $m = m_0 + m_1$, are unknown. Using these, and following the common definitions [25, 106], the errors of statistical significance testing are categorized as follows. A false

	Not declared significant	Declared significant	
True null hypotheses	U	V	m_0
Non-true null hypotheses	T	S	m_1
	$m - R$	R	m

Table 2.1. Multiple hypothesis testing. m and R are observed numbers, while all others are unknown. V is the number of Type I errors, and T is the number of Type II errors.

positive occurs when a null hypothesis is falsely rejected, declaring the alternative hypotheses true. It is also called a Type I error and the number of such errors is denoted by V. A false negative is a falsely accepted null hypothesis, or the failure to reject the null hypothesis. This error is also called a Type II error and their number is denoted by T. A summary of the errors, with the numbers for correct choices, is listed in Table 2.1. Traditionally the methods that correct for MHT control the number of Type I errors while minimizing the number of Type II errors, or in other words, maximizing the power of the test. Only such methods are considered in this thesis.

One of the most common errors to be controlled is the Family-wise Error Rate (FWER), which is defined as the probability of at least one Type I error,

$$FWER = Pr(V > 0).$$

The error rate is very intuitive and easy to understand. The most suitable applications for FWER are situations were Type II errors are not as disastrous as a Type I error. In other words, falsely rejecting a null hypothesis has much more adverse effect than failing to reject a non-true null hypothesis. Such an example is in medicine, where the condition of a patient is measured with several tests. Failing to correct for MHT may cause a false diagnose, and result in unnecessary and dangerous treatments.

FWER error has the drawback of losing power when m increases, because the probability of at least one V increases. Therefore, the methods that control FWER do not have great power with large m. A completely different error to control is the False Discovery Rate (FDR) introduced by Benjamini and Hochberg [11]. The error is the expected portion of false positives,

$$FDR = \mathbb{E}\left[\frac{V}{R}\middle| R > 0\right] Pr(R > 0).$$

This error rate is appropriate in cases where improved power is more im-

portant than few Type I errors. Such is the case when important decisions are not made based on the results of the statistical significance tests, but, for instance, the results are used to guide further studies. As such, FDR is likely to be a good error measure in data mining scenarios. For further information and a comprehensive list of common error rates, see [25].

Methods that control the Type I errors have either strong or weak control over the error. Some of the earlier works on MHT [11, 91] considered only weak control, where the Type I error is only controlled if all of the null hypotheses are true, $m = m_0$. Such a case is called *complete null hypothesis* and denoted by H_0^C. While yielding reasonable results in some cases, weak control fails to provide any guarantee for the amount of Type I errors in real situations, where both true and false null hypotheses exist, $m_0 > 0$ and $m_1 > 0$, which is called the case of *composite null hypothesis*. If a method does provide a provable guarantee for Type I error in all situations, the method is said to have strong control of the corresponding error measure. Naturally, strong control is desirable, but often very difficult to prove without assumptions about the joint distribution of the p-values.

The existing methods to control for the multiple hypothesis testing problem can be roughly categorized into two groups: adjusting p-values and resampling. The two categories are discussed next.

2.1.1 Adjusting p-values

The p-value adjustment methods assume given a set of unadjusted p-values, which are the p-values of each null hypothesis without considering the MHT problem. These p-values are then adjusted such that the user can reject all null hypothesis with an adjusted p-value of at most a given threshold. The MHT method guarantees that the corresponding MHT error is controlled at the level of the threshold.

Perhaps the most well known solution to MHT problem is the Bonferroni method, which is proven to obtain strong control of FWER [25]. The unadjusted p-values are multiplied by the number of null hypotheses,

$$\tilde{p}_i^B = \min(mp_i, 1). \tag{2.1}$$

The resulting p-values, $\tilde{p}_i$, are the adjusted p-values. In addition to being very simple, the method can be used with any dependency structure between the unadjusted p-values.

However, the Bonferroni method often lacks power because it is excessively conservative. Holm [45] proposed a sequential p-value adjustment

method that is also proven to have strong control of FWER for any dependency structure, while obtaining improved power. It is defined as

$$\tilde{p}_i^{HB} = \begin{cases} \min(mp_1, 1) & i = 1 \\ \max\left(\tilde{p}_{i-1}^{HB}, \min((m-i+1)p_i, 1)\right) & i > 1 \end{cases}, \tag{2.2}$$

where $p_1 \leq \cdots \leq p_m$. This definition can also be derived using the closed testing procedure [63].

Both of the former methods have strong control of FWER. Benjamini and Yekutieli [12] proposed a method to control FDR,

$$\tilde{p}_i^{BY} = \min_{k=i,\ldots,m} \left\{ \min\left(\frac{m \sum_{j=1}^{m} 1/j}{k} p_k, 1 \right) \right\}.$$

This method has strong control of FDR for any dependency structure between p-values. A fairly recent review of other methods is available at [25].

2.1.2 Resampling

All p-value adjustment methods either ignore the dependency structure between p-values, or assume some properties of it [25]. The former methods have less power, and the requirements of the latter may be difficult to prove. Resampling methods for MHT, as presented by Westfall and Young [106], implicitly consider almost any dependency structure and directly calculate the empirical adjusted p-values. The idea is to draw samples of test statistics from their joint distribution, and calculate empirical adjusted p-values from these test statistic values.

Resampling methods often draw samples from the joint distribution of test statistics where all null hypotheses are true, which is the case of complete null hypothesis H_0^C. It is often very difficult to consider the joint distribution of test statistics when some of the null hypotheses are true and some are false, which corresponds to the case of composite null hypothesis. However, if no assumptions are made about the joint distribution, these resampling methods have only weak control of the respective error measure, as only the case of complete null hypothesis is considered.

Strong control can be achieved if *subset pivotality* is assumed. The property is required for any resampling method that draws samples under H_0^C to achieve strong control of the respective MHT error. Let J_0 be the set of true null hypotheses, where for all $i \in J_0$ the null hypotheses H_{0i} are true and for all $i \notin J_0$ the null hypotheses H_{0i} are false. Subset pivotality states that for any $J_0 \subseteq [m]$, the joint distribution of the test statistics of

the true null hypotheses J_0 is unaffected by the truth or falsehood of the other null hypotheses [106].

Definition 1. *Subset pivotality*. The joint distribution of test statistics has the subset pivotality property if the distribution $Pr(\{T_i\}_{i \in J_0})$ is identical between the complete and composite null hypotheses, for any set $J_0 \subseteq [m]$ of the true null hypotheses.

Subset pivotality is an assumption about the dependency structure between the test statistics of the true null hypotheses. It states that the joint distribution of the test statistics for any set of true null hypotheses should not be affected by the truth or falsehood of the other hypotheses. In special situations, this assumption is automatically satisfied [81].

2.1.3 Resampling by randomization

Resampling approaches have been proposed in specific data mining scenarios. Lallich *et al*. [50] considered assessing the statistical significance of association rules. They proposed a MHT method, which is based on bootstrapping the original data set. The test statistics of the association rules in the original data set are compared to the test statistics of those association rules in the bootstrapped data sets. Finally, specific calculations are made to find out which of the association rules are statistically significant. Other authors [54, 65, 112] also use resampling and randomization of the original data set in the context of association rule mining (or similar) to adjust for multiple hypothesis testing.

The common factor of all of these is that the test statistics are not sampled directly from a distribution, as is done in traditional resampling methods, but random data sets are used and the test statistic values are calculated from the random data sets. This approach is called *resampling by randomization* in this thesis. Let Π_0 be the distribution of random data sets. The definition of Π_0 together with the definition of test statistic function f implicitly define the distribution for the test statistics. To draw samples of test statistics, it is sufficient to draw samples of data sets and calculate test statistic values from the random data sets. Therefore, Π_0 is later called the *null distribution* of data sets.

A benefit of resampling by randomization is that the user does not have to define a distribution for the test statistic, but can instead define a null distribution of data sets and use virtually any definition for a test statistic. This is often desirable, since in data mining scenarios, patterns are

often sought according to some goodness measure, and this measure is a natural choice for a test statistic. These measures vary greatly in different data mining scenarios, and it is a clear benefit if a MHT method allows for any test statistic function to be used in the statistical significance testing. Therefore, the user does not have to find a distribution for the test statistic when the null hypothesis is true. However, the user still needs to define the null distribution of data sets. Studies have been made to define meaningful null distributions [15, 31, 36, 46, 77, 86, 100]. These and other definitions, as well as randomization in general, are discussed in Chapter 3.

The biggest drawback of resampling by randomization is the added computational cost. Drawing many samples of data sets and calculating test statistic values from each may be infeasible for large data sets. However, this process is easily parallelized to be done on separate processes, and with modern multicore processors and clusters of computers, increasingly large data sets can be randomized in a reasonable time.

2.1.4 Bonferroni-style resampling

One of the simplest resampling methods is the Bonferroni-style resampling method presented in Westfall and Young [106]. The method calculates the empirical adjusted p-values, and provides a strong control of FWER if subset pivotality is assumed. The method can be used in the context of resampling by randomization, and is therefore presented here as a reference method from the existing literature.

The Bonferroni-style resampling method calculates for each pattern an empirical adjusted p-value, where the adjusted p-value is the fraction of data sets that have a pattern with an equal or smaller raw p-value,

$$\tilde{p}_x^B(D) = \frac{|\{i \in [n+1] \,|\min_{y \in \mathcal{P}} p_y(D_i) \leq p_x(D)\}|}{n+1}, \tag{2.3}$$

where D_i is the ith random data set and $p_x(D)$ is the raw p-value of pattern x in dataset D. The raw p-values can be defined analytically or calculated with Equation (1.1). The random data sets $i \in [n]$ are sampled from Π_0. The original data set is denoted by D_{n+1}, and is included as a sample from the null distribution. This guarantees that the p-values will always be strictly larger than 0. The choice is discussed in Publication 1, which follows the reasoning of North *et al.* [74].

Westfall and Young [106] assume the set of hypotheses is fixed before testing, which means that $A(D_i) = A(D)$ for all $i \in [n]$, *i.e.*, the algorithm

always outputs all the possible patterns $A(D) = \mathcal{P}$. Note that in [106], Westfall and Young also consider using the test statistics $f(x, D)$ in the calculations.

2.1.5 Varying set of hypotheses

A data mining algorithm outputs a set of patterns P for an input data set D, where P is a subset of all the possible patterns $P \subseteq \mathcal{P}$. The algorithm actually considers all possible patterns $\mathcal{P}$, but only returns some of them that meet the predefined criteria. For example, in frequent itemset mining, only the itemsets that have a frequency above a user-defined threshold are returned. However, all of the patterns in $\mathcal{P}$ are hypotheses [9, 65, 103, 104, 105], but are not taken into account in the statistical significance testing because they are not returned by the algorithm. The significance testing is biased towards interesting patterns, which also most likely have smaller p-values. If only the hypotheses corresponding to patterns in P were adjusted for MHT, the respective MHT error would not be controlled because the p-values do not marginally follow the uniform distribution over $[0, 1]$, which is an assumption of most of the existing MHT methods. If, on the other hand, all the patterns in $\mathcal{P}$ would be considered in the MHT adjustment, the test would be very conservative and most likely no pattern would be declared statistically significant due to the extremely large number of hypotheses. For example in frequent itemset mining, the set $\mathcal{P}$ is of size $2^m - 1$, where m is the number of items.

The problem is caused by the fact that the set of hypotheses is not fixed before the data is mined for patterns. If only the output patterns are tested for significance, the set of hypotheses is a random variable and this needs to be accounted for in the significance testing. This is called the problem of *varying set of hypotheses* in this thesis. However, if the user defines the set of hypotheses before looking at the data, the existing MHT methods can be used for that set of hypotheses. Webb [104] solves the problem of varying set of hypotheses by simulating this latter approach. He proposes to split the data into two. The first half is mined for patterns that define the set of hypotheses. Then, the second half is used to assess their statistical significance. With this, the problem is resolved and existing methods can be used to control for MHT error. However, Webb's method can only be used if the data set can be split into two independent parts, and there is enough data to split it. For example, a binary data set in frequent itemset mining can often be split into two parts, but splitting

a graph in frequent subgraph mining is more difficult.

