View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Aaltodoc Publication Archive

Department of Information and Computer Science

Randomization Algorithms for Assessing the Significance of Data Mining Results

Markus Ojala

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 99/2011

Randomization Algorithms for Assessing the Significance of Data Mining Results

Markus Ojala

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 E at the Aalto University School of Science (Espoo, Finland) on the 12th of November 2011 at 12 noon.

Aalto University
School of Science
Department of Information and Computer Science

Supervisor
Heikki Mannila

Instructor
Heikki Mannila

Preliminary examiners
Jean-Francois Boulicaut, University of Lyon, France
Bart Goethals, University of Antwerp, Belgium

Opponent
Rosa Meo, University of Torino, Italy

Aalto University publication series
DOCTORAL DISSERTATIONS 99/2011

© Markus Ojala

ISBN 978-952-60-4323-4 (pdf)
ISBN 978-952-60-4322-7 (printed)
ISSN-L 1799-4934
ISSN 1799-4942 (pdf)
ISSN 1799-4934 (printed)

Aalto Print
Helsinki 2011

Finland

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

Aalto University

Abstract

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

Author
Markus Ojala

Name of the doctoral dissertation
Randomization Algorithms for Assessing the Significance of Data Mining Results

Publisher School of Science

Unit Department of Information and Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 99/2011

Field of research Computer and Information Science

Manuscript submitted 12 April 2011 **Manuscript revised**

Date of the defence 12 November 2011 **Language** English

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

Abstract

Data mining is an interdisciplinary research area that develops general methods for finding interesting and useful knowledge from large collections of data. This thesis addresses from the computational point of view the problem of assessing whether the obtained data mining results are merely random artefacts in the data or something more interesting.

In randomization based significance testing, a result is compared with the results obtained on randomized data. The randomized data are assumed to share some basic properties with the original data. To apply the randomization approach, the first step is to define these properties. The next step is to develop algorithms that can produce such randomizations. Results on the real data that clearly differ from the results on the randomized data are not directly explained by the studied properties of the data.

In this thesis, new randomization methods are developed for four specific data mining scenarios. First, randomizing matrix data while preserving the distributions of values in rows and columns is studied. Next, a general randomization approach is introduced for iterative data mining. Randomization in multi-relational databases is also considered. Finally, a simple permutation method is given for assessing whether dependencies between features are exploited in classification.

The properties of the new randomization methods are analyzed theoretically. Extensive experiments are performed on real and artificial datasets. The randomization methods introduced in this thesis are useful in various data mining applications. The methods work well on different types of data, are easy to use, and provide meaningful information to further improve and understand the data mining results.

Keywords data mining, randomization, significance testing, MCMC, matrix, relational database, clustering, classification, iterative analysis

ISBN (printed) 978-952-60-4322-7 **ISBN (pdf)** 978-952-60-4323-4

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

Location of publisher Espoo **Location of printing** Helsinki **Year** 2011

Pages 178 **The dissertation can be read at** http://lib.tkk.fi/Diss/

A? Aalto-yliopisto

Tiivistelmä

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

Tekijä
Markus Ojala

Väitöskirjan nimi
Satunnaistusalgoritmeja tiedonlouhinnan tulosten merkitsevyyden arviointiin

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietojenkäsittelytieteen laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 99/2011

Tutkimusala Informaatiotekniikka

Käsikirjoituksen pvm 12.04.2011 **Korjatun käsikirjoituksen pvm**

Väitöspäivä 12.11.2011 **Kieli** Englanti

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

Tiivistelmä

Tiedonlouhinta on monitieteellinen tutkimusalue, joka kehittää yleisiä menetelmiä mielenkiintoisen ja hyödyllisen tiedon löytämiseksi suurista tietoaineistoista. Väitöskirjassa tarkastellaan laskennallisesta näkökulmasta ongelmaa, miten määritetään, ovatko tiedonlouhinnan tulokset vain satunnaisilmiöitä aineistossa vai jotakin mielenkiintoisempaa.

Satunnaistukseen perustuvassa merkitsevyystestauksessa tulosta verrataan satunnaistetuilla aineistoilla saatuihin tuloksiin. Satunnaistetuilla aineistoilla ja alkuperäisellä aineistolla oletetaan olevan joitain yhteisiä perusominaisuuksia. Lähestymistavan soveltamiseksi täytyy ensin määritellä, mitä nämä ominaisuudet ovat. Seuraavana vaiheena on kehittää menetelmiä, joilla voidaan tuottaa haluttuja satunnaistuksia. Alkuperäisellä aineistolla saadut tulokset, jotka poikkeavat selvästi satunnaistetuilla aineistoilla saaduista tuloksista, eivät selity suoraan tutkituilla aineiston ominaisuuksilla.

Väitöskirjassa kehitetään uusia satunnaistusmenetelmiä neljään erityiseen ongelmaan. Ensimmäiseksi tutkitaan matriisiaineistojen satunnaistamista, jossa rivien ja sarakkeiden arvojakaumat säilytetään likimääräisesti. Seuraavaksi esitellään yleinen satunnaistamistapa toistuvaan tiedonlouhintaan. Lisäksi tarkastellaan satunnaistamista relaatiotietokannoissa. Lopuksi esitellään permutaatiomenetelmä määrittämään, hyödynnetäänkö luokittelussa muuttujien välisiä riippuvuuksia.

Satunnaistusmenetelmien ominaisuuksia analysoidaan teoreettisesti. Kattavia kokeita suoritetaan oikeilla ja keinotekoisilla aineistoilla. Väitöskirjassa esitellyt satunnaistusmenetelmät ovat hyödyllisiä monenlaisissa tiedonlouhintasovelluksissa. Menetelmät toimivat hyvin erityyppisillä aineistoilla, ovat helppokäyttöisiä ja antavat mielekästä tietoa tiedonlouhinnan tulosten parantamiseen ja ymmärtämiseen.

Avainsanat tiedonlouhinta, satunnaistus, merkitsevyystestaus, MCMC, matriisi, relaatiotietokanta, ryvästys, luokittelu, toistuva analyysi

ISBN (painettu) 978-952-60-4322-7 **ISBN (pdf)** 978-952-60-4323-4

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

Julkaisupaikka Espoo **Painopaikka** Helsinki **Vuosi** 2011

Sivumäärä 178 **Luettavissa verkossa osoitteessa** http://lib.tkk.fi/Diss/

Contents

List of Publications

This thesis consists of an introductory part and the following five publications. Other publications by the author include [OVK+08, OG09, KVO+11].

I Markus Ojala, Niko Vuokko, Aleksi Kallio, Niina Haiminen, and Heikki Mannila. Randomization Methods for Assessing Data Analysis Results on Real-Valued Matrices. *Statistical Analysis and Data Mining*, 2(4):209–230, 2009.

II Markus Ojala. Assessing Data Mining Results on Matrices with Randomization. In *ICDM'10: Proceedings of the 10th IEEE International Conference on Data Mining*, pages 959–964. IEEE, 2010.

III Sami Hanhijärvi, Markus Ojala, Niko Vuokko, Kai Puolamäki, Nikolaj Tatti, and Heikki Mannila. Tell Me Something I Don't Know: Randomization Strategies for Iterative Data Mining. In *KDD'09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 379–388. ACM, 2009.

IV Markus Ojala, Gemma C. Garriga, Aristides Gionis, and Heikki Mannila. Evaluating Query Result Significance in Databases via Randomizations. In *SDM'10: Proceedings of the 10th SIAM International Conference on Data Mining*, pages 906–917. SIAM, 2010.

V Markus Ojala and Gemma C. Garriga. Permutation Tests for Studying Classifier Performance. *Journal of Machine Learning Research*, 11(Jun): 1833–1863, 2010.

Preface

This thesis has been carried out in the Department of Information and Computer Science at Aalto University. I am grateful for the funding and working possibilities offered by the department, Finnish Centre of Excellence for Algorithmic Data Analysis Research, and Helsinki Graduate School in Computer Science and Engineering. The work has also been supported by personal grants from Finnish Foundation for Technology Promotion, Nokia Foundation, and Emil Aaltonen's Foundation.

It has been a great privilege to have Professor Heikki Mannila as my supervisor and instructor. I am deeply grateful to Heikki and all my co-authors without whom this thesis would not exist. My good friend and colleague Niko Vuokko deserves my best compliments for our mutual support. Special thanks belong to Kai Puolamäki and Gemma Garriga for all the help and fun during my doctoral studies. Furthermore, I have truly enjoyed the pleasant working environment created by various magnificent colleagues, including Jefrey Lijffijt, Panagiotis Papapetrou, Sami Hanhijärvi, Esa Junttila, Jaakko Hollmén, Mika Sulkava, Mikko Korpela, Janne Toivola, Jaakko Talonen, and all weekly floorball players, among others.

Finally, I would like to thank my parents, family and friends for all the support and joy they bring to my life. Especially, the deep understanding and love from my wife Paula Vahermaa has encouraged me towards continuous study of my personal life behind the science.

Espoo, August 31, 2011,

Markus Ojala

CHAPTER 1

Introduction

In this chapter, we give motivation for using randomization methods to assess data analysis results and outline the contributions of this thesis.

1.1 Motivation

Data mining is an interdisciplinary research area developing general methods for finding interesting and useful knowledge from large real life collections of data [HMS01]. The data mining community has introduced various efficient algorithms for automatically extracting knowledge from databases. The algorithms are widely used in many application fields, such as in information technology and bioinformatics where the traditional methods are not sufficient.

Assessing the quality and significance of the results is an important part of scientific work. Traditional statistics has long been considering the issue of significance testing. There are various traditional statistical methods for measuring the significance [FPP07, GCSR03, MH06]. However, significance testing has been given less attention in the data mining community, partly because the traditional statistical tests are not directly applicable to most of the data mining tasks.

In a randomization-based significance testing a result is considered to be significant if it is unlikely to obtain such a result on randomized data sharing some basic characteristics with the original data [Goo00, Edg95, Bes04, CB01]. Randomization methods are applicable to many real world problems. For example, in some applications, a result in a binary matrix can be considered to be interesting if it is not explained by the row and column sums. There are various randomization-based approaches for significance testing of data mining results on binary matrices and contingency tables [CC03, GMMT07].

	Milk	Bread	Banana	Cheese	Ham	Salmon
Store 1	0.69	2.49	0.99	5.49	6.49	5.99
Store 2	0.79	2.79	1.19	6.69	7.13	6.99
Store 3	0.79	2.49	1.29	6.39	7.59	6.49
Store 4	0.89	2.89	0.99	6.59	6.99	7.49
Store 5	0.89	3.19	1.49	7.09	7.39	11.69
Store 6	0.99	3.59	1.79	8.09	8.69	9.59
Store 7	0.99	3.29	1.69	6.89	9.19	12.99
Store 8	1.19	4.59	1.99	8.49	8.59	16.99
Store 9	1.19	4.29	2.49	8.99	9.39	18.99
Store 10	1.29	3.99	2.19	7.79	9.99	14.49

Table 1.1: An example dataset of product prices in different grocery stores. The prices are in euros per kilogram for the same product in each store.