Another approach by Webb [103], called *layered critical values*, is to stop the data mining at a certain level, which can greatly reduce the size of $\mathcal{P}$. For example in frequent itemset mining, only itemsets of at most a given size would be returned. The justification is that at least in frequent itemset mining, only fairly small itemsets are returned and itemsets that contain almost all items are virtually never returned. The existing MHT methods can be used with the returned patterns to control for the respective MHT error. While the method can be used in a variety of settings, it is still limited to level-wise search and the interesting patterns need to be located in the lower levels.

2.2 Multiple hypothesis testing in data mining (Publication 1)

Publication 1 extended the Bonferroni-style resampling to settings with varying set of hypotheses and proved strong control of FWER when the subset pivotality is satisfied. The contributions of the publication are discussed next in detail.

2.2.1 Resampling with varying set of hypotheses

The original Bonferroni-style resampling method in Equation (2.3) is defined for a fixed set of hypotheses and it only uses the raw p-values of the patterns in the calculations. These requirements were lifted by extending the method as follows.

Definition 2. Let D be the original dataset, D_i, $i \in [n]$, be the datasets sampled from the null distribution Π_0 and $D_{n+1} = D$. Let $f(x, D_i)$ be the test statistic associated to an input pattern $x \in \mathcal{P}$ returned by algorithm A for dataset D_i. The FWER-adjusted p-values are defined as

$$\tilde{p}_x(D) = \frac{\left|\left\{i \in [n+1] \,\middle|\, (A(D_i) \neq \emptyset) \cap \left(\min_{y \in A(D_i)} f(y, D_i) \leq f(x, D)\right)\right\}\right|}{n+1}. \tag{2.4}$$

The adjusted p-value is the fraction of datasets for which the algorithm returned a pattern with equal or smaller test statistic value. If no patterns were returned, no pattern had an equal or smaller test statistic value for that dataset.

Using these adjusted p-values, strong control of FWER is obtained.

Theorem 3. *Given that subset pivotality holds, the null hypotheses* H_{0x}

of any pattern $x \in A(D)$ *with* $\tilde{p}_x(D) \leq \alpha$ *can be rejected with the certainty that FWER is controlled at level* α.

The theorem is proven in the publication.

The user is allowed to freely choose everything in the data mining setting, namely, the algorithm A, test statistic f, null distribution Π_0, which include the definitions for possible patterns $\mathcal{P}$. However, the combination of all of these definitions is required to satisfy the subset pivotality requirement.

2.2.2 Marginal probabilities as test statistic

The calculations in Equation (2.4) ignore the identity of patterns and only consider the distribution of the smallest test statistic value among returned patterns. If the marginal distributions of the test statistic are very different for different patterns, one can consider transforming the test statistic values to a common marginal distribution. One possibility is to use a variant of Equation (1.1),

$$f(x, D_j) = \frac{|\{i \in [n+1] \,|(x \in A(D_i)) \cap (g(x, D_i) \leq g(x, D_j))\}|}{n+1}, \tag{2.5}$$

where $g(x, D)$ is the original test statistic function. Consider this transformation for a pattern x. If the pattern was not output by the algorithm, it is assumed to have a larger test statistic value than with any dataset for which the pattern was output. In other words, the transformation acts as if the pattern is only output when the test statistic value is less than some unknown threshold value. This is called the *threshold* transformation.

Another possibility is to define $f(x, D)$ as the marginal probability of the pattern having equal or smaller test statistic value with the condition that the pattern is output by the algorithm,

$$f(x, D_j) = \frac{|\{i \in [n+1] \,|(x \in A(D_i)) \cap (g(x, D_i) \leq g(x, D_j))\}|}{|\{i \in [n+1] \,|(x \in A(D_i))\}|}, \tag{2.6}$$

Conversely to the threshold transformation, in here nothing is assumed about the test statistic value of a pattern if it is not output by the algorithm. This transformation is called the *conditional* transformation. The publication considers both of these and compares them with the original version.

2.3 Summary

Multiple hypothesis testing is a problem faced when several statistical significance tests are carried out simultaneously. If the tests are not corrected for multiplicity, the number of outcomes that are falsely declared significant is not controlled.

The existing methods to control the MHT errors are divided into two categories: adjusting p-values and resampling. The adjustment methods obtain adjusted p-values by changing the original p-values. The resampling methods directly calculate the adjusted p-values without calculating the original p-values. The adjusted p-values express the error measure that is controlled by the used MHT method.

The traditional methods can not be used in data mining scenarios, because the set of hypotheses to be tested is unknown before the data is mined and can change for different data sets. Publication 1 extends a known resampling method by allowing the set of hypotheses to vary and enabling the user to freely define the statistical test. Strong control of Family-wise Error Rate is proven if the significance test setting satisfies a general property.

3. Randomization

Resampling by randomization is a part of the solution to multiple hypothesis testing in data mining scenarios. In this chapter, randomization is introduced and discussed in the context of data mining.

3.1 Introduction

Randomization is the process of drawing samples of random data sets from a null distribution of data sets Π_0. The drawn random data sets usually share some properties with the original data, such as the size and type, but are otherwise completely random. The random data sets can be used, for example, to carry out statistical significance testing as was done in the previous chapter. Other applications include assessing the robustness of an algorithm [75], guaranteeing performance in randomized algorithms [69], and privacy preservation [3, 28, 47]. Randomization provides information about the surprisal of the results with the original data set when compared to results with random data from Π_0.

The models for random data can be roughly categorized into two groups: exact and property models. An exact model is an explicit definition of the null distribution Π_0. The distribution is completely user-defined and usually admits an easy method to draw samples from it. An example is the uniform distribution over all binary matrices of size $N \times M$ with each cell independently having a 1 with probability θ. The probability θ can be arbitrarily defined by the user, and drawing samples from the final distribution is very simple. Another example is the forest fire model [51] for graphs, which defines an intuitive way of constructing graphs that are similar to ones found in social networks. The model has few parameters that need to be set, and then the procedure of randomization starts from an empty graph and continues by adding nodes and edges in a pre-

defined way. Popular exact models include the Rasch model for binary matrices [82], and the Erdös-Renyi model for graphs [72].

A property model is the uniform distribution over all data sets that share a set of predefined properties with the original data set. The data sets are sampled by randomizing the original data set. The randomization process starts from the original data set and randomly changes it many times to finally obtain a random data set. The changes maintain the predefined properties, and therefore, all the sampled data sets share those properties. This procedure implicitly defines the null distribution Π_0. An example property model is the set of binary matrices that have the same size, column sums, and row sums as the original binary matrix. The next section discusses how to draw samples of random matrices from the model. Other meaningful property models have been defined, for example, for real matrices [77], graphs [64, 111], and time series [15, 55, 86].

Randomization is one of many general resampling methods, and it is closely related to permutation tests [33]. The idea in permutation tests is to calculate the value of the test statistic for all possible rearrangements of labels on the observed data points. The aim is to find out if the effect of a treatment is statistically significantly different between treatment and control groups. Other related resampling methods include bootstrapping [27], jackknife [26], and cross-validation [80].

3.2 Property models

In a property model, a set of properties are defined that are shared by the random data sets with the original data set. Let $r(D)$ be a function that measures these properties for a data set D, and returns a real valued vector. The general problem for property models can then be defined as follows.

Problem 4. Given a data set D and a property function $r(D)$, generate a data set D_i chosen independently and uniformly from the set of data sets that satisfy $r(D) = r(D_i)$.

The problem states that the null distribution of data sets is the uniform distribution over all data sets D_i for which $r(D) = r(D_i)$.

The problem statement is very general as the definition of r can contain a variety of properties. If, for example, all the data sets should be binary matrices and of the same size as the original, r may be constructed to re-

turn a vector of size three, where the first element is 1 if the input matrix is binary and 0 otherwise, and the latter two elements contain the number of rows and columns of the matrix. With this, the null distribution can be narrowed down to contain a reasonable space of data sets. However, type and size are often not explicitly encoded as properties in r, as they are maintained by other means discussed later.

The choice for r depends on the context, since it defines the knowledge that is assumed known. For example, if the elements of r are the column sums and row sums of a binary matrix D, also called the column and row margins, then all D_i will have the same column and row margins. The test for significance then asks if the results are surprising even if the column and row margins are known. Therefore, when choosing the properties to maintain, one is also defining what is the null hypothesis to be tested.

3.3 Markov-chain Monte Carlo

The choice of $r(D)$ affects the difficulty faced when sampling data sets. If D is a binary matrix and $r(D)$ only contains the type and size of the dimensions of D, Problem 4 can be solved easily by sampling each element independently and uniformly from $\{0, 1\}$. However, if $r(D)$ introduces high correlation between elements in the data set, the problem does not admit as simple an algorithm. Consider, for example, drawing samples of binary matrices with the same size, and column and row margins, as the original matrix. Disregarding either of the margins would result in a simple randomization scheme: just shuffle the values within rows or columns of the original matrix. However, since both are required to have certain values, all elements in the matrix are correlated through the margins.

This problem can be solved by Markov-chain Monte Carlo (MCMC) sampling [5]. The procedure starts from the original data set and changes it randomly while maintaining the margins. Such a method has been proposed [13, 32] for this example, and it involves repeatedly carrying out small changes, swaps, to the matrix. A swap is illustrated in Figure 3.1. When enough swaps have been done, the resulting matrix is a random sample from Π_0.

The idea in MCMC sampling is to consider each data set as a state in a Markov-chain and a small change to the data set as a transition from a state to another. The small change is defined so that it guarantees to maintain the desired properties in the data set, $r(D) = r(D_i)$, but makes

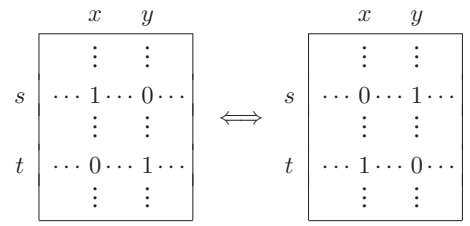


Figure 3.1. A swap in a 0–1 matrix.

it otherwise completely random. When random small changes are repeatedly made to the data set for the Markov-chain to have converged, the remaining data set is a random sample from Π_0, solving Problem 4.

In the binary matrix example, as the swap maintains the column and row margins, each state reachable from another state is a valid data set from the space of all data sets. However, the degree of each state might not be the same for all states, as the number of possible swaps might not be the same for each data set. This leads to non-uniform Π_0. To solve the problem, swaps are chosen by randomly choosing two 1's in the matrix, which correspond to (s, x) and (t, y) in Figure 3.1. Then, if a swap would result in an invalid data set, *i.e.*, (s, y) or (t, x) are not 0, the swap is not performed. This corresponds to a self-loop in the Markov-chain, resulting in equal degrees for each state and a uniform Π_0.

It is shown that, using the swaps, all binary matrices of the same size and with the same column and row margins can be reached. If the chain is run long enough for it to converge, the resulting state, a data set, is a uniformly random sample from the space of data sets with the same size, and column and row margins as the original data set.

3.4 Metropolis-Hastings

The above solution to Problem 4 depends on finding an appropriate small change that maintains all the properties in $r(D)$, and otherwise mixes the data set. Swap is an example of a small change that maintains exactly the row and column margins, but otherwise makes the data random. However, the existence of a small change is context specific, and a small change that maintains all $r(D)$ most likely does not exist for all but few cases. Furthermore, if several different sets of properties, different functions r, are used in several statistical significance tests, finding a small change to each may be prohibitively difficult, if not impossible. In the ex-

ample, if r also contained a requirement that frequencies of some itemsets should be maintained, the swap would not be appropriate alone. Therefore, another problem has been proposed that alleviates the requirement of maintaining the properties exactly.

Problem 5. Given a data set D, a property function $r(D)$, and a scaling constant ω, generate a data set D_i chosen with a probability

$$\rho(D_i) \propto \exp(-\omega||r(D) - r(D_i)||).$$

The scaling constant ω in the exponent controls how accurately the properties are maintained. The sampled data sets only approximately have the same properties as the original data set D_i. In the example, $r(D)$ could be the vector of frequencies of a given set of itemsets, thus introducing correlation constraints to the columns.

Problem 5 can be solved with the Metropolis-Hastings algorithm [5, 43, 66], that defines transition probabilities for the small changes when randomizing a data set. A small change still needs to be defined, but it does not have to maintain all the properties in $r(D)$. However, it should not maintain any other properties that are not included in $r(D)$, as then the randomization method would not solve Problem 5. The Metropolis-Hastings algorithm allows for a Markov-chain to converge to a chosen non-uniform distribution over all the states. This solution has been used in Publications 2 and 3 as well as in the literature [77].

The Markov-chain randomization process is altered in the Metropolis-Hastings algorithm by defining the probability

$$Pr(D_i'|D_i^j) = \min\left\{\frac{\rho(D_i')Q(D_i^j, D_i')}{\rho(D_i^j)Q(D_i', D_i^j)}, 1\right\}$$

for accepting the small change that would be made to the current data set D_i^j to arrive to the data set D_i', where Q is a proposal density discussed later. If the random test with probability $Pr(D_i'|D_i^j)$ succeeds, $D_i^{j+1} = D_i'$. If it fails, the small change is not made $D_i^{j+1} = D_i^j$, but the iteration is continued, randomizing a new small change.

If the proposal density Q is symmetric, in that for all D' and D^j it holds that $Q(D', D^j) = Q(D^j, D')$, it vanishes from the probability $Pr(D_i'|D_i^j)$. This is the case if the small change is reversible, in that for any legal small change from D_i^j to D_i^{j+1}, a small change exists that transforms D_i^{j+1} back to D_i^j, $D_i^{j+2} = D_i^j$, and their probabilities are equal. The transition probability $Pr(D_i'|D_i^j)$ only depends on the change in error

$$Pr(D_i'|D_i^j) = \min\left\{\exp\left(\omega(||r(D) - r(D_i^j)|| - ||r(D) - r(D_i')||)\right), 1\right\}.$$

It is thus beneficial to define a small change that is reversible and has a symmetric proposal density. The swap in the example meets these criteria.

However, in some contexts, the distribution of D_i may not strictly follow $\rho(D_i)$. Let $Pr_0(r(D_i))$ be the distribution of the properties when the data set is randomized with the small change without the constraints, *i.e.,* when $\omega = 0$. If $Pr_0(r(D_i))$ is uniform, D_i will follow $\rho(D_i)$ with this procedure. If, however, $Pr_0(r(D_i))$ is not uniform, the distribution for D_i will have the form $\rho(D_i) \propto Pr_0(r(D_i)) \exp(-\omega||r(D) - r(D_i)||)$. The effect of $Pr_0(r(D_i))$ can be reduced by adaptively shrinking ω in the Metropolis-Hastings process, or to define $Pr_0(r(D_i))$ and calculate the transition probabilities such that the distribution is canceled out.

3.5 Maintaining itemset frequencies (Publication 2)

Publication 2 considered randomizing binary matrices while maintaining the frequencies of a family of given itemsets, $\mathcal{F}$. The frequency constraints can be encoded as properties to preserve, and therefore, the following problem was considered.

Problem 6. Given a 0–1 matrix D and a family of itemsets $\mathcal{F}$, generate a matrix D_i chosen independently and uniformly from the set of 0–1 matrices having the same row and column margins as well as the same frequencies for the itemsets in $\mathcal{F}$ as the dataset D.

This is a special case of Problem 4, where the property function $r(D)$ outputs the frequencies of the itemsets in $\mathcal{F}$ for the matrix D. The size and margins are maintained by using the swap as a small change.

However, it was proven that the solution to this problem is NP-hard by providing a reduction from Hamiltonian cycle to Problem 6. Therefore, solving the problem is infeasible in general as randomization involves drawing several samples of data sets to obtain reasonable accuracy for the empirical p-values. The strict constraints on itemset frequencies were proposed to be relaxed and, instead, the frequencies are maintained approximately. The relaxed problem is a special case of Problem 5 with the same definition for $r(D)$, and again the swap is used as the small change in randomization.

3.6 Maintaining cluster structure (Publication 2)

Publication 2 also considered randomizing binary data sets while maintaining the cluster structure on rows. A clustering of a binary matrix is a grouping of rows to a specific number of groups, where rows within a group are similar and rows in different groups are dissimilar. The randomization procedure proposed uses the same swap as used before. However, the admissible swaps are restricted to within a cluster: only swaps that involve rows in a cluster are made. No swap is performed that would exchange values between clusters. The randomization of a swap was slightly modified by first randomizing a cluster in which a swap is performed, and then performing a random swap in that cluster. This procedure maintains the within-cluster margins, and therefore, maintains the overall cluster structure in that sense.

Maintaining the cluster structure has been studied recently by Vuokko and Kaski [100].

3.7 Randomizing graphs (Publication 3)

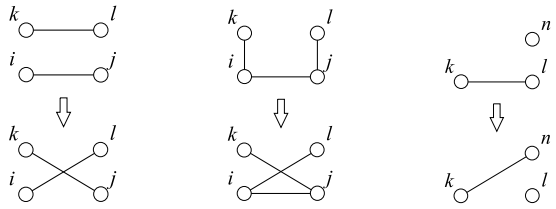
Publication 3 considered the following problem of sampling from property models, which is a special case of Problem 5.

Problem 7. Given an unweighted undirected graph G with n nodes and m edges and a property function r, generate a graph G_i chosen independently and uniformly from the set of graphs with n nodes and m edges and with the same degree distribution as G, which have approximately the same properties r.

Three different small changes were proposed that maintain the number of nodes and edges, and the degree distribution of the nodes. Using any of the small changes, Problem 7 can be solved similarly as Problem 5.

Figure 3.2 depicts the three different small changes, of which XSWAP has been previously used [64, 89]. Their corresponding interpretations are as follows.

XSWAP The small change freely exchanges edges between nodes. It maintains the node degrees, the number of edges each node has, but otherwise changes the graph completely. Given that these small changes are randomized uniformly, the converged Markov-chain results in a sample from the space of graphs with the same number of



(a) XSwap **(b)** LOCALSWAP **(c)** FLIP

Figure 3.2. Different swaps for randomization. The FLIP in c) is further conditioned with $\delta(l) - \delta(n) = 1$, where δ returns the degree of the node.

nodes and edges as the original matrix, as well as the same individual node degrees.

LOCALSWAP The small change is very similar to XSWAP. The difference is that changes are allowed only locally, and therefore, no edges are exchanged between unconnected components.

FLIP When individual node degrees are not important, then FLIP can be used. It maintains the overall node degree distribution.

Ying *et al.* [111] proposed and solved a very similar problem, with the difference that the graph statistics are allowed to change freely within strict user-defined bounds. They used XSWAP in the solution. Problem 7 requires that the expected value of the graph statistic in random graphs has to be the same as in the original graph. In the work of Ying *et al.*, the expected value need not be equal to that of the original graph. The expected value can even be close to either of the bounds, which constraint the admissible values for the graph statistic. Therefore, the solutions to the different problems create slightly different null distributions of graphs[1].

3.8 Convergence

An intrinsic question of Markov-chains is their convergence. A Markov-chain is said to have converged, or mixed, when it loses all information about the initial state and thus each state is equally likely to be the initial state [52]. Only when a Markov-chain has converged does the final state represent a sample from the steady state distribution, which is the null distribution Π_0. Before that, the sample is biased towards the initial state.

Knowing when a Markov-chain has converged is a difficult task, unsolved in general [21, 52]. The approaches are divided into two main

[1]Both studies were published in the same conference.

categories: estimating how many iterations are required for the chain to converge, and iterating and diagnosing when has the chain converged. The latter is simpler and it is used in present literature [32, 77].

One common way of approximating the needed number of transitions is to measure when the distance to the original data set has converged [32, 77]. This can be achieved by trying different amount of transitions, randomizing with each amount separately, and calculating the distance between the random matrix and the original one. When an increase in the number of transitions has sufficiently small effect in the distance, that number of transitions is chosen.

3.9 Drawing multiple samples

If the Markov-chain is always started from the original data set when producing a new sample, all the samples are dependent on the initial state [13]. This problem is solved by using Backward-Forward sampling. The procedure is to first run the Markov-chain backwards for the specified number of steps to arrive at D_0, and then start each new Markov-chain for each new sample from D_0. The samples drawn this way are still not independent, but are exchangeable under the null hypothesis, which means that the original data set and the random data sets are in theory sampled from the same distribution.

If the small change is reversible, then running the Markov-chain backwards equals running it forwards. With this, the procedure is essentially to first sample a single data set, and then use that sample as the initial state from which to produce all the actual samples.

3.10 Experiments on graph randomization (Publication 3)

Publication 3 experimented with the solution to Problem 7 in the contexts of graph clustering and frequent subgraph mining. In both, the statistical significance of the found results was assessed with varying constraints in randomization.

A multitude of graph statistics have been proposed in the literature [51, 71]. Two of these were chosen for consideration and it was shown how they can be used as maintained graph properties in randomization.

Average clustering coefficient Clustering coefficient for a node is de-

fined as the proportion of edges between the node's neighbors of all possible such edges,

$$CC(v) = \frac{|\{(u,w)|u,w \in \Gamma(v) \wedge (u,w) \in E(G)\}|}{|\Gamma(v)|(|\Gamma(v)|-1)/2},$$

where $\Gamma(v)$ is the set of neighbors of the node v, and E is the collection of edges, pairs of nodes, of graph G. The average clustering coefficient is then defined as

$$ACC(G) = \frac{1}{|V(G)|} \sum_{v \in V(G)} CC(v),$$

where $V(G)$ is the set of nodes in graph G.

Characteristic path length The path length, denoted by $d_G(u,v)$, between two nodes u and v is the shortest graph theoretic distance between the nodes, or equally, the minimum number of edges needed to travel to get from u to w using edges in $E(G)$. The characteristic path length of a graph is then defined as the average of all the pairwise path lengths

$$CPL(G) = \frac{1}{|V(G)|^2} \sum_{u,v \in V(G)} d_G(u,v).$$

The average clustering coefficient of a graph expresses its clusteredness. Maintaining the average clustering coefficient may be reasonable in, for example, social networks which are known to have a clustered structure [102]. The characteristic path length is also a reasonable statistic in social networks, where distances between people tend to be short [67]. The uses do not limit to social networks, but many other applications can be considered.

In the experiments with graph clustering, the statistical significance of the clusterings of the original graph to different number of clusters was assessed. The significance was tested with different null distributions Π_0, varying the small change to be used and the graph statistic to preserve. Clustering error was used as the test statistic, since it naturally expresses the goodness of the found clustering. The assessment was carried out for different number of clusters, with the aim of discovering the relation between empirical p-values and number of clusters for different Π_0.

It was discovered that, with a graph of 34 nodes and 78 edges, increasing the number of clusters to above some threshold increased the empirical p-values from near zero to around one. Essentially, the correct number of clusters in the graph was discovered. When this number is exceeded, the

clustering algorithm has to cluster the existing clusters, which results in a high clustering error. As the random graphs do not have such clustering structure, increasing the number of clusters will have only a small increasing effect in the clustering error.