However, there are many data analysis tasks for which randomization methods do not exist. For example, in bioinformatics, gene expression measurements produce large real-valued matrices where data mining methods are used to find relevant information [BH02, BV01]. Nevertheless, the problem of randomizing real-valued matrices for assessing whether data mining results are independent of basic row and column statistics has not been previously studied in the literature.

In Table 1.1, we give an artificial example of retail prices of basic food products in ten different grocery stores. This is an example of a real-valued matrix with different types of features. From such dataset, we can for example observe that the prices of milk and bread have a strong correlation, 0.93, i.e., if milk is cheap, bread is also cheap in the same store. But is this observation interesting or is it just a consequence of some simpler fact? If we look the dataset more closely, we notice that some stores are generally much cheaper than the other stores. That is, the strong correlation between the prices of milk and bread is likely to be explained by the general price levels of the stores. Thus some basic background properties of the data can explain other seemingly significant patterns in the data. Later, we will consider this example more closely and show how to randomize a dataset of this type.

The objective of this thesis is to develop new randomization-based significance testing methods for different types of data and for various data mining tasks. As data types, we consider different kinds of matrices, relational databases, and labeled data. The general computational task in this thesis is the following.

Problem 1.1 (General randomization)**.** *Given a dataset D, generate a randomized dataset $\widehat{D}$ chosen independently and uniformly from the set of all datasets having the same values of the predescribed basic statistics as the original dataset D.*

There are three main phases in developing new randomization methods for a given type of data. The first phase is to analyze the underlying phenomena and distinguish what are the general features of the given type of data and what features are specific to each dataset. The aim is to formulate what type of random datasets should be used in the sense that if results similar to an original result are obtained on randomized data, then the original result is considered to be uninteresting. Otherwise, the result is considered to be specific to the given data and not explained by the general features of the data.

The second phase is to develop methods that produce randomized datasets having the properties formulated in the first phase. We concentrate on Markov chain Monte Carlo methods where the randomized datasets are produced by starting from the original data and performing local transformations until the chain has converged [Bes04]. The tests are valid even without proper convergence [BC89]. However, better convergence provides more powerful test. Thus, the main problem is to control that the methods produce well-randomized datasets preserving the formulated properties with sufficient accuracy.

The last phase is to analyze the algorithms both theoretically and empirically. The methods have different running times and they produce different degrees of randomness. The usefulness of the methods are tested on artificial and real datasets, and the methods are compared to other existing methods that are mainly simple permutation methods. If the methods do not yet work satisfactorily, the developing process is restarted either from the first or second step and iterated until a good method is found.

In this thesis we employ the above procedure for different types of data and for various data mining tasks. We formulate new statistical null-models for diverse data mining applications and develop new randomization methods to implement the corresponding statistical tests. Theoretical properties of the methods are studied in detail. The developed randomization algorithms are tested and applied to real world applications where various data mining methods are used. The main result of the thesis is a set of practical algorithms for randomizing different types of data for significance testing of data mining results.

1.2 Summaries and Contributions of Publications

Brief summaries of the original publications included in this thesis are given below with detailed contributions by the author.

Publication I

Markus Ojala, Niko Vuokko, Aleksi Kallio, Niina Haiminen, and Heikki Mannila. Randomization Methods for Assessing Data Analysis Results on Real-Valued Matrices. *Statistical Analysis and Data Mining*, 2(4):209–230, 2009.

In Publication I we consider the problem of assessing data mining results on real-valued matrices containing rows of similar type as well as columns of similar type. We introduce two tasks for randomizing real-valued matrices. The first task is to generate randomized real-valued matrices having approximately the same value distributions in each row and column as the given original matrix. In the second task only the row and column means and variances are preserved in randomized matrices. These tasks are generalizations of the randomization approach of binary matrices [GMMT07] where the number of ones is preserved in each row and column. We give two algorithms, *SwapDiscretized* and *GeneralMetropolis*, with various local modifications and difference measures for solving the tasks. The methods are analyzed theoretically and empirically showing the usefulness of the concept. In the preliminary version [OVK+08] of Publication I only the second task was considered with simple algorithms and concise theoretical and experimental evaluation.

The concept and the basic approach of randomizing real-valued matrices were developed jointly by all authors. In the preliminary version [OVK+08] of Publication I, the methods were developed jointly with Niko Vuokko, the experiments were mainly performed by the author, and the text written by all authors. The idea of the new randomization task in Publication I was developed jointly. Most of the other improvements in Publication I, including the new algorithms, experiments, and analysis, were made by the author of this thesis.

Publication II

Markus Ojala. Assessing Data Mining Results on Matrices with Randomization. In *ICDM'10: Proceedings of the 10th IEEE International Conference on Data Mining*, pages 959–964. IEEE, 2010.

In Publication II the concept of Publication I is generalized to a practical tool for assessing results on different types of matrices. A new approach is introduced for randomizing matrices containing features measured using different scales. In such matrices, the feature-wise ranks in each row are preserved in randomization as well as the value distributions of the features. A new algorithm, *SwapConstrained*, is given that can preserve the value distributions more accurately. Furthermore, it supports matrices containing dissimilar fea-

tures, nominal values, non-Gaussian value distributions, missing values, and sparse structure. Extensive experiments on various real-life datasets show the usefulness of the improved approach.

The author of the thesis is the sole author of Publication II.

Publication III

Sami Hanhijärvi, Markus Ojala, Niko Vuokko, Kai Puolamäki, Nikolaj Tatti, and Heikki Mannila. Tell Me Something I Don't Know: Randomization Strategies for Iterative Data Mining. In *KDD'09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 379–388. ACM, 2009.

In Publication III we consider the problem of finding independent patterns from a single dataset. We give a general approach where the previously found patterns are fixed in randomization and other patterns not explained by them are considered to be interesting. An iterative strategy is given for finding significant patterns. Specifically, the approach is applied to binary matrices where the clustering structure or itemset frequencies are preserved in addition to the row and column sums. By experimental analysis it is observed that some very different types of data mining patterns can explain each other.

The concept of the iterative data mining and the basic approach and algorithms in Publication III were developed jointly by all authors. The detailed methods were developed by the author and Sami Hanhijärvi. These two also contributed the most to the experiments and analysis. The final text was written by all the authors.

Publication IV

Markus Ojala, Gemma C. Garriga, Aristides Gionis, and Heikki Mannila. Evaluating Query Result Significance in Databases via Randomizations. In *SDM'10: Proceedings of the 10th SIAM International Conference on Data Mining*, pages 906–917. SIAM, 2010.

In Publication IV we apply the randomization approach to assess queries in multi-relational databases. In multi-relational databases there is no unambiguous way to randomize. We propose to randomize each binary-relation in the query and each connection between two relations in the query separately. In this way we can find what is the structural impact of each relation to the query. Theoretical connections between the randomizations are shown. Various hypotheses are studied on a MOVIELENS database by using the approach.

The main ideas, concepts, and theoretical framework in Publication IV were developed jointly by all authors. The analytical results were developed and written mostly by the author. The experiments were performed and written by the author. The rest of the paper was written jointly.

Publication V

Markus Ojala and Gemma C. Garriga. Permutation Tests for Studying Classifier Performance. *Journal of Machine Learning Research*, 11(Jun): 1833–1863, 2010.

In Publication V we use permutation tests for studying classifier performance. We propose a new test for assessing whether the classifier exploits the dependencies between the features to improve the classifier accuracy. The randomized datasets are produced by permuting the columns inside each class. For comparison we also study the traditional permutation test that assesses whether the classifier has learned a real connection between the data and the class labels. The properties of the tests are extensively analyzed theoretically and empirically. In the preliminary version [OG09] of Publication V the new test was briefly introduced with short analysis.

The idea of the new test was developed jointly with Gemma Garriga. The theoretical and experimental analysis of the methods was performed and written by the author. The rest of the paper was written jointly.

1.3 Outline of the Thesis

The purpose of the introductory part of this thesis (Chapters 1–5) is to provide the necessary background for a reader to understand the publications and to summarize the main results of the publications. We discuss in further detail the related work, introduce the contributions of the publications and perform some new experiments.

In Chapter 2, we first discuss data mining in general and describe a few common data mining tasks and methods including clustering, principal component analysis, frequent itemset mining, and classification that are used throughout this thesis. Next, we discuss how randomization can be applied in significance testing and introduce the basic approach with empirical p-values that is used in all publications of this thesis.

In Chapter 3, we discuss different approaches for generating randomized datasets. First we give a brief background to Markov chain Monte Carlo methods. Next, we discuss how the randomized samples are produced in the publications of this thesis.

In Chapter 4, we apply the new randomization methods to different data mining problems. We discuss the practical properties of the methods and give a few case examples. Chapter 5 concludes the introductory part of the thesis with discussion on the randomization approaches studied in the thesis.

CHAPTER 2

Randomization and Data Mining

In this chapter, we introduce the basic concepts of this thesis. First, we discuss data mining in general and describe a few common data mining algorithms that are also used in the experiments of this thesis. After that we discuss how to apply randomization to assess the significance of data mining results, and give a small example with a binary dataset.

2.1 Data Mining Methods

Data mining is the process of extracting new useful information from large amount of data [HMS01]. *Machine learning* is a closely related field that shares a majority of the methods with data mining [DHS00]. In this section, we describe a few traditional data mining methods [WKRQ+07] for some specific problems. Note, however, that the randomization based significance testing approach given in this thesis can be used to assess the results of any data mining or machine learning method. Here, we just give some examples of basic data mining methods. They are, however, widely used in many real applications and also in the experiments of this thesis.

Consider a dataset D that can be, for example, an $n \times d$ binary table or an $n \times d$ real-valued matrix. Throughout this thesis, the rows of the tables correspond to sample points (observations) and the columns to dimensions (features). We denote by D_i the row i of the dataset D, by D^j the column j of D, and by D_{ij} the element of D in the row i and column j. Assume that some data mining task, such as clustering, is performed on D. We assume that the quality strength of the result can be described by a single number $\mathcal{S}(D) \in \mathbb{R}$ that we call a *structural measure* of D:

Definition 2.1. *A structural measure $\mathcal{S}$ for a data analysis task is a function that maps each dataset D to a real number $\mathcal{S}(D)$.*

The structural measure can be, for example, the clustering error of the matrix, the correlation between some specific columns, or the number of frequent itemsets. Any function can be used, as long as it can be summarized by one number so that smaller (or larger) values mean stronger presence of the measured structure. In the following subsections, we will give examples of structural measures for various data mining methods.

2.1.1 Clustering

In clustering, the task is to assign a set of observations into a number of subsets called *clusters* so that the observations in the same cluster resemble each other in some sense. There are various clustering methods for different types of data with different objectives. However, in this thesis we only discuss the most classical *k-means clustering* problem in detail. See Xu and Wunsch [XW05] and Berkhin [Ber06] for good surveys of different clustering algorithms. Clustering algorithms are used in Publications I, II, and III.