The graph clustering results with all of the used graphs and null distributions Π_0 showed that using different small changes and graph statistics to preserve affected the empirical p-values. In essence, less constraints produced smaller empirical p-values.

In frequent subgraph mining, a graph transaction data set was used as input data and it was mined for frequent subgraphs. Each frequent subgraph was calculated an empirical p-value, where the frequency of the respective subgraph was used as a test statistic. The numbers of frequent subgraphs with less than 0.05 empirical p-value were compared with different null distributions Π_0, where the small change, graph statistics to preserve and minimum frequency threshold was varied. FLIP produced smaller p-values when compared to XSWAP and LOCALSWAP. However, constraining with average clustering coefficient did not have a big effect in the results.

3.11 Summary

Resampling by randomization is a statistical significance testing framework that is based on drawing random samples of data sets and testing the significance of the results against the random data sets. Randomization is a general concept of creating random data sets from a null distribution of data sets.

Models for random data can be divided into two groups: exact and property models. In exact models, the null distribution of random data is explicitly defined. In a property model, the null distribution is defined as a uniform distribution over all data sets that share some predefined properties with the original data sets.

Markov-chain Monte Carlo can be used to draw samples of data sets from various null distributions that do not admit a simple constructive randomization method. MCMC is based on repeatedly carrying out random small changes to the original data set. When enough changes have been made for the Markov-chain to have converged, the final data set is a sample from the null distribution. Metropolis-Hastings algorithm can be used to alter MCMC by defining a probability to perform a random small

change. The probabilities guarantee that a random data set is a sample from a desired null distribution.

Publication 2 introduced the problem of drawing samples of binary matrices that have the same size, row and column margins, and frequencies for a given family of itemsets as the original binary matrix. The problem was proven to be NP-hard, and is, therefore, relaxed such that the frequencies are only maintained approximately. The solution is based on the Metropolis-Hastings algorithm. The publication also proposed to maintain the clustering of rows of a binary matrix during randomization, and proposed a modified MCMC method to draw samples of such binary matrices.

Publication 3 proposed a graph randomization framework for unweighted undirected graphs. All the random graphs share the number of nodes and edges, and the degree distribution of nodes, with the original graph. Such random graphs could be drawn by using one of the three small changes defined in Publication 3. It was also shown how the random graphs can be further constrained to have approximately same values for user-defined graph statistics. The solution is based on the Metropolis-Hastings algorithm to maintain the statistics. The proposed graph randomization framework was used in experiments involving graph clustering and frequent subgraph mining.

4. Iterative data mining application

Iterative data mining is the process of repeatedly carrying out statistical significance tests on data mining results, where at each step, the test is constrained with a part of the result previously found statistically significant. As that, it is a good application for demonstrating the use of multiple hypothesis testing in data mining. This chapter is based on Publication 2.

4.1 Iterative data mining (Publication 2)

Often in real data mining situations, the user applies several algorithms on the data set at hand to obtain as much information of it as possible. The different results express different aspects of the data, and may be based on completely different theory. For instance, a user might first cluster the data, then reduce its dimensionality with PCA to plot it, and finally find all the frequent itemsets in it. While the combination of different algorithms provide meaningful viewpoints to the data, the user may not know if the results of different algorithms express the same phenomenon in the data.

Data mining algorithms also often produce a large collection of patterns, from which the user needs to decipher which of them represent true phenomena and which are mere consequence of these other patterns. For example in frequent itemset mining, a large pattern may be the effect of a real phenomenon, while its subset becomes frequent only because its superset is.

Iterative data mining process tackles this issue by assuming part of the result to be known, and then finding out what is left in the data that is not explained by what is known. The process starts from scratch, with no result assumed, and iteratively builds up knowledge about the data, at each step testing the results that are not explained by the already known

knowledge.

The proposed solution is based on statistical significance testing. It is used to assess the unexpectedness of the results at each iteration, while constraining the test with the already known knowledge. Any pattern that is declared statistically significant is not explained by the constraints of the test. Conversely, if a pattern is not statistically significant, the knowledge contained in the constraints of the significance test explain the pattern. Therefore, at each iteration of the approach, the statistical significance of the results is assessed and the significant results are examined. When some part of the significant results are considered understood and chosen to be fixed, that part is added as constraints in the significance test for the succeeding iterations. This idea has recently been extended by Mampaey *et al.* [58].

As many hypotheses are considered at each step of the iterative data mining process, the problem of multiple hypothesis testing needs to be taken into account.

4.1.1 Iterative data mining with binary matrices

Iterative data mining can be applied, for example, in frequent itemset mining and clustering in binary matrices. The underlying idea is that if an itemset is statistically significant, it is not explained by the itemsets in the constraints for randomization. Using this, the procedure is based on incrementally adding itemsets as constraints to randomization and assessing the statistical significance of the other itemsets.

First the input binary matrix is mined for frequent itemsets and their statistical significance is assessed by randomizing the binary matrix while maintaining the column and row margins. The randomization method is presented in Section 3.3. The statistically significant itemsets are maintained for further study, and the rest are scrapped.

Then the iterative procedure starts by selecting the itemset with the smallest adjusted p-value and including that in the constraints in randomization. Randomizing binary matrices while maintaining the column and row margins, as well as the frequency of a given set of itemsets is presented in Section 3.5. Randomization is used to assess the statistical significance of the remaining itemsets while maintaining the frequencies of the itemsets in the constraints. Then, all not statistically significant itemsets are scrapped and the procedure is iterated. This is continued, for example, until a certain number of itemsets is selected, or no statisti-

cally significant itemsets remain.

The itemset with the smallest adjusted p-value is considered the most interesting in this example, and when the user has examined it and understood its implications, it is considered as understood knowledge. Therefore, the itemset with the smallest p-value is added to the constraints at each iteration. When the iterative procedure has ended, the itemsets in the constraints can be considered to be the most surprising itemsets in the data, and which express different phenomena. Experiments with this procedure were presented, which showed that the amount of statistically significant itemsets almost always decreases with increasing number of constraints in randomization.

Another example scenario is to compare the results of frequent itemsets with clustering results. The intuition is that a good clustering result may explain the frequencies of itemsets. To this end, the statistical significance of itemsets were assessed when maintaining the row and column margins, as well as the cluster structure of rows. Such randomization was presented in Section 3.6. Another significance test was also carried out, where only the row and column margins were maintained. The number of significant itemsets in the different cases showed that maintaining the cluster structure in randomization is very restrictive. Many of the itemsets found significant while maintaining only column and row margins were not found significant when also maintaining the cluster structure.

The ability of itemsets to explain the clustering result was also tested. Itemsets, that had the largest increase in their p-values between the cases of not maintaining cluster structure and maintaining it, were selected as constraints. These itemsets were thought to potentially explain the clustering structure, because their p-values are most affected. In the experiments, it turned out that maintaining these itemsets during randomization did not explain the clustering structure, and therefore, the clustering structure was found statistically significant when having the itemsets as constraints in randomization.

4.2 Experiments (Applying Publication 1 in Publication 2)

Mining frequent itemsets iteratively provides a setting that can be easily experimented with. The iterative data mining scenario of Publication 2, where the most significant itemsets were incrementally added to the constraints in randomization, is used in this section. The purpose of this

section is to display that the method from Publication 1 can be used to carry out iterative data mining of Publication 2.

In contrast to Publication 2, here the p-values are calculated with the method defined in Equation (2.4) to directly obtain FWER-adjusted p-values. Publication 2 controlled FDR using the method by Webb [104] in combination with the method by Benjamini-Yekutieli [12]. The test statistics are also different. The contents of this section is new and unpublished.

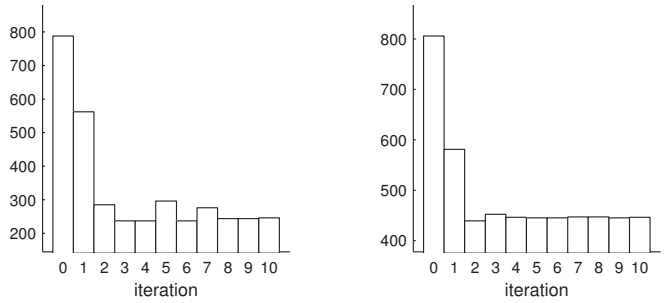
The used data sets are *Paleo* and *Courses*. *Paleo* has 124 rows and 139 columns, and 11.5% density. It contains paleontological discoveries in Europe, where each column represents a species and each row represents a cite of excavation. A 1 in a cell signifies that the species has been found in the excavation site. *Courses* has 2404 rows and 41 columns, and 32% density. The data consists of information about students taking courses, each row corresponding to a student and a column to a course. A 1 in a cell signifies that the student has taken the corresponding course.

At each step of the iteration, the convergence of the Markov chain is assessed by using the method presented in Section 3.8. The scaling parameter ω was set to 4, which causes the probability of accepting a swap that would increase the error in constrains by one to be close to 2%, when L_1 is used to measure the difference in the properties $r(D)$. For each null hypothesis, 1000 samples were drawn and the data set was mined for frequent itemsets. For *Paleo* and for *Courses*, the minimum support thresholds were set to 7 and 400, respectively, as was done in [32]. The test statistic of an itemset was an approximation of the lift,

$$f(x) = -\frac{\mathrm{fr}(x)}{\prod_{A \in x} \mathrm{fr}(A)},$$

where $\mathrm{fr}(x) \in [0, 1]$ is the fractional frequency of the itemset x, $\mathrm{fr}(x) = \frac{\mathrm{freq(x)}}{\mathrm{N}}$. It removes the effect of monotonicity of frequency and focuses on itemsets that have an exceptional frequency with respect to their individual items. The set of test statistic values calculated for the frequent itemsets found from a random data set are used in the method in Equation (2.4). A total of 10 iterations were run, and at each iteration, the number of statistically significant itemsets were calculated. Only the itemsets of size 2 or 3 were considered, and any other itemset was discarded.

Figure 4.1 depicts for both of the data sets the number of itemsets found significant with FWER≤ 0.05. The number of statistically significant itemsets rapidly decreased for both data sets when the first few new con-



(a) *Paleo* **(b)** *Courses*

Figure 4.1. Iterative data mining results. Number of itemsets found significant with FWER$\leq$5% in different iterations. The iteration number on the horizontal axis signifies the number of itemsets in constraints.

straints were added. This was an expected result. However, after two added constraints, the number significant itemsets did not change notably. Clearly, while the itemsets restricted the null distribution of data sets, it did not affect the fraction of high lift itemsets found in random data.

These results differ from the results in Publication 2, where the number of significant itemsets without constraints was around 50% higher and the general trend in the numbers is decreasing. The differences are most likely caused by the different MHT errors to control, since FDR was controlled in Publication 2, but also the different test statistics and p-value calculations contribute in the difference. This also makes the detailed comparison of the results unmeaningful.

As a conclusion, the presented multiple hypothesis testing method can also be used in iterative data mining.

4.3 Summary

Iterative data mining is a concept of applied data mining, where the goal is to find a set of results that represent approximately disjoint phenomena in the original data. In other words, the results are not explained by one other, but express some structure of the data that is not expressed by other results. One example scenario is the mining of statistically significant frequent itemsets iteratively. The idea in that is to assess the statistical significance of the frequent itemsets with varying set of constraints for the null distribution. The set of constraints is extended at

each iteration, building the amount of information assumed to be known of the data and discovering what else there is in the data.

Publication 2 presented experiments with such a procedure, and showed that the number of significant itemsets generally decreases as the number of constraints increases. Another scenario in Publication 2 involved testing the relation of clustering and frequent itemset results. The experiments in the publication showed that maintaining the cluster structure of a matrix may have a strong effect in the statistical significance of the itemsets. Conversely, maintaining the frequency of many itemsets did not have as strong effect in the clustering results.