The k-means clustering

The k-means clustering is a specific problem of clustering. It is also a name of a simple algorithm introduced by Lloyd in 1957 [Llo57, Llo82] for solving the problem. In k-means clustering the objective is to divide the samples into k clusters where each sample belongs to the cluster with the nearest center. The number of clusters k is usually decided beforehand. There exist methods for finding a suitable k by calculating the clustering for many different values of k and selecting the best k by using different validation measures, such as Bayesian information criterion [Sch78, PM00]. However, in this thesis, for simplicity, we will use fixed values for k, as the problem of model selection is not in the scope of this thesis.

In k-means clustering, the task is to partition all n samples of dataset $D \in \mathbb{R}^{n \times d}$ into k clusters $\mathcal{C} = \{C_1, \ldots, C_k\}$ while minimizing the sum of the within-cluster squared distances,

$$\mathcal{S}_{k\text{-means}}(D) = \sum_{j=1}^{k} \sum_{i \in C_j} \|D_i - \mu_j\|_{L_2}^2, \tag{2.1}$$

where μ_j is the mean vector of the data points in cluster C_j. The Euclidean L_2 distance is used to measure the similarity. Thus, a small clustering error

means a good clustering structure. The clustering error (2.1) can be used as a structural measure $\mathcal{S}$.

There are various algorithms for optimizing the clustering error (2.1). The basic k-means method by Lloyd is given in Algorithm 2.1. It starts by randomly selecting k cluster centers μ_j. The algorithm repeatedly assigns each point into the cluster C_j with the nearest center μ_j, and after that updates each cluster center μ_j to be the average of the points in the corresponding cluster C_j.

Algorithm 2.1 The k-means algorithm

Input: Dataset $D \in \mathbb{R}^{n \times d}$, number of clusters k
Output: Clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$, cluster centers $\mu = \{\mu_1, \ldots, \mu_k\}$

```
Randomly pick k cluster centers μ_j ∈ R^d for j = 1,...,k
while not converged do
    for j ← 1 to k do
        C_j ← {i | j = arg min_l ||D_i − μ_l||_{L_2}}
    end for
    for j ← 1 to k do
        μ_j ← (1/|C_j|) Σ_{i∈C_j} D_i
    end for
end while
return C, μ
```

The clustering error decreases in each iteration of the k-means algorithm. However, the method can converge to a local minimum. To alleviate this problem, the clustering is usually repeated a few times with different random starts, and only the best clustering is reported. Traditionally, the cluster centers μ_j are initialized to be a random subset of the original data points D_i.

Various optimizations has been introduced to the traditional k-means algorithm. The k-means++ algorithm by Arthur and Vassilvitskii [AV07] is a simple modification that provides a significant improvement to the performance of the clustering. Their method only changes the initialization of the cluster centers. The initial centers are selected step-by-step. The first center μ_1 is chosen uniformly at random from the original n points. Let $\delta(x)$ be the distance from a point $x \in \mathbb{R}^d$ to the closest center μ_j already selected. The next center μ_j is selected from the original n data points with probability

$$\Pr(\mu_j = D_i) = \frac{\delta(D_i)^2}{\sum_{l=1}^{n} \delta(D_l)^2}.$$

A recent pruning method by Hamerly [Ham10] can be used to make the k-means even faster by using some clever optimization to prune out unnecessary

calculation. It can be combined also with the k-means++ algorithm. However, this optimization is not used in the experiments of this thesis.

2.1.2 Principal Component Analysis

Principal component analysis (PCA) is a classical linear dimensionality reduction method [Pea01, Hot33]. The task is to linearly map the original data into a lower dimensional space while preserving as much as possible of the original structure. There are also various nonlinear dimensionality reduction methods, see, for example, the book by Lee and Verleysen [LV07] for a good tutorial. Principal component analysis is used in Publications I and II.

PCA can be derived from different viewpoints. One way is to maximize the preserved variance under some constraints as introduced by Hotelling in 1933 [Hot33]. Assume that the columns of D have zero means. The objective is to find a projection to a lower dimension $\widehat{d} < d$ such that the points in the projected data are uncorrelated and preserve as much of the variance as possible. Let $W \in \mathbb{R}^{d \times \widehat{d}}$ be the corresponding orthonormal axis change, that is, $W^T W = I_{\widehat{d}}$, and $\widetilde{D} = DW$ the projected data. Then the covariance matrix of the projected data $\widetilde{D}$ is

$$\begin{aligned} \mathrm{Cov}(\widetilde{D}) &= \frac{1}{n}\widetilde{D}^T\widetilde{D} \\ &= \frac{1}{n}W^T D^T DW \\ &= W^T \mathrm{Cov}(D) W. \end{aligned}$$

Let $\mathrm{Cov}(D) = V\Lambda V^T$ be the eigenvalue decomposition [TI97] of the covariance matrix of D. We assume that the eigenvalues are sorted in decreasing order in the diagonal of Λ. Then the maximum variance is obtained by choosing the first $\widehat{d}$ eigenvectors of $\mathrm{Cov}(D)$ as the projection, that is,

$$W = V I_{d \times \widehat{d}},$$

which gives

$$\mathrm{Cov}(\widetilde{D}) = I_{\widehat{d} \times d} \Lambda I_{d \times \widehat{d}}.$$

The method for PCA is presented in Algorithm 2.2. The notation 1_n corresponds to a column vector of n ones and $I_{d \times \widehat{d}}$ to an identity matrix of size $d \times \widehat{d}$. In the first line, the average of each column is subtracted. The method outputs also the fraction ρ of the explained variance. If it is close to one, the data contains a clear inner structure and the intrinsic dimension is really close to $\widehat{d}$. The value of ρ can be used directly as a structural measure.

Algorithm 2.2 Principal component analysis

Input: Dataset $D \in \mathbb{R}^{n \times d}$, new dimension $\widehat{d} < d$
Output: Projected data $\widetilde{D} \in \mathbb{R}^{n \times \widehat{d}}$, fraction ρ of the explained variance
1: $D \leftarrow D - \frac{1}{n} 1_n 1_n^T D$
2: $V \Lambda V^T \leftarrow \mathrm{Evd}(D)$
3: $W \leftarrow V I_{d \times \widehat{d}}$
4: $\widetilde{D} \leftarrow DW$
5: $\rho \leftarrow \sum_{i=1}^{\widehat{d}} \lambda_i / \sum_{i=1}^{d} \lambda_i$
6: **return** $\widetilde{D}$, ρ

2.1.3 Frequent Itemset Mining

Frequent itemset mining is a popular and well researched problem in data mining [AIS93]. It is especially used as a subtask in *association rule learning*. Consider a binary dataset $D \in \{0,1\}^{n \times d}$ where the rows correspond to transactions and the columns to items. The element D_{ij} is 1 if the item j is in the corresponding transaction i, and 0 otherwise. For example, in market basket data each transaction corresponds to a basket of products a customer has bought.

The task of frequent itemset mining is to find all sets of items that occur frequently together. For example, $\{\mathrm{milk}, \mathrm{bread}\}$ can be a frequent itemset. In association rule learning, the task is to find itemsets whose existence predicts other itemsets. For example, in market basket data one possible rule is $\{\mathrm{mustard}, \mathrm{beer}\} \rightarrow \{\mathrm{sausage}\}$, saying that if a customer buys both mustard and beer, he or she is likely to buy also sausages. In frequent itemset mining, the number of frequent itemsets or the frequency of some specific itemset can be used as structural measures.

Apriori is a classical algorithm for finding frequent itemsets from binary data [AMS+96]. It finds all itemsets whose frequencies are larger than or equal to a given threshold f_t. It proceeds in a breadth-first search manner by generating candidate itemsets of size k from frequent itemsets of size $k-1$ such that an itemset is candidate if its $k-1$ sized subsets are frequent. After that the method prunes infrequent itemsets of size k from the candidates. A sketch of the method is given in Algorithm 2.3 without any optimizations. Frequent itemset mining is used in Publication III.

2.1.4 Mining in Relational Databases

Next, we briefly describe relational databases and how queries are used to answer specific questions in such databases [RG02]. A relational database

Algorithm 2.3 Apriori

Input: Dataset $D \in \{0,1\}^{n \times d}$, frequency threshold f_t
Output: Family $\mathcal{F}$ of itemsets whose frequency in D is at least f_t

1: $\mathcal{F}_1 = \{j \mid \sum_i D_{ij} \geq f_t\}$
2: $k \leftarrow 2$
3: **while** $\mathcal{F}_{k-1} \neq \emptyset$ **do**
4: $\quad \mathcal{C}_k \leftarrow$ Generate candidates from $\mathcal{F}_{k-1}$
5: $\quad \mathcal{F}_k \leftarrow \{F \in \mathcal{C}_k \mid \sum_i \prod_{j \in F} D_{ij} \geq f_t\}$
6: **end while**
7: **return** $\mathcal{F} = \cup \mathcal{F}_k$

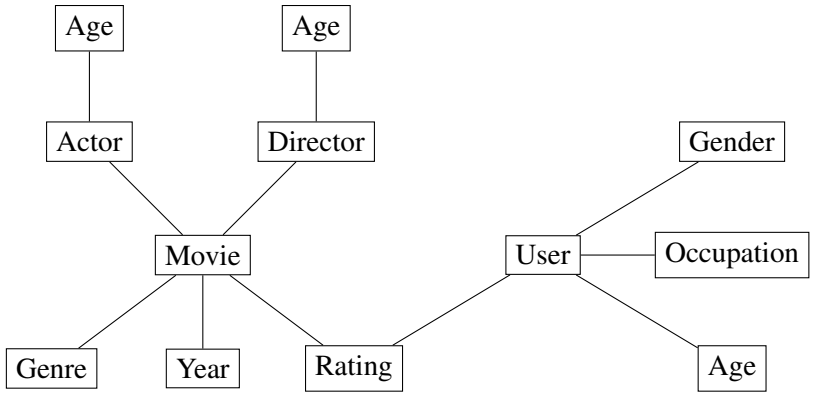


Figure 2.1: A simplified presentation of an example movie database with multiple relations.

consists of multiple tables, *relations*, each containing a set of tuples having the same attributes. In this thesis we concentrate mainly on binary relations that consist of pairs $A \subseteq I \times J$ between two sets I and J. For example, a movie database could contain relations such as directors direct movies, movies are classified by genres, and users have rated certain movies, see Figure 2.1.

In relational databases queries are used to access the data and they can be used as part of the data mining process. For example, a query can be used to check the number of people who have watched movies directed by a specific director. Queries consist of three basic operations: *joins*, *selections* and *projections*. A join between two relations A and B, denoted by $A \bowtie B$, is the set of all combinations of entries in A and B that are equal on their common attributes. Selection $\sigma_\phi(A)$ returns all the tuples t in the relation A for which $\phi(t)$ holds.

Projection $\pi_X(A)$ restricts tuples of A to attributes in X. For example, the query

$$q = \pi_{\text{Movie,Age}}(\sigma_{\text{Genre=Drama}}(\text{Genre-Movie} \bowtie \text{Movie-Director} \bowtie \text{Director-Age})$$

returns the ages of the directors of the drama movies. The concepts of relational databases and queries are used in Publication IV.