Mining frequent itemsets was experimented with new and unpublished tests using the method presented in Chapter 2. The experiments showed that the method can be used in such a setting and it produces meaningful results.

5. Biclustering application

Biclustering is a general application of data mining, where groups of rows and columns are searched that express some local structure in the input data matrix. Biclustering is introduced and the contributions of Publications 4 and 5 are discussed here, together with some previously unpublished experiments.

5.1 Introduction

Several approaches exist to discover the structure between genes, among which are the tasks to infer the gene regulatory network [23], and to discover protein functions [90], to name only two. To this end, *biclustering* [19] has been proposed, where groups of genes and experiments are sought such that the genes behave similarly across the respective experiments. The underlying idea is that the data set contains substructures of genes and experiments, while the same genes for other experiments may or may not behave similarly. However, the biclustering can also be used in many other settings, where an interesting local structure in a data matrix can be expressed by a subset of rows and a subset of columns [16].

Perhaps the simplest definition of a bicluster is that the submatrix, defined by a group of rows and a group of columns of a data matrix, contains as similar values as possible. Such a bicluster is called constant. All biclusters can be divided into four major classes:

1. Constant values.

2. Constant values on rows or columns.

3. Coherent values.

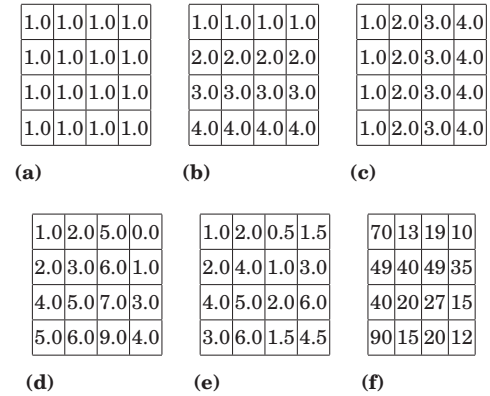


Figure 5.1. Examples of different types of biclusters. (a) Constant values, (b) constant rows, (c) constant columns, (d) additively coherent values, (e) multiplicatively coherent values, and (f) coherent evolution on the rows. The figure and examples follow the work of Madeira and Oliveira [57].

4. Coherent evolutions.

The first three directly consider the numeric values in the data matrix, and try to identify the respective structure. The second class requires that the values on each row (or column) are constant, but values between rows (or columns) may be different. The third class requires that all of the values are similar under either additive or multiplicative model. The values within such a bicluster are explained by a row and a column vector, which are either added or multiplied element-wise to create the bicluster.

The fourth class does not consider the numeric values as is, but views them as symbols. An example of this is to find biclusters, where the order of elements on rows are the same for all rows, as in Figure 5.1(f). Other examples of bicluster structures are shown in Figure 5.1.

In addition to the definition of a bicluster, one also needs to know how biclusters can be located in the matrix with respect to each other. One definition is to partition the rows to groups and the columns to groups, effectively creating biclusters between all pairs of row and column groups. Such orientation is called a checkerboard. Other definitions include, for example, exclusive rows, non-overlapping nonexclusive biclusters, and arbitrarily positioned, etc. Figure 5.2 lists all of the possible positions and structures for biclusters.

A comprehensive review of biclustering is available in [57]. Since then, new biclustering methods have been proposed [44, 87, 92, 108, 113].

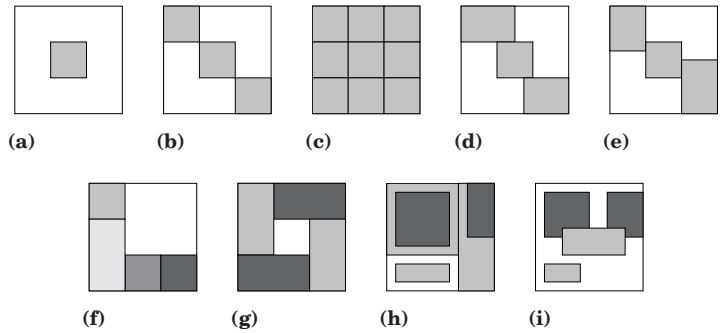


Figure 5.2. Different structures and positions of biclusters. (a) Single bicluster, (b) exclusive rows and columns, (c) checkerboard, (d) exclusive rows, (e) exclusive columns, (f) non-overlapping with tree structure, (g) non-overlapping nonexclusive, (h) overlapping with hierarchical structure, and (i) arbitrary. The figure and definitions follow the work of Madeira and Oliveira [57].

5.2 Finding constant biclusters approximately (Publication 4)

Among the first approaches for biclustering was to search for constant biclusters [42]. The algorithm works top-down, at each iteration splitting the data appropriately to form new biclusters, until a user-given number of biclusters have been found. Later approaches for biclustering consider more elaborate models, but contain the constant biclusters as a special case.

Publication 4 proposed biclustering the input matrix to constant biclusters by clustering the rows and columns of the input matrix independently. The benefit of this is that the method is very simple and one can use the existing clustering methods, which have been studied extensively [6, 37, 79]. An approximation ratio for such an algorithm was proven for the cases of binary and real valued input matrices.

Clustering rows and columns independently produces groupings for the rows and the columns. This produces a checkerboard positioning of the biclusters, which can be seen in Figure 5.2(c). Furthermore, constant valued biclusters were sought, an example of which can be seen in Figure 5.1(a).

The optimization criteria for such biclustering is defined as follows. Let D be the data matrix to bicluster, and K_r and K_c the number of row and column clusters, respectively. The submatrix induced by the set of rows $R \subseteq [N]$ and the set of columns $C \subseteq [M]$ is denoted by $D(R, C)$. For a checkerboard type biclustering, the groups of rows R_i form the set $\mathcal{R} = \{R_i\}_{i=1}^{K_r}$, and the groups of columns C_i form the set $\mathcal{C} = \{C_i\}_{i=1}^{K_c}$. The total

Algorithm 1 Finding constant biclusters in checkerboard structure.

Input: data set D, number of clusters K_r and K_c, clustering algorithm $A_{cluster}$

Output: $\mathcal{R} = \{R_i\}_{i=1}^{K_r}$, $\mathcal{C} = \{C_i\}_{i=1}^{K_c}$

$\mathcal{R} = A_{cluster}(D, K_r)$

$\mathcal{C} = A_{cluster}(D^\top, K_c)$

error of such a biclustering is

$$L_D = \sum_{R \in \mathcal{R}} \sum_{C \in \mathcal{C}} \mathcal{V}(D(R, C)), \tag{5.1}$$

where $\mathcal{V}()$ is a measure containing the norm-dependent part. For L_1-norm, $\mathcal{V}(Y) = \sum_{y \in Y} |y - \text{median}(Y)|$, and for L_2-norm, $\mathcal{V}(Y) = \sum_{y \in Y} (y - \text{mean}(Y))^2$. The optimal solution to Equation (5.1) is denoted by L_D^*. The literature knows several algorithms to cluster rows and columns simultaneously to minimize L_D [19, 20, 42].

Equation (5.1) is optimized by finding $\mathcal{R}$ and $\mathcal{C}$ independently. Algorithm 1 illustrates the biclustering method in pseudocode.

The main contribution of Publication 4 was to prove the following approximation ratios for Algorithm 1.

Theorem 8. *There exists an approximation ratio of α such that $L_D \leq \alpha L_D^*$, where $\alpha = 1 + \sqrt{2} \approx 2.41$ for L_1-norm and $D \in \{0, 1\}^{N \times M}$, and $\alpha = 2$ for L_2-norm and $D \in \mathbb{R}^{N \times M}$.*

The approximation ratios hold if the algorithm $A_{cluster}(D, K)$ returns the optimal clustering of the rows of D to K clusters, *i.e.*, finds the optimal minimum to Equation (5.1) with $\mathcal{C} = \{[M]\}$ and $|\mathcal{R}| = K$. If $A_{cluster}$ is approximate, the approximation ratios of Theorem 8 have to be multiplied by the approximation ratio of the used clustering algorithm. The proof of the theorem is presented in the publication. A similar study was later presented by Anagnostopoulos *et al.* [4].

5.2.1 Experiments (Applying Publication 1 in Publication 4)

The biclustering method locates a number of biclusters in the given data matrix. If the statistical significance of the found biclusters is assessed, multiple hypothesis testing has to be considered. The number of biclusters is constant, but the biclusters are most likely different for different data sets. Therefore, the context of biclustering is suitable for the MHT

method presented in Equation (2.4) to be used. This section contains new and unpublished experiments with the biclustering method and the MHT method presented earlier.

The data sets are the same as was used in Section 4.2, *Paleo* and *Courses*. In addition, an artificial data set of size 20×20 is constructed by first setting to ones the upper right and lower left parts of the matrix, *i.e.*, the submatrices $(\{1,\ldots,10\},\{1,\ldots,10\})$ and $(\{11,\ldots,20\},\{11,\ldots,20\})$, and then introducing noise by randomly selecting 10% of the cells and flipping their values. The artificial data therefore contains four clear biclusters with equal size. The data set is named *Artificial*.

Different number of clusters K_r and K_c were used, and k-means was used to cluster the rows and columns independently. Each bicluster was considered as a single pattern, and the statistical significance of all the biclusters were assessed. The test statistic of bicluster was $f((R,C),D) = |R||C| - \mathcal{V}(D(R,C))$, where $\mathcal{V}()$ was defined with the L_1-norm. The test statistic calculates the number of values in the majority group. This is used because a large bicluster with a small error is chosen to be most interesting.

Four different null distributions for data sets are used. The null distributions always maintain the size and type of the data set, but may additionally have the same row or column margins as the original data. The null distributions are named SHUFFLE, ROWSHUFFLE, COLSHUFFLE and SWAP. The first shuffles all values within the matrix, and therefore, does not maintain any margins. The middle two shuffle the values within each row or column separately, respectively. The randomizations maintain the row or column margins. The last null distribution, SWAP, was introduced earlier in Section 3.3.

For each combination of original data set, null distribution of data sets and number of row and column clusters, 100 random data sets are produced from the null distribution. Each of the random data sets is biclustered and the test statistic values for each bicluster are calculated. The obtained test statistics are used in Equation (2.4).

Table 5.1 illustrates the number of biclusters found statistically significant with FWER≤ 0.05. There were four statistically significant biclusters in *Artificial* for all randomizations. This is expected, since the data was constructed to have four biclusters. Notice that the number of biclusters searched is more than the number biclusters found significant. The two biclusters in the original data that were not statistically significant were

	Artificial	*Paleo*	*Courses*
	2×3	3×3	3×3
SHUFFLE	4	0	0
ROWSHUFFLE	4	0	0
COLSHUFFLE	4	0	0
SWAP	4	0	0

Table 5.1. Number of statistically significant clusters with FWER≤ 0.05 for different data sets and randomization methods. The numbers under the data set names signify the number of clusters $K_r \times K_c$.

very small and contained mostly noise.

The results for *Paleo* and *Courses* are not as interesting. None of the biclusters were statistically significant in any of the randomizations with any number of clusters. Equally good biclusters could be found in random data than could be found in the original data sets. Apparently the data sets do not have clear biclustering structures that could be defined by partitioning rows and columns.

5.3 Finding coherent biclusters with minimum description length (Publication 5)

The previously used bicluster definition is adequate if constant biclusters are sought and it is reasonable to obtain a global clustering. However, the data may sometimes contain only local structures and not a global one. Then it is better to find arbitrarily positioned biclusters, that only express structure in a local environment. Furthermore, as the bicluster structure expresses the relation between the rows and columns of the bicluster, more elaborate structures may be sought than simple constant values within the bicluster. Publication 5 considered finding arbitrarily positioned biclusters, seen in Figure 5.2(i), that describe the local linear structure, seen in Figure 5.1(e). The biclusters are searched using the Minimum Description Length (MDL) principle [35, 83, 84, 85].

5.3.1 Minimum description length

MDL is used to find the best model among a family of models to describe the data. It is based on Kolmogorov complexity: complex data requires a long description and simple data requires only a short description. For example, consider the following binary sequences:

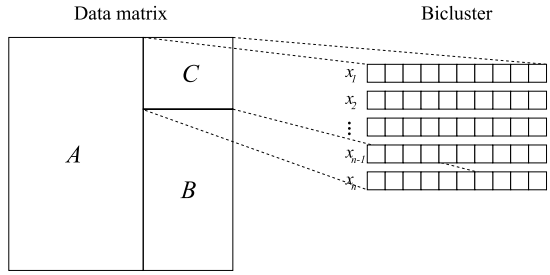


Figure 5.3. Complete model of a single bicluster for the data matrix. Areas A and B are modeled as background, and separately from the bicluster C. The bicluster is considered to comprise of vectors $\{x_i\}_{i=1}^{n}$. The rows and columns are sorted for clarity.

11011101110111010111011101

10101110010101010111011011

The first can be described with the sentence "1101 6 times", which requires 14 characters. As for the second, there is no obvious simple description of it except to write the sequence as is, which requires 24 characters. The first sequence has clear structure and can be described easily, while the latter has no obvious structure and is lengthy to describe.

However, only the length of the description is of interest, and not the description itself. The idea is to calculate the number of bits required to describe the data with a selected model, plus the number of bits required to describe the model, which is the *description length* of the data using the model. If the description length using a certain model is small, then that model describes the structure in the data well. Conversely, if the description length is large, the model is a poor model for the structure of the data. The model that minimizes the description length is considered the best model.

5.3.2 Bicluster model

The bicluster model is used to calculate how well a single bicluster describes the structure of a local neighborhood in the data matrix. The complete model for a single bicluster is defined in two parts: a model to describe the data in the bicluster and a model to describe the rest of the data. An overview of the complete model is illustrated in Figure 5.3, where area C is the data within the bicluster and areas A and B cover the rest of the data. Each part of the data matrix is modeled such that the rows are independent samples from some distribution.

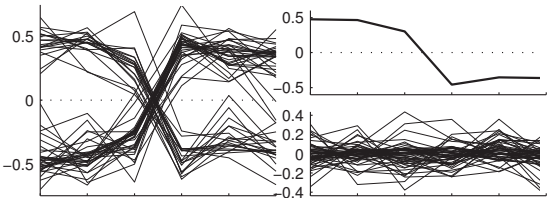


Figure 5.4. Example bicluster structure. Left, each line represents the values of a single row over the columns of the bicluster. Top-right, strongest linear component separated from the rows. Bottom-right, what is left after the strongest linear component is removed from the rows, *i.e.*, the residuals.

The complete model is such that a larger bicluster fits the data better, and, therefore, the model for a larger bicluster is more complex. However, MDL takes this into account and penalizes complexity, since a more complex model has more model parameters that also need to be encoded. In the end, the description length of the complete model is used to calculate a goodness of fit for a single bicluster, that requires no user-specified parameter values.

The bicluster model for the rows in the area C in Figure 5.3 is as follows. Each row vector of the bicluster is normalized to unit length and modeled with the strongest linear component among the unit row vectors, and with a residual vector, that captures the deviance of a row vector from the strongest linear component. Let the normalized rows of a bicluster be a set of vectors $\{\overline{x}_i\}_{i=1}^n$. The strongest linear component of a set of vectors is the eigenvector v_1 of the covariance matrix $\Sigma = \sum_{i=1}^n \overline{x}_i \overline{x}_i^\top$ that has the largest eigenvalue λ_1. Using this, each vector is modeled with $\overline{x}_i = \alpha_i v_1 + r_i$, where α_i is a multiplier used to scale v_1, and r_i is the residual. Figure 5.4 illustrates this model and its different parts.

The rest of the data was modeled such that the matrix values do not influence the search for biclusters, but only the goodness of the bicluster. Likelihood was defined for these models and, in accordance with the MDL methodology, the likelihood was normalized over all data matrices of the same size. The normalization could not be carried out analytically, and thus, it was approximated with sampling. The exact formula for the description length is not presented here, since it is complex and is derived in Publication 5.

5.3.3 Biclustering algorithm

The actual biclustering algorithm, called BiMDL, first locates a set of bicluster seeds, and then, for each seed, uses the description length to ex-

tend the seed by alternating between selecting rows and columns of the bicluster.

A single seed is a small bicluster that expresses a potentially interesting region in the data matrix. The seeds are generated by first discretizing the data matrix by setting to 1's all the cells that have an absolute value above a user-specified threshold. Then, all distinct triplets of columns are iterated, and, for each triplet, a seed is generated if sufficiently many rows have 1's on all of the columns of the triplet. The threshold for sufficiently many rows to generate a seed is defined by the user. The generated seeds represent regions of extremely deviant expression levels, and therefore, are thought to express interesting regions in the data matrix. Other seeding methods are possible; for example, the biclusters of another method can be used as seeds for BiMDL, or random 3 by 3 seeds could be used.

Each seed is extended by alternating between fixing the set of columns and selecting the set of rows, and fixing the set of rows and selecting the set of columns. To select rows, only the submatrix of the original matrix that is covered by the fixed set of columns is considered. The row vectors are normalized to unit length and sorted according to their concordance to the other row vectors. Then the three best rows are set as the bicluster rows and its description length is calculated. Rows are then added to the bicluster one by one according to the order and the description length is calculated for each set of rows. Finally, the set of rows that had the smallest description length is selected as the new set of bicluster rows. This procedure is the same when fixing the set of rows and selecting the set of columns. However, the data matrix is transposed so that columns are handled equally to rows. Alternating between selecting rows and columns is continued until neither rows nor columns change, or the algorithm has trapped into an infinite loop.

All the bicluster seeds are extended this way to produce a set of final biclusters that express some local structure in the input data matrix. However, the seed generation system can create more seeds than there are actual biclusters in the matrix, and hence, the extended biclusters may be overlapping or even exactly the same. Overlap between biclusters is removed by finding the pair of biclusters that have the largest Jaccard's index, and removing the bicluster with the larger description length. This procedure is continued until no bicluster pair has a larger Jaccard's index than a user-specified threshold. In the end, the user is presented with a set of biclusters that are guaranteed to have at most an allowed amount

of overlap.

5.3.4 Statistical significance using Publication 1

Experiments were carried out in Publication 5 that utilize the method in Publication 1. Testing the statistical significance of biclustering results has been largely ignored and is thus new to the context. Furthermore, in contrast to finding a constant number of biclusters, this time the number of found biclusters is not known before the biclustering is carried out. It was shown in the publication that the multiple hypothesis testing method could be used in this context. This resulted in 6 statistically significant biclusters out of 12 biclusters for BiMDL when bicluster seeds were found as described above. When seeds were generated randomly, 51 out of 58 biclusters were found statistically significant. The acceptable level of FWER was 0.05. As a conclusion, the MHT method was able to distinguish the statistically significant biclusters among the found biclusters, and thus, provided meaningful results.

5.4 Summary

Biclustering is a method for locating areas of local structure in a data matrix. A single bicluster is defined as a set of rows and columns of the data matrix, and the cells within the bicluster behave somehow similarly. Similarity can be defined in a variety of ways, for example, as constant values or coherent evolution on columns. Biclusters can also be located in different ways with respect to each other. For example, biclusters can form a checkerboard structure, or can be arbitrarily located in the data matrix.

Obtaining a biclustering involves considering simultaneously the rows and columns of the data matrix. However, a biclustering can also be obtained by separately clustering the rows and the columns of the data matrix. Publication 4 provided an approximation ratio for such a biclustering algorithm.

However, more elaborate algorithms are needed when the structure of a bicluster is complex or the biclusters are not mutually exclusive and do not have to cover the whole data matrix. Publication 5 proposed an algorithm to locate biclusters that describe local structures in the data matrix. The algorithm utilized the Minimum Description Length (MDL)

principle, with which the model for a bicluster could be defined parameter-free.

Assessing the statistical significance of all the found biclusters falls under the multiple hypothesis testing scenario. The method in Publication 1 can be used to assess the statistical significance of the biclusters found with the methods of Publications 4 and 5.

6. Discussion

This work introduced multiple hypothesis testing in data mining with specific applications to iterative data mining and biclustering. The applications were also introduced and discussed, and the presented MHT method was used in the contexts of these applications. The MHT method and applications are discussed next in general with possible future work.

6.1 Multiple hypothesis testing

Multiple hypothesis testing in data mining is an important question. If statistical significance tests are not controlled for multiple comparisons, the user may make false judgments about the validity of the results and commit vast amounts of time and money to seemingly interesting results. Only by careful consideration and treatment, can the data mining results be tested for statistical significance and be trusted to represent exceptional phenomena in the data.

A method to solve the problem was reviewed and experimented with. The method provides provable strong control over FWER in very general scenarios, which is its main benefit. Another favorable aspect is the very simple calculation of the p-values. This is important in randomization, where the computational cost is large due to the large number of samples required. Thus minimizing the complexity of any part of the process will reduce the overall cost. The experiments carried out on the method provide evidence of its utility and power.

The presented method adds slight complexity to the selection of a test statistic. As always in statistical significance testing, selecting an appropriate test statistic has to be done very carefully so that the statistical test is able to answer the correct question. However, when using the presented method, it should also be considered that the p-values are calculated by

comparing the test statistics of all the patterns. If, for example, one would like to use frequency as a test statistic in frequent itemset mining, the comparison may not be reasonable. The frequency of a large itemset is always at most the smallest frequency among all of its subsets. Therefore, larger itemsets tend to have smaller frequencies. If frequency is used as a test statistic, smaller itemsets are favored because the test statistics of different patterns are compared. However, this may not be a problem if a better score of the optimized function is always more interesting, which is common in data mining scenarios. Such is the case in the presented biclustering application, BiMDL. Furthermore, some measures to translate the test statistics to better comparable ones, as done in the presented publications, may be necessary. Specifically, the user may choose to use the marginal probabilities as test statistics, and by that, translate the distributions of the test statistic of all the patterns to the uniform distribution between $[0, 1]$.

A limitation of the presented method is that it controls the strict measure of FWER, while as other errors, such as FDR, may be more intuitive choices in some cases. It may be possible to provide similarly simple method to control FDR. One possibility could be to consider extending some of the existing work of resampling-based control of FDR [56, 109] to data mining scenarios and proving their validity.

Comparing test statistics between all patterns is also a good candidate for future work. It may not always be reasonable to compare all patterns to all other patterns, but to somehow restrict the comparison meaningfully. One approach could be to divide all the patterns to different groups, where all patterns within a group are known to have similar test statistic values in random data. In the frequent itemset mining scenario, one could consider grouping all itemsets of same size to one group and compare only the test statistics of equal sized itemsets. While maybe not the optimal solution, this approach could provide direction for future study.

6.2 Randomization

Randomization is a very general method to carry out statistical significance testing. It may be much easier to define a reasonable randomization process for the input data than to define the distribution for a test statistic under the null hypothesis. Furthermore, in many data mining scenarios, assumptions are much easier to make about the data generation

process than about some measures of the patterns produced by complex data mining methods. However, randomization has its drawbacks. Analytical distributions for test statistics often admit very fast calculations with arbitrary precision. Conversely, randomization methods require vast computations that limit the size of the problem that can be considered. Furthermore, to increase the precision of empirical p-values, the number of randomized data sets has to be increased. This in turn requires more computations, and therefore, takes more time to carry out the testing. When the number of random data sets increases to thousands, even to tens of thousands, the implementation has to be optimized for the randomization and the data mining algorithm. Even small improvements often accumulate to a significant amount of time saved in computations.