2.1.5 Classification

In classification, items are assigned into given groups based on their properties and a training set of samples whose real groups are known. For example, given gene expression measurements from a patient, the task is to classify whether the patient is "sick" or "healthy". Formally, given a training set $\{(X_1, y_1), \ldots, (X_n, y_n)\}$, where each data point $X_i \in \mathcal{X}$ has a known class label $y_i \in \mathcal{Y}$, the task is to learn a *classifier* $f : \mathcal{X} \to \mathcal{Y}$ that maps any $X \in \mathcal{X}$ to its true class label $y \in \mathcal{Y}$.

Usually, the available set of labeled data is divided into a *training set* and a *test set*, where the former is used for learning a classifier and the latter to evaluate the performance of the classifier for classifying new unknown data. The *classification error*, that is, the proportion of misclassified samples in the test set, can be used directly as a structural measure for a classifier. Cross-validation is a more general technique for assessing the accuracy of a classifier [DHS00]. In cross-validation the data is divided into k folds (subsets) and in turns each fold is used as a test set while the other folds form the training set. When $k = n$ it is called the leave-one-out cross-validation and then the classification error can be calculated by

$$e(f, D) = \frac{1}{n} \sum_{i=1}^{n} I(f_{D \setminus D_i}(X_i) \neq y_i)$$

where $f_{D \setminus D_i}$ is the function learned by the classification algorithm by removing the i-th observation from the data and $I(\cdot)$ is the indicator function. Cross-validation can suffer from large variance [BG04]. Thus, for model selection, other validation approaches can produce better results [MIM10]. Next, we will briefly describe several algorithms for learning various types of classifiers. Classification algorithms are used extensively in Publication V.

The k-nearest neighbor

The k-nearest neighbor classifier is one of the simplest machine learning methods [FH51]. An item is classified to the majority class of its neighbors, that is, it is assigned to the most common class among its k-nearest neighbors in

the training set where the distance is measured in the feature space. In the case of $k = 1$, the item is assigned to the class of its nearest neighbor. Usually, Euclidean L_2 distance is used as the distance metric.

Decision tree

The decision tree classifier forms a classification tree where the classification process starts from the root and at each node the samples are divided into two children nodes according to some property of the data. Finally, each sample is assigned to the class of the leaf node where the sample ends up. The classical C4.5 algorithm [Qui93] builds the decision tree by choosing at each node one attribute of the data that most effectively splits the data in separate classes.

Naive Bayes

The naive Bayes classifier is a simple probabilistic classifier applying Bayes' rule with the assumption that the features are conditionally independent given the class [HY01]. That is, the value of any particular feature does not affect the probability of the value of any other feature. The classifier assigns each new sample x to the class y having the highest conditional probability by using the Bayes' rule and the independence assumption:

$$\Pr(y \mid x) = \frac{\Pr(y)\Pr(x \mid y)}{\Pr(x)} = \frac{\Pr(y)\Pr(x_1 \mid y)\cdots\Pr(x_d \mid y)}{\Pr(x)}.$$

Support vector machine

The support vector machine classifier (SVM) is a modern versatile method that finds the best separating hyper-plane between the samples belonging to two classes [SC08]. Usually, the data is first mapped nonlinearly into a high dimensional space where the classes are easier to separate linearly. In the case of multiple classes, the SVM classifier is repeatedly applied to separate all classes from each other.

2.2 Applying Randomization in Significance Testing

Randomization is a widely used method in statistics [Goo00, WY93, Edg95, Bes04, CB01]. In this section, we describe the basic approach behind the randomization tests, give the definition of empirical p-values, and present an example of the usefulness of preserving row and column sums in randomizing binary matrices.

2.2.1 Basic Approach

In the previous section, we described various data mining methods and defined some structural measures $\mathcal{S}$ for measuring the quality of the results returned by the algorithms. Recall that any measure can be used as long as smaller (or larger) values mean stronger presence of the measured structure. For example, in k-means clustering, the clustering error given in Equation (2.1) is a natural choice as a structural measure.

To assess whether a result $\mathcal{S}(D)$ is explained by certain characteristics of the original dataset, we generate randomized datasets $\widehat{D}$ sharing these predescribed statistics with the original dataset D, and compare the original structural measure $\mathcal{S}(D)$ against the distribution of structural measures $\mathcal{S}(\widehat{D})$. If the original structural measure $\mathcal{S}(D)$ clearly deviates from most of the structural measures on the randomized datasets, the result is considered not to be explained by the preserved statistics.

The main benefit of using randomization is that the user is relieved from the often difficult, and sometimes impossible, task of defining an analytical distribution for the test statistic. It is sometimes easier to devise a way of sampling from the null-hypothesis than to actually define it analytically. Also, integrating over the analytical distribution, which is needed for the p-value calculation, may not be straightforward.

In this thesis, we give methods for generating randomized datasets preserving specific statistics on various types of data. Thus, the general computational task we are addressing is the following (introduced already in Section 1.1).

Problem 1.1 (General randomization)**.** *Given a dataset D, generate a randomized dataset $\widehat{D}$ chosen independently and uniformly from the set of all datasets having the same values of the predescribed basic statistics as the original dataset D.*

To get a concrete idea, we will use throughout this chapter the randomization of binary matrices as an example of the approach. With binary tables, one commonly used significance testing approach is to randomize the dataset while preserving the row and column margins, that is, the row and column sums [CDHL05, GMMT07]. Thus, for example, a k-means clustering on binary dataset D is considered to be interesting if the clustering errors on randomized datasets having the same row and columns sums as the original dataset are larger than the original k-means clustering error. Thus, with 0–1 datasets the computational task is the following.

Problem 2.1 (Binary matrix randomization)**.** *Given an $n \times d$ binary dataset D, generate a randomized dataset $\widehat{D}$ chosen independently and uniformly from the set of $n \times d$ binary datasets having the same row and column margins as D.*

In Section 3.2.2, we give an algorithm originally introduced by Gionis *et al.* [GMMT07] for generating such randomized datasets. The randomization of binary tables serves as an introduction to the more complex randomization methods given in this thesis. It is also used as such as a part of various experiments.

2.2.2 Empirical p-Values

To measure how different the original result $\mathcal{S}(D)$ is from the results $\mathcal{S}(\widehat{D})$ on randomized datasets, we use empirical p-values [Goo00]. In the significance testing approach, the structural measure $\mathcal{S}(D)$ is used as a *test statistic*. Let $\widehat{\mathcal{D}} = \{\widehat{D}_1, \ldots, \widehat{D}_k\}$ be a set of independent randomized versions of the original dataset D. The one-tailed *empirical p-value* of the structural measure $\mathcal{S}(D)$ with the hypothesis of $\mathcal{S}(D)$ being small is

$$p_{\text{small}} = \frac{\left|\left\{\widehat{D} \in \widehat{\mathcal{D}} \mid \mathcal{S}(\widehat{D}) \leq \mathcal{S}(D)\right\}\right| + 1}{k+1}.$$

The empirical p-value captures the fraction of the randomized datasets that have a smaller or equal value of the structural measure than the original data. That is, these randomized datasets contain stronger presence of the measured structure than the original data. The addition of one in numerator and denominator guarantees a conservative estimate of the unknown true p-value, p_0, with slight bias [NCS02]. If the obtained empirical p-value is less than a given small threshold α, we can regard the result to be independent of the characteristics preserved in randomization. In this thesis, we will use the common value of $\alpha = 0.05$ for this threshold. Thus, if the original data contains stronger structure than 95% of the randomized datasets, the original result is considered to be significant and not to be explained by the preserved characteristics.

The one-tailed empirical p-value with the hypothesis of $\mathcal{S}(D)$ being large is defined similarly,

$$p_{\text{large}} = \frac{\left|\left\{\widehat{D} \in \widehat{\mathcal{D}} \mid \mathcal{S}(\widehat{D}) \geq \mathcal{S}(D)\right\}\right| + 1}{k+1}.$$

This definition is used when larger values of the structural measure correspond to stronger presence of the measured structure. If we are interested in the amount of the structure in either direction, we can use the two-tailed empirical p-value,

$$p_{\text{two-tailed}} = 2\min(p_{\text{small}}, p_{\text{large}}). \tag{2.2}$$

The standard deviation of the empirical p-value with k samples is

$$\sqrt{\frac{p_0(1-p_0)}{k}},$$

where p_0 is the unknown real p-value. This follows from the variance of binomial distribution. The upper bound of the standard deviation is

$$\frac{1}{2\sqrt{k}},$$

which is obtained when $p_0 = 0.5$.

Throughout this thesis, we will use either 999 or 99 randomized samples in calculating the empirical p-values. These correspond to upper bounds of 0.0158 and 0.0503 for the standard deviation of the empirical p-value, respectively. On the other hand, if the true p-value equals the threshold α, that is, $p_0 = \alpha = 0.05$, then the standard deviations are 0.007 and 0.022, respectively. Thus, the empirical p-values are fairly reliable with 999 and 99 randomized samples.

In practice, we will generate most of the collections $\widehat{\mathcal{D}} = \{\widehat{D}_1, \ldots, \widehat{D}_k\}$ of randomized versions of D by using Markov chains. In this approach, care has to be taken since the samples are not necessarily independent. We use the ideas by Besag and Clifford [BC89, BC91, Bes04] to guarantee the *exchangeability* of the samples $\widehat{D}_i$ by first running the chain backwards to some state $\widehat{D}_0$ and then k times separately forwards from state $\widehat{D}_0$; see Section 3.1.4 for more details.

An alternative way would be to use sequential probability ratio test [BC91, Wal45, FKH07], where randomizations of D are sampled until it is possible to accept or reject the null-hypothesis. However, we do not use it as we want to make the comparison of different randomization methods simple. With sequential probability ratio tests, often already 30 samples are enough for statistical inference with significance level $\alpha = 0.05$. This is related to the *power* of a statistical test, that is, the probability that the test will reject a false null hypothesis.

2.2.3 Example of Using Randomization

We give a simple toy example to demonstrate the concept of applying randomization for assessing the interestingness of data mining results. Consider the 9×8 binary dataset D given in Figure 2.2. The dataset seems to contain a fairly clear clustering structure with two clusters where the first four rows define the first cluster and the last five rows define the second cluster.

To analyze the clustering structure of the dataset D, we use the k-means clustering algorithm. As the structural measure $\mathcal{S}$, we use the corresponding k-means clustering error given in Equation (2.1) with two clusters, that is, $k = 2$. Thus, a strong clustering structure corresponds to a small clustering error. When the k-means clustering algorithm is applied to the original dataset,

1	1	1	0	0	0	1	1
1	0	1	0	0	1	0	0
1	1	1	0	0	0	0	1
1	1	1	0	1	1	0	1
0	0	0	1	0	0	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	1	1	0
0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	1

Dataset D

Figure 2.2: An example of a 0–1 dataset D.

we obtain an expected clustering where the first four rows form the first cluster and the last five rows form the second cluster. The corresponding structural measure, that is, the clustering error is $\mathcal{S}(D) = 9.2$.

Evaluating the quality of the clustering structure given only the original structural measure is troublesome. What does the number $\mathcal{S}(D) = 9.2$ tell us? It gives some numerical value for the amount of clustering structure, but without any further knowledge it is useless. To infer something about the clustering structure on the original dataset D, we have to compare the original result to clusterings on datasets that arguably do not contain any interesting clustering structure. This is the point where randomization tests come to the fore.