Another yet generally unsolved problem with randomization is the convergence of the Markov chain. A method was presented to approximately measure convergence. However, the determined number of steps for the Markov chain may still be far too little for the chain to converge. If the chain has not converged, the random data set is not a sample from the null distribution of data sets, but is likely to resemble the original data set more than is wanted. Therefore, the actual null distribution may be different from the desired null distribution. This affects the conclusions of the statistical significance test, since the null hypothesis is different from what is wanted. Therefore, the user needs to be careful about convergence, so that the conclusions from the statistical significance test are correct.

Future work on randomization could include new and meaningful ways to randomize other types of data. For example, Ojala *et al.* [76] show some properties of the problem of randomizing relational databases that consists of multiple tables. The tables are connected, or joined, by a query, and the goal is to assess the significance of the query result. Their approach is to randomize the connections between tables, or to randomize individual tables. The problem is more complex than randomizing a single matrix, since one has the possibility, and burden, of choosing which connections or tables to randomize, when a query involves multiple tables. Therefore, the randomization problem is yet unsolved in general.

6.3 Iterative data mining

Iterative data mining was introduced in Publication 2 and the idea is to use statistical significance testing to find a set of results that represent approximately disjoint phenomena in the data set. The problem is very meaningful, since often a data mining algorithm returns a collection of patterns; too many for the user to analyze by hand. A collection of studies have been published, where the aim is to choose the results that are interesting to the user. For example, in frequent itemset mining, instead of returning all the itemsets that have a frequency above some threshold, the user could be presented with closed or maximal frequent itemsets [34, 78]. Both of these sets are a summary of all the frequent itemsets in the data, and therefore, the number of closed or maximal frequent itemsets is much smaller than the number of all the frequent itemsets. However, the methods that choose interesting patterns among a collection of patterns are often problem specific and can not be applied in other scenarios. The iterative data mining setting can be used in a variety of settings, where a part of the result can be fixed in the statistical significance testing.

A drawback of the method is that the selection of patterns to the constraints is yet quite arbitrary. Selecting the pattern with the smallest adjusted p-value may be reasonable. However, it is possible that there are many patterns that have the same adjusted p-value, and then it is not straightforward which of these is selected. Should all of them be selected?

Future work on this problem could include finding new ways of selecting patterns as constraints in the iterative data mining process. A related study, while for different approach, has been published as a technical report [53]. The idea in the report is, instead of iterative data mining, to select the subset of patterns that, when used as constraints in statistical significance testing, will cause no pattern to be statistically significant. This subset of patterns is then considered to be a set of patterns that completely explain the result. While the approach is very different from the iterative data mining, they have similarities and thus the iterative data mining approach was compared to in the report.

Another problem to be considered in the future is that if a pattern is fixed and its statistical significance is tested with different null hypotheses, does one have to correct for multiple hypotheses? The p-values for both null hypotheses are completely dependent, and therefore, the degree of freedom is 1. However, the p-values may still be different. This needs

careful study.

6.4 Biclustering

Two biclustering algorithms were considered in this thesis. The first algorithm was to cluster the rows and columns of the data matrix independently. The algorithm was proven an approximation ratio for binary and real valued matrices, which needs to be multiplied by the approximation ratio of the clustering algorithm if it is approximate. This simple approach provides a very quick and easy way of obtaining a biclustering result, when constant biclusters are sought. The approximation ratio also suggests that a complex algorithm may not be required as such a simple approach is guaranteed to produce reasonable results.

The second biclustering algorithm, BiMDL, was constructed together with biologists to mine their gene expression data. The models of existing algorithms did not suit their purposes, and therefore, a new biclustering algorithm was constructed. The main benefit of the new algorithm is the lack of parameters for the model. When extending the bicluster seeds, MDL makes it possible to have no parameters in that step. Only the seed generation and merging steps have parameters. And if the seeding is replaced with randomizing seeds, the number of parameters is reduced to 2: the number of random seeds and maximum amount of overlap. Furthermore, these parameters are very easy to understand.

The normalization in the description length of a BiMDL bicluster could not be calculated analytically. It is likely that no analytical solution exists, as it involves the distribution of the largest eigenvalue under the solution to the row selection in Section 5.3.3. Therefore, approximation might be the only possibility. The current solution relies on sampling for each possible size of bicluster and height of data matrix, which limits the size of the problem. One possible solution to this is to calculate the normalization for a large number of different matrix and bicluster sizes, study the values carefully and fit some well designed function to it. This way the normalization could be approximated directly and no further sampling would be needed.

Another future work on the BiMDL method could include improving the seed generation algorithm. Now it is possible that a good bicluster with not so high expression values is missed because a seed is not generated that would be extended to that bicluster. Some experiments have been

done on adaptive seeding, where at first a new matrix is generated that has the absolute values of the original matrix. A seed is generated by finding the 3 by 3 bicluster in that matrix with the highest sum of values within the seed. This seed is extended and the cells covered by the resulting bicluster is set to 0. Then, a new seed is found the same way and the procedure is continued. The iteration is stopped when no 3 by 3 seed is found that does not have any 0's in it. This way most of the data matrix is searched for biclusters. However, it is left as future work to consider this method further.

6.5 Future perspectives

Most of the current data mining methods operate only on homogeneous types of data sets, such as matrices, graphs, or time series, and even require homogeneous data within a data set. However, homogeneous data offers only a single viewpoint to the underlying processes or phenomena, and is therefore quite restrictive. Combining different types of data and using them together increases the amount of information and is likely to reveal some of the underlying processes and phenomena that would not be discovered by using homogeneous data alone. As an example, highly heterogeneous data have been used to discover modularity and organization in gene interaction networks [96, 98]. However, the use of heterogeneous data is negligible in comparison to the use of homogeneous data. Future data mining methods are likely to focus on utilizing heterogeneous data.

A related theme is the combination of the results of different data mining methods. This problem was discussed in Publication 2. This problem is similar to using heterogeneous data. Only now the results are heterogeneous. It would be interesting to further study the similarities between completely different approaches to data mining, and the relation between their results. For example, how is the binary PCA projection of a data set related to frequent itemsets? And in general, could the results of completely different data mining methods be combined to better capture all the interesting phenomena and structure in the data? Some work has been done on binary matrices [97], but the problem is still unsolved in general.

Bibliography

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *Proceedings of ACM SIGMOD Conference on Management of Data (SIGMOD'93)*, pages 207–216, Washington, D.C., USA, May 1993. ACM.

[2] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 12, pages 307–328. AAAI Press, Menlo Park, CA, 1996.

[3] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 439–450, New York, NY, USA, 2000. ACM.

[4] Aris Anagnostopoulos, Anirban Dasgupta, and Ravi Kumar. Approximation algorithms for co-clustering. In *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '08, pages 201–210, New York, NY, USA, 2008. ACM.

[5] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

[6] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[7] Yonatan Aumann and Yehuda Lindell. A statistical theory of quantitative association rules. *Journal of Intelligent Information Systems*, 20(3):255–283, 2003.

[8] Simon Barkow, Stefan Bleuler, Amela Prelić, Philip Zimmermann, and Eckart Zitzler. Bicat: a biclustering analysis toolbox. *Bioinformatics*, 22(10):1282–1283, 2006.

[9] Stephen D. Bay and Michael J. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001.

[10] Amir Ben-Dor, Benny Chor, Richard Karp, and Zohar Yakhini. Discovering local structure in gene expression data: The order-preserving submatrix problem. *Journal of Computational Biology*, 10(3–4):373–384, 2003.

[11] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[12] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188, 2001.

[13] Julian Besag and Peter Clifford. Generalized Monte Carlo significance tests. *Biometrica*, 76(4):633–642, 1989.

[14] Sergey Brin, Rejeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. In J. M. Peckman, editor, *Proceedings of ACM SIGMOD Conference on Management of Data (SIGMOD'97)*, pages 265–276, Tucson, AZ, May 1997. ACM.

[15] Ed Bullmore, Chris Long, John Suckling, Jalal Fadili, Gemma Calvert, Fernando Zelaya, T. Adrian Carpenter, and Mick 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.

[16] Stanislav Busygin, Oleg Prokopyev, and Panos M. Pardalos. Biclustering in data mining. *Computers & Operations Research*, 35(9):2964 – 2987, 2008. Part Special Issue: Bio-inspired Methods in Combinatorial Optimization.

[17] Toon Calders and Bart Goethals. Non-derivable itemset mining. *Data Mining and Knowledge Discovery*, 14(1):171–206, 2007.

[18] G. Casella and R.L. Berger. *Statistical inference*. Duxbury advanced series in statistics and decision sciences. Thomson Learning, 2002.

[19] Yizong Cheng and George M. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB'00)*, pages 93–103, 2000.

[20] Hyuk Cho, Inderjit S. Dhillon, Yuqiang Guan, and Suvrit Sra. Minimum sum-squared co-clustering of gene expression data. In *2004 SIAM International Conference on Data Mining (SDM'04)*, 2004.

[21] Mary Kathryn Cowles and Bradley P. Carlin. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

[22] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule discovery from time series. In Rakesh Agrawal, Paul Stolorz, and Gregory Piatetsky-Shapiro, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, pages 16–22, New York, NY, USA, August 1998. AAAI Press.

[23] Patrik D'haeseleer, Shoudan Liang, and Roland Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726, 2000.

[24] Wilfrid J. Dixon and Frank J. Massey. *Introduction to statistical analysis*. McGraw-Hill, New York, 1957. 2.ed.

[25] Sandrine Dudoit, Juliet Popper Shaffer, and Jennifer C. Boldrick. Multiple hypothesis testing in microarray experiments. *Statistical Science*, 18(1):71–103, 2003.

[26] Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[27] Bradley Efron and Robert Tibshirani. *An introduction to the bootstrap*. Chapman and Hall, 1993.

[28] Alexandre Evfimievski. Randomization in privacy preserving data mining. *SIGKDD Explorations Newsletter*, 4(2):43–48, 2002.

[29] R. L. Fernando, D. Nettleton, B. R. Southey, J. C. M. Dekkers, M. F. Rothschild, and M. Soller. Controlling the proportion of false positives in multiple dependent tests. *Genomics*, (166):611–619, 2004.

[30] David Freedman, Roger Pisani, and Robert Purves. *Statistics*. W.W. Norton & Company, 2007.

[31] Aristides Gionis, Heikki Mannila, Taneli Mielikäinen, and Panayiotis Tsaparas. Assessing data mining results via swap randomization. In *Proceedings of the 12th ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

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

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

[34] Karam Gouda and Mohammed Javeed Zaki. Efficiently mining maximal frequent itemsets. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 163–170, Washington, DC, USA, 2001. IEEE Computer Society.

[35] Peter Grünwald. *The Minimum Description Length principle*. MIT Press, 2007.

[36] Niina Haiminen, Heikki Mannila, and Evimaria Terzi. Comparing segmentations by applying randomization techniques. *BMC Bioinformatics*, 8(1):171, 2007.

[37] Greg Hamerly. Making k-means even faster. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*, pages 130–140. SIAM, 2010.

[38] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15:55–86, 2007.

[39] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, January 2004.

[40] David Hand, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining*. MIT Press, 2001.

[41] Sami Hanhijärvi, Kai Puolamäki, and Gemma C. Garriga. Multiple hypothesis testing in pattern discovery. Technical Report TKK-ICS-R21, Helsinki University of Technology, Department of Information and Computer Science, 2009. arXiv:0906.5263v1 [stat.ML].

[42] J.A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[43] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrica*, 57:97–109, 1970.

[44] Sepp Hochreiter, Ulrich Bodenhofer, Martin Heusel, Andreas Mayr, Andreas Mitterecker, Adetayo Kasim, Tatsiana Khamiakova, Suzy Van Sanden, Dan Lin, Willem Talloen, Luc Bijnens, Hinrich W. H. Göhlmann, Ziv Shkedy, and Djork-Arné Clevert. Fabia: factor analysis for bicluster acquisition. *Bioinformatics*, 26(12):1520–1527, 2010.