As discussed in Section 2.2.1, one possible approach in binary datasets is to compare the original structural measure to structural measures on datasets having the same number of ones in each row and in each column as the original binary dataset but being otherwise totally random. The idea is that these statistics define a basic background model for binary data. Any additional structure to the row and column margins is considered to be interesting. In Section 3.2.2, we will show how to generate such randomized datasets, that is, we give an algorithm for solving Problem 2.1.

To assess the clustering structure of the dataset D in Figure 2.2, we generated 999 randomized datasets having the same row and column margins as D. The same k-means clustering algorithm was applied to the randomized datasets, and for each randomized dataset $\widehat{D}$ the clustering error $\mathcal{S}(\widehat{D})$ was calculated. The average clustering error of the randomized datasets was 11.4 with the standard deviation being 0.7 and the minimum clustering error 8.6. There were in total 8 randomized datasets having a smaller or equal clustering error than the original dataset. This gives an empirical p-value of 0.009. Thus with

significance threshold level $\alpha = 0.05$, we can regard the original clustering structure to be independent of the row and column margins. Hence, the dataset D contains a clustering structure that is not explained by the number of ones in rows and columns.

2.2.4 Notes about Using the Approach

In the previous example, we saw how the value of the structural measure itself does not show whether the result is significant. Later in this thesis, we will give examples of datasets having the same numerical value of some structural measure but where the results have different significances when the structures of the background models of the datasets are taken into account. In randomization test, the background model is explicitly parametrized with the original data thus giving a useful null-model for analyzing the original result. Therefore, we do not have to make strong assumptions on how the data is distributed.

When randomization is used for significance testing, it is important to perform exactly the same data mining procedure on the randomized datasets as for the original dataset to guarantee the correctness of the results. Take the previous example where we wanted to assess the significance of the clustering structure where the first four rows formed the first cluster and the last five rows the second cluster. To obtain meaningful results, we need to use the same k-means clustering algorithm to find the best partitions to two clusters in the randomized datasets. Finally, the clustering errors are calculated by using these new clusterings on the randomized data and not, for example, by using the original clustering.

If the original pattern, for example, a clustering of the data, is chosen based purely on some background knowledge and not on the data in question, we have to calculate the structural measures for exactly the same pattern also in the randomized datasets. In this case, we can use the same clustering error as the structural measure but now the partition into the clusters is fixed for the randomized datasets.

2.2.5 Multiple Hypotheses Testing

While assessing the significance of local structures, we typically perform multiple tests. If we test multiple hypotheses at the same time, for example, the significances of various pairwise correlations in the data, it would be imprudent to use the same threshold value α for inferring the statistical significance of the results as it is used for determining the significance of one hypothesis. As the number of hypotheses increases, the probability of making Type I error, that is, incorrectly rejecting one null-hypothesis, also increases.

There exist various methods for controlling the error made in statistical inference in multiple hypotheses testing. The classical method by Bonferroni [Bon35] uses the transformed threshold value alpha $\alpha_0 = \alpha/N$, where N is the number of the tests. This approach controls the *familywise error rate* (FWER), that is, the probability of making one or more false discoveries. The Bonferroni criterion is often viewed [WY93] as quite conservative—the probability of making Type II error, that is, the error of failing to reject a null-hypothesis when it is not true, is high. The extended Holm-Bonferroni method [Hol79] alleviates this problem slightly.

Controlling the *false discovery rate* (FDR) is a less conservative procedure for correcting multiple comparisons. FDR measures the expected false positive rate, that is, the proportion of incorrectly rejected null-hypotheses. It is appropriate for selecting a list of rejected null-hypothesis, especially in exploratory data analysis. For example, the method by Benjamini-Hochberg [BH95] is a simple way to limit the FDR below the chosen threshold α. If the original p-values are $p_1, \ldots, p_N$ in increasing order, then the Benjamini-Hochberg method regards the results $1, \ldots, M$ as significant, where M is the largest index such that $p_M \leq \frac{M}{N}\alpha$.

The methods for correcting multiple hypotheses are compromises between making Type I and Type II errors. As the number of multiple comparisons is small in this thesis, to keep the experimental results simple and easily interpretable we usually do not use any correction for multiple comparisons. When applying the methods in large real problems with multiple hypotheses, any multiple hypotheses testing correction approach can be directly used to adjust the p-values obtained.

2.2.6 Handling Randomness in Computed Structural Measures

In calculating the empirical p-values, we have assumed that the structural measure $\mathcal{S}(D)$ is constant for a given dataset D. In practice, however, the computed values of the structural measures returned by many data mining algorithms vary from run to run due to the randomness applied in the algorithms. The reason for this is that many data mining problems are too hard to solve exactly. Thus we need to rely on approximate solutions, where different runs may lead to different results. For example, the k-means clustering algorithm starts from an initial random clustering and improves it until convergence. So the problem is, how can we ensure that the calculated empirical p-values are stable to such variance?

In many cases, the variation is so small that there is no need to apply any corrections. However, if the amount of variation is comparably large, the effect of the variation should be taken into account. The variation of the original data

mining result $\mathcal{S}(D)$ can especially have a large impact to the final empirical p-value. We propose a couple of different solutions to this problem having different time complexities and effects on the resulting accuracy.

A standard approach to decrease the variability of the results is to repeat the same data mining method a reasonable number of times with different random initial seeds and keep only the best solution. The problem with this approach when applied to the randomization tests is that we have to repeat the same data mining method with the same number of random starts also for producing each randomized dataset. However, if the methods are fast and the datasets small enough to do this in acceptable time, increasing the accuracy is a good solution.

Since we cannot arbitrarily increase the accuracy of the structural measures $\mathcal{S}(\widehat{D})$ on all randomized datasets without the calculation time growing too large, we can concentrate on decreasing the effect of the variability of the original structural measure $\mathcal{S}(D)$ as it has the most impact to the empirical p-value. A solution to this is to repeat the data mining method (that may itself also contain repetition for obtaining a better result) on the original dataset a few times, and then use the median value of these as the final structural measure for the original data. Another solution is to calculate an empirical p-value for all of these results and report the mean value of these as the final empirical p-value. Note that in these approaches, it suffices to apply the original data mining method only once on each randomized dataset.

The latter solution is closely related to calculating the test statistic U of the Wilcoxon-Mann-Whitney two-sample rank-sum test [Goo00]. The proposed solution has the same good properties as the U statistic as well as it generalizes the concept of empirical p-value to instable results.

In the experiments, we will use the approaches explained above to decrease the variability. However, it turns out that in the experiments of this thesis the stability issue is not vital for the final results; the reported empirical p-values are stable in practice.

CHAPTER 3

Generating Randomized Datasets

In this chapter, we discuss different approaches for generating randomized datasets. First, we describe general approaches and give a brief background to Markov chain Monte Carlo (MCMC) methods. Next, we discuss the specific randomizing tasks and approaches for different types of data that are studied in the publications of this thesis. Finally, we briefly discuss other related randomization methods.

3.1 General Randomization Approaches

There are three types of general approaches for generating randomized datasets. They can be summarized as follows:

1. Direct sampling.
2. Performing local modifications with hard constraints.
3. Performing local modifications with soft constraints.

The approaches are in increasing order of generality and complexity. The first approach is the simplest but can be used only in few cases. Direct sampling means that we can produce random datasets just by drawing values from some distribution or by permuting some values. In this approach, the random datasets are also guaranteed to be independent of each other. Especially, all the methods introduced in Publication V fall in this category. However, most of the other methods given in this thesis use the last two approaches.

The last two approaches are based on MCMC methods. The idea is to iteratively perform small changes on the original data while preserving the

given statistics. Sufficiently many local modifications have to be performed to guarantee that the output is practically independent from the original data. The difference between the two approaches is in the handling of the statistics preserved in randomization. In the second approach, we perform local modifications that preserve the given statistics exactly while randomizing the data otherwise. However, often such local modifications do not exist. Then, we need to use the third approach where the given statistics are preserved approximately.

3.1.1 Markov Chains

In this section, we provide a brief background on the theory and notation of Markov chains. A Markov chain is a discrete-time stochastic process where the future depends only on the present state [RC04, GCSR03, AdFDJ03, Bes04]. Formally, a sequence of random variables $X_0, X_1, \ldots$ is called a *Markov chain* if it satisfies the *Markov property*

$$\Pr(X_{t+1} = x \mid X_t = x_t, \ldots, X_0 = x_0) = \Pr(X_{t+1} = x \mid X_t = x_t).$$

In this thesis, we study only *time-homogeneous* Markov chains satisfying

$$\Pr(X_{t+1} = x \mid X_t = y) = \Pr(X_t = x \mid X_{t-1} = y) \quad \text{for all } t.$$

In case of a finite state space $\mathcal{X}$, we use a *transition probability matrix* P that contains the probabilities P_{ij} of moving from state $i \in \mathcal{X}$ to state $j \in \mathcal{X}$:

$$P_{ij} = \Pr(X_{t+1} = j \mid X_t = i).$$

A Markov chain is said to be *irreducible* (or *connected*) if all states are reachable in finite number of steps from any other state. The chain is *aperiodic* (or *acyclic*) if returns to any state can occur at irregular times. A probability vector π is a *stationary distribution* of a time-homogeneous Markov chain if it satisfies

$$\pi = P^T \pi.$$

If the chain is irreducible and aperiodic, it converges to a unique stationary distribution regardless of the starting state. Informally, the *mixing time* of a Markov chain is the number of steps needed for approximate convergence to the stationary distribution. See Levin [LPW06] for a formal definition.

An important special case are time-reversible Markov chains for which we cannot identify whether the chain is running forwards or backwards. A Markov chain is *time-reversible* if there exists a probability distribution π satisfying the *detailed balance* condition:

$$\pi_i P_{ij} = \pi_j P_{ji}. \tag{3.1}$$

Summing over all states $i \in \mathcal{X}$ gives

$$\sum_{i \in \mathcal{X}} \pi_i P_{ij} = \pi_j \sum_{i \in \mathcal{X}} P_{ji} = \pi_j.$$

Thus, for a time-reversible Markov chain, a distribution π satisfying the detailed balance condition is always a stationary distribution. This leads directly to the following theorem.

Theorem 3.1. *The stationary distribution of an irreducible, aperiodic Markov chain with symmetric transition probabilities, $P^T = P$, is uniform.*

Proof. Let π be a uniform distribution. Then, $\pi_i = \pi_j$ for all i and j. Because P is symmetric, also $P_{ij} = P_{ji}$. Thus π satisfies the detailed balance condition, $\pi_i P_{ij} = \pi_j P_{ji}$, and therefore, π is the stationary distribution of the chain. □

3.1.2 Markov Chain Monte Carlo Methods

In general, Markov chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from probability distributions [RC04, GCSR03, AdFDJ03, Bes04]. They are based on constructing a Markov chain that has the desired probability distribution as its stationary distribution. The state of the chain after a large number of steps is then used as a random sample from the desired distribution.

The MCMC methods introduced in this thesis are based on local modifications. Each step in a Markov chain is a local modification, that is, a small change of the current state. Each state corresponds to a partly randomized version of the original data. The algorithms start from the original dataset D. Given a current dataset $\widehat{D}_t$, the next step selects at random some local modification from the collection of allowed operations, and applies it to $\widehat{D}_t$. If the change is accepted, this yields $\widehat{D}_{t+1}$. Otherwise $\widehat{D}_{t+1} = \widehat{D}_t$.

In this thesis, we use the original dataset as the starting state as it is in general the only dataset that we know to have the correct values in the statistics to be preserved. Optimally, we could select the starting state from the stationary distribution. However, in this case, there would be no need for MCMC methods as we could directly obtain the random samples. Recall that the mixing time describes the number of steps after which the state distribution of the Markov chain has approximately converged to the stationary distribution. Only samples obtained after the mixing time of the chain should be accepted as random samples from the stationary distribution.

The mixing time is usually very hard to evaluate theoretically. In principle, the asymptotic rate of convergence is determined by the second largest eigenvalue in magnitude of the transition probability matrix P [BDX04, MT06,

LPW06]. However, this is feasible only for relatively small state spaces. In this thesis, all the Markov chains we consider have a very large number of states. Thus, in practice, we use some distance measure to assess the convergence. That is, when the distance between the starting state and the current state has converged, we can assume that the distribution has converged [CC96]. Instead of studying some distance measure, we could control that the value distribution of the given structural measure has converged.

3.1.3 Metropolis Algorithm

The Metropolis algorithm [MRR+53] is a general MCMC method for obtaining samples from a probability distribution π. Let $\mathcal{X}$ denote the set on which π is defined, and let $Q(y \mid x)$ be a symmetric proposal distribution on $\mathcal{X}$, that is, $Q(y \mid x) \geq 0$, $\sum_y Q(y \mid x) = 1$, and $Q(y \mid x) = Q(x \mid y)$ for all $x, y \in \mathcal{X}$. At each step, the Metropolis algorithm samples a state y as a proposal for the next state with probability $Q(y \mid x)$ given the current state x. The proposal state y is accepted as the new state with probability

$$\min(1, \pi_y / \pi_x). \tag{3.2}$$

Otherwise, the current state x becomes the new state. Using the method assumes that sampling from the proposal density $Q(y \mid x)$ is easy and that the ratio π_y / π_x can be computed. This holds if the probability π_x can be calculated up to a constant factor. The next theorem follows from the detailed balance condition (3.1) of time-reversible Markov chains [GCSR03].

Theorem 3.2. *The Markov chain M produced by the Metropolis algorithm is time-reversible. Let π be the distribution used to calculate the acceptance probabilities in Equation* (3.2) *for M. If M is irreducible and aperiodic, then π is the stationary distribution of M.*

If the proposal distribution $Q(y \mid x)$ is not symmetric, we can use the extended Metropolis-Hastings algorithm [Has70] where the proposal state y is accepted with probability

$$\min\left(1, \frac{\pi_y Q(x \mid y)}{\pi_x Q(y \mid x)}\right).$$

The proposal distribution Q has a huge impact on the mixing time. It should be as global as possible while allowing a high acceptance rate. The optimal acceptance rate under some general normality assumptions is around 25% [GCSR03]. In this thesis, we use the Metropolis algorithm in such cases where we need soft constraints, that is, where there do not exist local modifications that preserve the given statistics exactly while randomizing the data.

3.1.4 Obtaining Exchangeable Set of Samples

The samples produced by MCMC methods are generally not independent, unless the number of steps taken between the samples is at least the mixing time. It is very hard to estimate this quantity in any application. Furthermore, using dependent samples in calculating an empirical p-value breaks the validity of the result. To solve this problem, we will use the approach by Besag and Clifford [BC89] to produce an exchangeable set of samples that guarantees the validity of the empirical p-value. A set of samples is *exchangeable* if the samples are produced from the same distribution when the null-hypothesis is true, meaning that we cannot distinguish the samples from each other.

In the approach by Besag and Clifford, the chain is started from the original dataset D and run backwards for I steps to produce a new starting state $\widehat{D}_0$. Then for the desired number k of samples, we start for each $i = 1, \ldots, k$ a new chain from $\widehat{D}_0$ and run the chain I steps forwards, obtaining sample $\widehat{D}_i$. Then the set $\{D, \widehat{D}_1, \ldots, \widehat{D}_k\}$ forms an exchangeable set of samples. Thus, each of the samples $D, \widehat{D}_1, \ldots, \widehat{D}_k$ can be produced by I steps from $\widehat{D}_0$, including the original dataset D. Therefore, the rank of $\mathcal{S}(D)$ is uniform among the values $\{\mathcal{S}(D), \mathcal{S}(\widehat{D}_1), \ldots, \mathcal{S}(\widehat{D}_k)\}$ under the null-hypothesis, implying the validity of the empirical p-value regardless of irreducibility and convergence of the chain. If the chain has not converged fully, we just obtain more conservative p-values [Bes04, BC89].

All the methods studied in this thesis are time-reversible, that is, running the chain backwards is the same as running the chain forwards. In this thesis, we will always apply the Besag-Clifford approach when we use MCMC methods for producing the randomized samples. Although it guarantees the validity of the p-value, good convergence of the chain increases the power of the randomization test.

3.2 Specific Randomization Approaches

Next we discuss the different randomization approaches used in the publications of this thesis. Additionally, we briefly discuss other randomization methods proposed in the literature.

3.2.1 Permuting Labeled Data

We start by introducing the randomization methods used in Publication V as they are based on the easiest general technique, direct sampling. In Publication V we study how to assess the performance of a classifier on labeled

data. Recall that classification as a data mining technique was discussed in Section 2.1.5.

Assume that we have trained a classifier f on labeled data $D = \{(X_i, y_i)\}_{i=1}^{n}$. In Publication V we study two statistical tests that supply answers for the following questions:

Test 1: Has the classifier found a significant class structure, that is, a real connection between the data and the class labels?

Test 2: Is the classifier exploiting dependencies between the features to increase the accuracy of the classification?

The first test is a standard test used in statistics [Goo00] whereas the second test is new. Note, that these two tests study whether the classifier is using the described properties and not whether the plain data contain such properties. For studying the characteristics of a population represented by the data, other statistical test could be used [CB01].

Producing the corresponding randomized datasets with both tests is relatively easy and can be obtained by direct sampling. With Test 1 we break the connection between the data and the class labels by permuting the labels. Repeating the classification on such permuted datasets and comparing the obtained classification accuracies to the original accuracy tells whether the classifier has found a real class structure. The test has been used for assessing the classifier accuracy [GF03, GLMP05, HAD03], for validating the model [MIM10, BND04, MSP05], as well as for selecting the features [FW98, Jen92].

With Test 2 the randomization is more sophisticated but still possible to do by direct sampling. The corresponding null-hypothesis assumes that the columns in X are mutually independent given the class, that is,

$$p(X^1, \ldots, X^m \mid y) = p(X^1 \mid y) \cdots p(X^m \mid y).$$

Randomized samples are obtained by independently permuting the values in each column inside each class. Such randomization removes the dependencies between the features given the class. Repeating the classification on such data will reveal whether the dependencies between the features are used in explaining the class structure by the chosen classification algorithm. The inference results of this tests can help the user to understand the properties of the classifier and to improve the classification accuracy. In Figure 3.1, an example of labeled data and corresponding randomized versions produced by Test 1 and Test 2 are shown.

0.96	0.75	0.89	+
0.34	0.26	0.96	+
0.59	0.51	0.55	+
0.22	0.70	0.14	+
-0.85	-0.19	-0.80	−
-0.74	-0.76	-0.75	−
-0.16	-0.07	-0.38	−
-0.75	-0.65	-0.53	−

(a) Original data

0.96	0.75	0.89	−
0.34	0.26	0.96	+
0.59	0.51	0.55	−
0.22	0.70	0.14	+
-0.85	-0.19	-0.80	+
-0.74	-0.76	-0.75	−
-0.16	-0.07	-0.38	−
-0.75	-0.65	-0.53	+

(b) Test 1

0.22	0.70	0.55	+
0.59	0.75	0.96	+
0.96	0.51	0.14	+
0.34	0.26	0.89	+
-0.85	-0.76	-0.75	−
-0.74	-0.07	-0.38	−
-0.16	-0.65	-0.53	−
-0.75	-0.19	-0.80	−

(c) Test 2

Figure 3.1: An example of labeled data and corresponding randomized versions: (a) the original data with two classes, + and −; (b) one randomized sample produced by Test 1; (c) one randomized sample produced by Test 2.

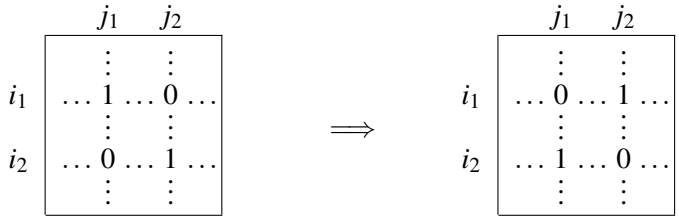


Figure 3.2: A swap in a 0–1 matrix. The four elements shown are rotated and the rest of the matrix is kept fixed. The number of ones in each row and column does not change.

3.2.2 Randomizing Binary Matrices

We use an MCMC method, see, e.g., Cobb *et al.* and Gionis *et al.* [CC03, GMMT07], for producing random binary matrices with given row and columns margins, that is, a method for solving Problem 2.1. This method shares the basic ideas with the new methods in Publications I and II for randomizing more general types of matrices. The method is based on iteratively performing *swaps* as shown in Figure 3.2. In a swap, a pair of ones is changed with a pair of zeros preserving the row and column sums. A randomized dataset $\widehat{D}$ is produced by starting from the original matrix D and performing I attempts of swaps. The method is presented in Algorithm 3.1. In Figure 3.3 is given an example of binary data and its randomized version. The original data is the same as in Section 2.2.3.

In the experiments by Gionis *et al.* [GMMT07], the authors found that using five times the number of ones in the matrix as the number of attempts

Algorithm 3.1 SwapBinary

Input: Dataset $D \in \{0,1\}^{n \times d}$, number of attempts I
Output: Randomized dataset $\widehat{D}$ preserving row and column margins of D

```
1: D̂ ← D
2: for i ← 1 to I do
3:     Pick i1, i2 and j1, j2 randomly such that D̂_{i1 j1} = 1 and D̂_{i2 j2} = 1
4:     if D̂_{i1 j2} = 0 and D̂_{i2 j1} = 0 then
5:         D̂ ← swapped version of D̂
6:     end if
7: end for
8: return D̂
```

1	1	1	0	0	0	1	1
1	0	1	0	0	1	0	0
1	1	1	0	0	0	0	1
1	1	1	0	1	1	0	1
0	0	0	1	0	0	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	1	1	0
0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	1

(a) Original data

0	0	1	1	1	1	1	0
0	0	1	0	1	0	0	1
0	1	0	0	1	1	0	1
1	1	1	1	1	0	0	1
1	0	0	0	0	0	0	0
0	1	1	1	0	1	0	0
1	0	1	1	0	0	0	1
1	0	1	0	0	0	0	0
0	0	1	1	0	1	1	0

(b) Swapped data

Figure 3.3: An example of a 0–1 data and its randomized version.