[45] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[46] Aleksi Kallio, Niko Vuokko, Markus Ojala, Niina Haiminen, and Heikki Mannila. Randomization techniques for assessing the significance of gene periodicity results. *BMC Bioinformatics*, 12(1):330, 2011.

[47] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. Random-data perturbation techniques and privacy-preserving data mining. *Knowledge and Information Systems*, 7(4):387–414, 2005.

[48] Mehmet Koyutürk, Wojciech Szpankowski, and Ananth Grama. Assessing significance of connectivity and conservation in protein interaction networks. *Journal of Computational Biology*, 14(6):747–764, 2007.

[49] Michihiro Kuramochi and George Karypis. An efficient algorithm for discovering frequent subgraphs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1038–1051, 2004.

[50] Stéphane Lallich, Olivier Teytaud, and Elie Prudhomme. Association rule interestingness: measure and statistical validation. *Quality Measures in Data Mining*, pages 251–275, 2006.

[51] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *SIGKDD International Conference of Knowledge Discovery and Data Mining*, 2006.

[52] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[53] Jefrey Lijffijt, Panagiotis Papapetrou, Niko Vuokko, and Kai Puolamäki. The smallest set of constraints that explains the data: a randomization approach. Technical Report TKK-ICS-R31, Aalto University, 2010.

[54] Guimei Liu, Haojun Zhang, and Limsoon Wong. Controlling false positives in association rule mining. *Proceedings of the VLDB Endowment*, 5(2):145–156, October 2011.

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

[56] Xin Lu and David L. Perkins. Re-sampling strategy to improve the estimation of number of null hypotheses in fdr control under strong correlation structures. *BMC Bioinformatics*, 8(157), 2007.

[57] Sara C. Madeira and Arlindo L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[58] Michael Mampaey, Nikolaj Tatti, and Jilles Vreeken. Tell me what i need to know: succinctly summarizing data with itemsets. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 573–581, New York, NY, USA, 2011. ACM.

[59] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Efficient algorithms for discovering association rules. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *Knowledge Discovery in Databases,Papers from the 1994 AAAI Workshop (KDD'94)*, pages 181–192, Seattle, Washington, USA, July 1994. AAAI Press.

[60] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Improved methods for finding association rules. In *Proceedings of the Conference on Artificial Intelligence Research in Finland (STeP'94)*, pages 127–136, Turku, Finland, August 1994.

[61] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovering frequent episodes in sequences. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pages 210–215, Montreal, Canada, August 1995. AAAI Press.

[62] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, November 1997.

[63] Ruth Marcus, Eric Peritz, and K. R. Gabriel. On closed testing procedures with special reference to ordered analysis of variance. *Biometrica*, 63(3):655–660, 1976.

[64] Sergei Maslow. Specificity and stability in topology of protein networks. *Science*, 296(910), 2002.

[65] Nimrod Megiddo and Ramakrishnan Srikant. Discovering predictive association rules. In *Knowledge Discovery and Data Mining*, pages 274–278, 1998.

[66] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Mici Teller, and Edward Teller. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21(1):1087–91, 1953.

[67] Stanley Milgram. The small world problem. *Psychology today*, 2:60–67, 1967.

[68] Shinichi Morishita and Jun Sese. Transversing itemset lattices with statistical metric pruning. In *PODS '00: Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 226–236, New York, NY, USA, 2000. ACM.

[69] Rajeev Motwani and Prabhakar Raghavan. Algorithms and theory of computation handbook. chapter Randomized algorithms, pages 12–12. Chapman & Hall/CRC, 2010.

[70] T.M. Murali and Simon Kasif. Extracting conserved gene expression motifs from gene expression data. In *Pasific Symposium on Biocomputing*, number 8, pages 77–88, 2003.

[71] Mark Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[72] Mark Newman, Dunkan J. Watts, and Steven Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99:2566–2572, 2002.

[73] Jerzy Neyman and Egon S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London*, (231):289–337, 1933.

[74] Bernard V. North, David Curtis, and Pak 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.

[75] Markus Ojala and Gemma C. Garrica. Permutation tests for studying classifier performance. In *Proceedings of the 9th IEEE International Conference on Data Mining*, pages 908–913, 2010.

[76] Markus Ojala, Gemma Garriga, Aristides Gionis, and Heikki Mannila. Evaluating query result significance in databases via randomizations. In *SDM'10: Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 906–917, 2010.

[77] 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*, 4(2):209–230, 2009.

[78] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering frequent closed itemsets for association rules. In *ICDT '99: Proceedings of the 7th International Conference on Database Theory*, pages 398–416, London, UK, 1999. Springer-Verlag.

[79] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 727–734, San Francisco, CA, USA, 2000.

[80] Richard R. Picard and R. Dennis Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984.

[81] Katherine S. Pollard and Mark J. van der Laan. Choice of a null distribution in resampling-based multiple testing. *Journal of Statistical Planning and Inference*, 125(1–2):85–100, 2004. The Third International Conference on Multiple Comparisons.

[82] G. Rasch. On general laws and the meaning of measurement in psychology. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, IV*, pages 321–334. Berkeley: University of Chicago Press, 1980.

[83] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[84] Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, 1989.

[85] Jorma Rissanen. Strong optimality of the normalized ML models as universal codes and information in data. *IEEE Transactions on Information Theory*, 47(5):1712–1717, 2001.

[86] Thomas Schreiber. Constrained randomization of time series data. *Phys. Rev. Lett.*, 80:2105–2108, Mar 1998.

[87] Akdes Serin and Martin Vingron. Debi: Discovering differentially expressed biclusters using a frequent itemset approach. *Algorithms for Molecular Biology*, 6(1):18, 2011.

[88] Juliet Popper Shaffer. Multiple hypothesis testing: A review. *Annual Review of Psychology*, 46:561–584, 1995.

[89] Roded Sharan, Trey Ideker, Brian Kelley, Ron Shamir, and Richard M. Karp. Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. *Journal of Computational Biology*, 12(6):835–846, 2005.

[90] Roded Sharan, Igor Ulitsky, and Ron Shamir. Network-based prediction of protein function. *Molecular Systems Biology*, 3(88), 2007.

[91] Zbynek Sidak. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318):626–633, 1967.

[92] Martin Sill, Sebastian Kaiser, Axel Benner, and Annette Kopp-Schneider. Robust biclustering by sparse singular value decomposition incorporating stability selection. *Bioinformatics*, 27(15):2089–2097, 2011.

[93] R. J. Simes. An improved bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3):751–754, 1986.

[94] John D. Storey. The positive false discovery rate: a bayesian interpretation and the q-value. *The Annals of Statistics*, 31(6):2013–2035, 2003.

[95] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 32–41, New York, NY, USA, 2002. ACM.

[96] Amos Tanay, Roded Sharan, Martin Kupiec, , and Ron Shamir. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proceedings of the National Academy of Sciences*, 101(9):2981–2986, 2004.

[97] Nikolaj Tatti and Jilles Vreeken. Comparing apples and oranges. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6913 of *Lecture Notes in Computer Science*, pages 398–413. Springer Berlin / Heidelberg, 2011.

[98] Olga G. Troyanskaya, Kara Dolinski, Art B. Owen, Russ B. Altman, and David Botstein. A bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae). *Proceedings of the National Academy of Sciences*, 100(14):8348–8353, 2003.

[99] Takeaki Uno, Masashi Kiyomi, and Hiroki Arimura. Lcm ver.3: collaboration of array, bitmap and prefix tree for frequent itemset mining. In *OSDM '05: Proceedings of the 1st international workshop on open source data mining*, pages 77–86, New York, NY, USA, 2005. ACM.

[100] Niko Vuokko and Petteri Kaski. Testing the significance of patterns in data with cluster structure. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 1097 –1102, dec. 2010.

[101] Takashi Washio and Hiroshi Motoda. State of the art of graph-based data mining. *ACM SIGKDD Explorations Newsletter*, 5(1):59–68, 2003.

[102] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[103] Geoffrey Webb. Layered critical values: a powerful direct-adjustment approach to discovering significant patterns. *Machine Learning*, 71:307–323, 2008.

[104] Geoffrey I. Webb. Discovering significant rules. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 434–443, New York, NY, USA, 2006. ACM.

[105] Geoffrey I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.

[106] Peter H. Westfall and S. Stanley Young. *Resampling-based multiple testing: examples and methods for p-value adjustment*. Wiley, 1993.

[107] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey McLachlan, Angus Ng, Bing Liu, Philip Yu, Zhi-Hua Zhou, Michael Steinbach, David Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14:1–37, 2008. 10.1007/s10115-007-0114-2.

[108] Xin Xu, Ying Lu, Anthony K. H. Tung, and Wei Wang. Mining shifting-and-scaling co-regulation patterns on gene expression profiles. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, 2006.

[109] Daniel Yekutieli and Yoav Benjamini. Resampling-based false discovery rate controlling multiple test procedures for correlated test statistics. *Journal of Statistical Planning and Inference*, 82:171–196, 1999.

[110] Xiaowei Ying and Xintao Wu. Randomizing social networks: a spectrum preserving approach. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 739–750, 2008.

[111] Xiaowei Ying and Xintao Wu. Graph generation with predescribed feature constraints. In *Proceedings of the 2009 SIAM Internatioal Conference on Data Mining*, 2009.

[112] Hong Zhang, Balaji Padmanabhan, and Alexander Tuzhilin. On the discovery of significant statistical quantitative rules. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383, New York, NY, USA, 2004. ACM.

[113] Mengsheng Zhang, Wei Wang, and Jinze Jiu. Mining approximate order preserving clusters in the presence of noise. In *Proceedings of the 24th International Conference on Data Engineering (ICDE'08)*, pages 160–168, 2008.

DISSERTATIONS IN INFORMATION AND COMPUTER SCIENCE

TKK-ICS-D21 Sorjamaa, Antti.
Methodologies for Time Series Prediction and Missing Value Imputation. 2010.

TKK-ICS-D22 Schumacher, André
Distributed Optimization Algorithms for Multihop Wireless Networks. 2010.

Aalto-DD99/2011 Ojala, Markus
Randomization Algorithms for Assessing the Significance of Data Mining Results. 2011.

Aalto-DD111/2011 Dubrovin, Jori
Efficient Symbolic Model Checking of Concurrent Systems. 2011.

Aalto-DD118/2011 Hyvärinen, Antti
Grid Based Propositional Satisfiability Solving. 2011.

Aalto-DD136/2011 Brumley, Billy Bob
Covert Timing Channels, Caching, and Cryptography. 2011.

Aalto-DD11/2012 Vuokko, Niko
Testing the Significance of Patterns with Complex Null Hypotheses. 2012.

Aalto-DD19/2012 Reunanen, Juha
Overfitting in Feature Selection: Pitfalls and Solutions. 2012.

Aalto-DD33/2012 Caldas, José
Graphical Models for Biclustering and Information Retrieval in Gene Expression Data. 2012.

Aalto-DD45/2012 Viitaniemi, Ville
Visual category detection: an experimental perspective. 2012

Data mining is the process of discovering structures and regularities in data. It has become increasingly important with the vast amounts of data that is gathered. However, without proper assessment for significance, the found patterns can be spurious and artifacts of mere chance. This thesis discusses the aspects of statistical significance testing in data mining by using randomization to create meaningful statistical tests. The focus is on the problem of multiple hypothesis testing, which is inherent to data mining where several patterns are tested simultaneously. A method to solve the problem is presented, which is very general and can be applied in various scenarios. Randomization methods are then presented, that can be used with the multiple hypothesis testing method. The utility and ease of use of the presented methods are displayed through practical applications.

ISBN 978-952-60-4604-4
ISBN 978-952-60-4605-1 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Information and Computer Science
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS

Aalto-DD 51/2012