I is usually sufficient to guarantee convergence. To increase the acceptance rate, the first two corners, $\widehat{D}_{i_1 j_1}$ and $\widehat{D}_{i_2 j_2}$, are required to contain ones (and not zeros) since binary matrices are usually sparse.

Algorithm 3.1, *SwapBinary*, samples uniformly from the set of all the binary matrices with the same margins as the original matrix. This follows from Theorem 3.1: The chain is aperiodic, since it can stay in the current state. The chain is irreducible as proved by Ryser [Rys57]. Finally, the transition probabilities are symmetric as the probability of a reversing swap equals the probability of the original swap.

The idea of swapping matrix elements as a randomization technique has a long history, see Cobb *et al.* [CC03]. There are also various other methods for randomizing binary matrices and contingency tables [CDHL05, CC03, BBV06, Dye03, DG95].

3.2.3 Randomizing Real-Valued Matrices

Defining an appropriate randomization approach for binary matrices was relatively straightforward. The row and column margins in binary matrices, that is, the number of ones in each row and column, describe the basic properties of the data. Thus preserving them in randomization allows us to find patterns that are not directly explained by the row and column sums. In Publication I we generalize this approach to real-valued matrices where the rows are of similar type as well as the columns. We consider two randomization problems where we fix the statistics of the rows and columns in different degrees:

Problem 3.1 (Real-valued matrix randomization: preserving value distributions)**.** *Given an $n \times d$ real-valued matrix D, generate a matrix $\widehat{D}$ chosen independently and uniformly from the set of $n \times d$ real-valued matrices having approximately the same value distributions in rows and columns as D.*

Problem 3.2 (Real-valued matrix randomization: preserving means and variances)**.** *Given an $n \times d$ real-valued matrix D, generate a matrix $\widehat{D}$ chosen independently and uniformly from the set of $n \times d$ real-valued matrices having approximately the same row and column means and variances as D.*

In these randomization approaches the row and column value distributions or the row and column means and variances are thought to describe the basic properties of the underlying phenomenon. One notable difference in these problem statements compared to Problem 2.1 for binary matrices is that the statistics are required to be preserved *approximately*. We will specify later what it means exactly. There are two reasons for using approximations. Firstly, there is no easy way to produce exact randomized samples. Secondly, preserving the row and column value distributions exactly is sometimes too strict, as seen by the following theorem.

Theorem 3.3. *Let $D \in \mathbb{R}^{n \times d}$ be a real-valued matrix with unique values, that is, no value is present twice. Then, D is the only matrix $\widehat{D}$ in $\mathbb{R}^{n \times d}$ having exactly the same row and column value distributions as D.*

Proof. Let $\widehat{D} \in \mathbb{R}^{n \times d}$ have exactly the same values in each row and column as the matrix D, that is, their row and column value distributions are equal. Because all the values are unique, the value D_{ij} appears exactly once in row i and once in column j of $\widehat{D}$. This is only possible if $\widehat{D}_{ij} = D_{ij}$ because otherwise $\widehat{D}$ would contain the value D_{ij} twice which is a contradiction. Thus, $\widehat{D} = D$. □

Thus, a matrix with unique values is the only matrix having the corresponding row and column value distributions exactly. By introducing uncertainty to the values we overcome the problem of uniqueness and can produce

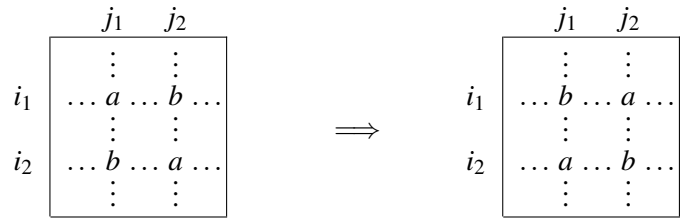


Figure 3.4: A swap in a discretized matrix. The four elements shown are rotated and rest of the matrix is kept fixed. The discretized values in each row and column do not change in a swap.

0.90	0.22	0.93	0.49	0.10	0.72
0.98	0.12	0.73	0.62	0.26	0.90
0.44	0.30	0.49	0.68	0.34	0.89
0.11	0.32	0.58	0.40	0.68	0.33
0.26	0.42	0.24	0.37	0.14	0.70
0.41	0.51	0.46	0.99	0.72	0.20
0.59	0.09	0.96	0.04	0.11	0.03
0.26	0.26	0.55	0.89	0.65	0.74
0.60	0.80	0.52	0.91	0.49	0.50
0.71	0.03	0.23	0.80	0.78	0.48

(a) Original data

0.34	0.03	0.80	0.89	0.72	0.14
0.98	0.22	0.49	0.93	0.68	0.59
0.99	0.32	0.50	0.44	0.33	0.26
0.72	0.48	0.65	0.46	0.09	0.49
0.52	0.04	0.26	0.11	0.30	0.37
0.26	0.23	0.55	0.60	0.41	0.78
0.10	0.12	0.24	0.71	0.03	0.89
0.62	0.58	0.42	0.49	0.51	0.90
0.26	0.68	0.91	0.73	0.80	0.74
0.40	0.20	0.96	0.90	0.11	0.70

(b) Randomized data

Figure 3.5: An example of a real-valued dataset and its randomized version with *SwapDiscretized*.

meaningful randomizations. By allowing more uncertainty, the methods are able to produce matrices with more randomness.

In Publication I we give two algorithms, *SwapDiscretized* and *GeneralMetropolis*, for solving Problems 3.1 and 3.2. The *SwapDiscretized* algorithm generalizes the 0–1 swap of binary matrices shown in Figure 3.2 to real-valued matrices. The values are first replaced by N equally-spaced discrete values. Then we perform swaps that preserve the discretized value distributions exactly, as shown in Figure 3.4. Finally, the original values are returned back. In Figure 3.5 an example of real-valued data and its randomized version with *SwapDiscretized* is given. In Publication I we also introduce a more general version of *SwapDiscretized* that can support different discretizations for rows and columns. Note that the larger the data the better the algorithm can preserve the row and column value distributions.

The *GeneralMetropolis* algorithm uses the Metropolis method described in Section 3.1.3 to produce samples that approximately preserve the chosen row and column statistics. We first define a difference measure $E(\widehat{D}, D)$ that gives the total difference between the row and column statistics of the original matrix D and a randomized matrix $\widehat{D}$. As $E(\widehat{D}, D)$ we use the sum of the differences in each row and column. The difference between D and $\widehat{D}$ in one row or column is defined either as the L_1 distance between the cumulative distribution functions or as the sum of differences in means and variances. The sampling distribution for randomized datasets $\widehat{D}$ is defined as

$$\Pr(\widehat{D} \mid D) = c \exp\{-wE(\widehat{D}, D)\}, \tag{3.3}$$

where w is an error scaling constant and c is a normalizing constant. Matrices with smaller error have a higher probability to be produced. As there are more matrices with larger error, the peak of the distribution of the error of randomized matrices produced is not on zero. The randomized matrices are sampled from (3.3) by using Metropolis algorithm with five different types of local modifications that either modify one or four elements at a time. The parameter selection especially for the *GeneralMetropolis* method was found to be problematic in Publication I.

3.2.4 Randomizing General Matrices

The randomization approach presented in Publication I is theoretically sound but it suffers from various practical shortcomings. In Publication II we give solutions to these problems and introduce a new algorithm for randomizing different types of matrices. The first improvement is support for numerical matrices with dissimilar features, that is, features measured using different scales.

Problem 3.3 (Numerical matrix randomization with dissimilar features)**.** *Given an $n \times d$ real-valued matrix D where the features (columns) are of different types, generate a matrix $\widehat{D}$ chosen independently and uniformly from the set of $n \times d$ real-valued matrices having approximately the same values in each column as D and approximately the same column-wise ranks in each row as D.*

The *column-wise rank* of a value is its ordinal number among the values in the same column. Thus, the randomization preserves the distribution of extreme values in each row when the features have equal importance. This approach can be used also if the features are measured using the same scale but the value distributions differ from each other substantially, causing severe problems with the previous approaches.

The second improvement is a new algorithm, *SwapConstrained*. It generalizes the *SwapDiscretized* algorithm by allowing different tolerance ranges for each value. That is, we allow an original value D_{ij} to be replaced in row i by a value in a tolerance range R_i^j, for example, $R_i^j = [D_{ij} - \varepsilon, D_{ij} + \varepsilon]$, and in column j by a value in a tolerance range C_j^i. A randomized matrix $\widehat{D}$ satisfies these constraints if for each row i we can reorder the original row constraints $R_i^1, \ldots, R_i^d$ in an order $\widehat{R}_i^1, \ldots, \widehat{R}_i^d$ such that

$$\widehat{D}_{i1} \in \widehat{R}_i^1, \ldots, \widehat{D}_{id} \in \widehat{R}_i^d,$$

and similarly for column constraints. Thus, the original matrix satisfies $D_{ij} \in R_i^j \cap C_j^i$ whereas a randomized matrix satisfies $\widehat{D}_{ij} \in \widehat{R}_i^j \cap \widehat{C}_j^i$ for some legally reordered row and column constraints $\widehat{R}_i^j$ and $\widehat{C}_j^i$.

In the *SwapConstrained* algorithm we do similar swaps as in Figure 3.4, but we require that each swap satisfies the new unique row and column constraints. Surprisingly, this can be done almost as fast as with *SwapDiscretized*. To overcome the problem of value distributions having long tails and narrow peaks, we use tolerance ranges where each element has about $1/\sqrt{n}$ of all elements in its column tolerance range and about $1/\sqrt{d}$ of all elements in its row tolerance range. These tolerance ranges are theoretically justified by the properties of Kolmogorov-Smirnov test [Mas51, Con98] guaranteeing that the row and column value distributions will be preserved with good statistical accuracy while allowing as random matrices as possible. By using different constraints we can solve both Problems 3.1 and 3.3. Furthermore, the new algorithm can be easily modified to support sparse matrices and missing values. The theoretical analysis in Publication II gives that the mixing time of *SwapConstrained* on full matrices is under certain assumptions $O(min(\sqrt{n}, \sqrt{d})nd\log(nd))$.

The improved approach is now theoretically justified and produces good randomizations without any manual tuning. Furthermore, the new *SwapConstrained* method reduces on binary matrices to exactly the same method as described in Section 3.2.2. The support for sparse matrices, missing values and any value distributions means that the approach can be used to randomize practically any types of matrices.

In Figure 3.6, an example of real-valued data with dissimilar features is given with randomization by *SwapConstrained*. This is the same dataset as in Figure 1.1 of introduction. Recall that the rows correspond to retail stores and the columns to products. As the original stores have very strict order between their general price levels, the randomized dataset is also fairly close to original dataset as the distributions of product-wise price-ranks are preserved in each store. That is, the general price levels of the retail stores explain most of the

0.69	2.49	0.99	5.49	6.49	5.99
0.79	2.79	1.19	6.69	7.13	6.99
0.79	2.49	1.29	6.39	7.59	6.49
0.89	2.89	0.99	6.59	6.99	7.49
0.89	3.19	1.49	7.09	7.39	11.69
0.99	3.59	1.79	8.09	8.69	9.59
0.99	3.29	1.69	6.89	9.19	12.99
1.19	4.59	1.99	8.49	8.59	16.99
1.19	4.29	2.49	8.99	9.39	18.99
1.29	3.99	2.19	7.79	9.99	14.49

(a) Original data

0.79	2.49	0.99	6.39	6.49	6.99
0.69	3.19	1.19	6.39	6.99	6.99
0.69	2.49	1.19	6.69	7.59	6.49
0.89	2.49	1.19	6.89	7.39	9.59
0.89	3.59	1.49	7.79	7.39	6.99
0.99	3.19	1.79	7.09	8.69	11.69
1.29	3.29	1.79	7.09	7.39	11.69
1.19	4.59	2.19	7.79	9.39	16.99
1.19	4.59	1.99	8.99	9.39	14.49
1.19	3.99	2.49	8.49	9.39	18.99

(b) Randomized data

Figure 3.6: An example of a real-valued dataset with dissimilar features (columns) and its randomized version with *SwapConstrained*. This is the same original dataset as in Figure 1.1.

data. The correlation between the first two columns, milk and bread, in the randomized dataset is 0.9466 while the original correlation is 0.9323.

3.2.5 Randomizing Multi-Relational Databases

Randomization in multi-relational databases is by no means unambiguous. In Publication IV we discuss different approaches for producing meaningful randomizations on databases consisting of multiple binary relations. Binary relations can be presented in different ways that help us to understand the different randomization approaches. The same multi-relational database with three binary relations is shown in Figure 3.7 as standard relations, in Figure 3.8 as binary matrices and in Figure 3.9 as bipartite graphs. Especially, the latter two help us in forming the randomizations.

As seen in Figure 3.8 the binary relations can be presented as binary matrices. This allows us to directly use the swap randomization for binary matrices described in Section 3.2.2. By swap randomizing one relation we can find out what is its structural impact to the final result of the query. The binary swap is easy to understand also in the bipartite graph presentation where it corresponds to swapping two edges as shown in Figure 3.10. In Publication IV a swap randomized version of a binary relation AB is noted as $\mathsf{sw}(AB)$.

The other randomization studied in Publication IV is the label permutation of a relation AB that randomly permutes the labels assigned to the attribute A or B, as shown in Figure 3.11. In Publication IV this operation was called as

GM	
Genre	Movie
Romance	m_1
Romance	m_2
Drama	m_3
Drama	m_4
Drama	m_5
Drama	m_6
Drama	m_7
History	m_6
History	m_7

(a) Genre × Movie

MD	
Movie	Director
m_1	C. Waitt
m_2	C. Waitt
m_3	C. Waitt
m_4	C. Waitt
m_5	C. Waitt
m_6	T. George
m_7	T. George

(b) Movie × Director

DA	
Director	Age
C. Waitt	30
T. George	60

(c) Director × Age

Figure 3.7: A toy example of a multi-relational database with three binary relations: movies classified by genres, GM; movies directed by directors, MD; and ages of directors, DA.

GM							
	m_1	m_2	m_3	m_4	m_5	m_6	m_7
Romance	1	1	0	0	0	0	0
Drama	0	0	1	1	1	1	1
History	0	0	0	0	0	1	1

(a) Genre × Movie

MD		
	C. Waitt	T. George
m_1	1	0
m_2	1	0
m_3	1	0
m_4	1	0
m_5	1	0
m_6	0	1
m_7	0	1

(b) Movie × Director

DA		
	30	60
C. Waitt	1	0
T. George	0	1

(c) Director × Age

Figure 3.8: The binary matrix representation of the toy database in Figure 3.7.

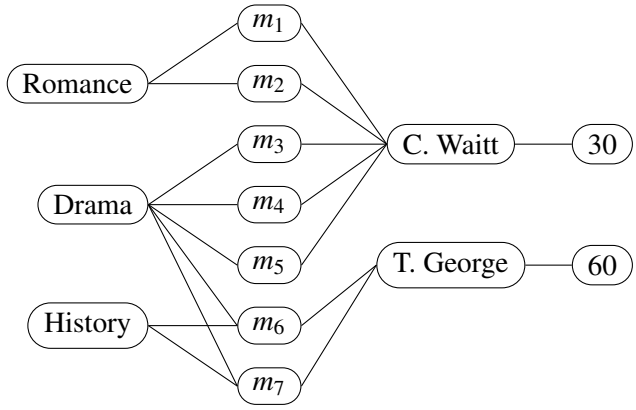


Figure 3.9: The multi-layer bipartite graph representation of the movie database shown in Figure 3.8. The graph shows all the possible paths from the source nodes, Genre, to the destination nodes, Age.

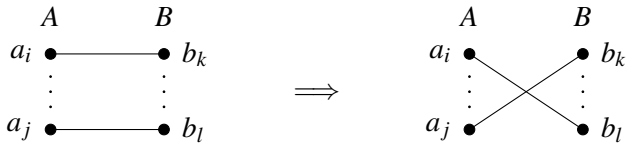


Figure 3.10: A binary swap in a bipartite graph.

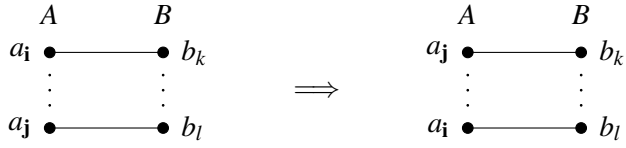
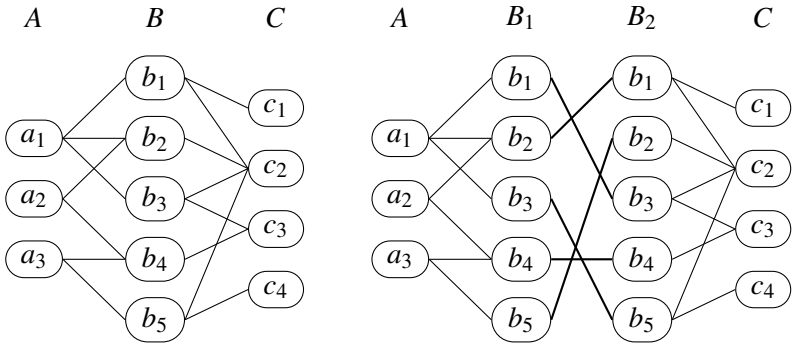


Figure 3.11: Label permutation of attribute A.

the row-order or column-order permutation, referring to the binary matrix representation. This label permutation breaks the connection between two binary-relations sharing one attribute. Permuting the labels of attribute B in relation AB is actually the same as adding an identity relation $I_B = B_1B_2$ and swap randomizing this relation, as shown in Figure 3.12.

Thus, in a multi-relational database we have two different types of randomizations: swap randomizing one relation or permuting the connection between two relations. Sometimes these approaches can produce equal set of randomized databases, as discussed in Publication IV. For example, if relation AB has



(a) Two relations AB and BC. (b) Label permutation of attribute B.

Figure 3.12: Label permutation of attribute B in relation AB: (a) original database with two relations AB and BC; (b) the label permutation equals swap randomizing an added identity relation $I_B = B_1B_2$.

exactly one 1 in each column, then the swap randomization of AB equals the label permutation of B.

3.2.6 Constraining Randomization with Previous Results

In Publication III we consider the problem of iterative data mining. In practice, many different methods are applied to the same dataset to find the patterns that describe the data the best. However, this can lead to problems. Firstly, the multiple hypotheses correction should be applied as discussed in Section 2.2.5. Secondly, some of these patterns can be dependent observations on each other. That is, if we have found a good clustering structure in the data and a few nice association rules in the data, how can we be sure that they are independent phenomena?

The approach suggested in Publication III to solve this problem is simple in principle but often complex in practice. The idea is to start with the original data and apply some interesting data mining method. Then the significances of the found patterns are assessed, for example, by using the randomization of binary or real-valued matrices. Some of the assessed patterns are then discovered to be significant. The next step is to find new patterns that are independent of these previous significant patterns. This is done by applying some new data mining method and assessing the found new patterns by randomization method where the previously found significant patterns are preserved in randomization in addition to some other basic statistics like the row and column value distributions. This iterative process can be continued until no new sig-

nificant patterns are found or until a sufficient number of patterns is obtained. In the end we are left with a set of mutually independent significant patterns.

The hard part of the approach is to produce all of these randomizations where we preserve the previous significant results. In practice, we have to develop a new randomization method for each specific case. The generic approach consists of making a difference measure $E(\widehat{D}, D)$ that measures the combined difference in the preserved statistics between a randomized dataset $\widehat{D}$ and the original dataset D. The problem is then the following:

Problem 3.4 (Soft randomization)**.** *Given a dataset D, a difference measure $E(\widehat{D}, D)$ for some statistics and a scaling constant $w > 0$, generate a dataset $\widehat{D}$ chosen with a probability*

$$\Pr(\widehat{D} \mid D) \propto \exp\{-wE(\widehat{D}, D)\} \tag{3.4}$$

from all datasets having the same size as D.

Note that this is exactly the same equation as the one used in Section 3.2.3 to generate randomized real-valued matrices; the only change is in the difference measure $E(\widehat{D}, D)$. If $w = \infty$, the statistics are preserved exactly. The Metropolis algorithm described in Section 3.1.3 can be used to sample datasets from the probability distribution (3.4). The problem is that when many patterns are added as constraints, the randomization becomes very hard to do in practice. One reason for this is that sometimes only few patterns are needed to explain the other patterns. But the other more common reason is that this randomization approach does not work with too many constraints as either we produce too large error in the preserved statistics or the method is not able to move away from the original data. To alleviate this problem we can use the parallel tempering [Gey91] but still the running time can be too large in practice.

Despite the general hardness of the problem, some specific cases are easy. For example, in Publication III the approach is applied to study the clustering structure and frequent itemsets on binary datasets. Preserving a clustering structure turns out to be easy. Randomized binary datasets preserving the clustering structure are produced by swap-randomizing each cluster separately. This randomization, *ClusterSwap*, preserves the cluster centers and variances exactly. This approach could be generalized also to real-valued matrices. For preserving the itemset frequencies, the Metropolis approach with soft constraints is used.

In Figure 3.13 examples of randomizations preserving either the clustering structure or the frequencies of two itemsets in addition to the row and columns sums are shown. The original matrix is the same as shown in Figure 2.2 in

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

(a) A 0–1 matrix

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

(b) Clustering preserved

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

(c) Frequencies of itemsets *AB* and *BH* preserved

Figure 3.13: Examples of randomizations that preserve clustering structure or itemset frequencies in addition to row and column margins: (a) original 0–1 dataset; (b) a randomized sample preserving the two clusters (dashed rule); (c) a randomized sample preserving the frequencies of the itemsets *AB* and *BH*.

Section 2.2.3. We note that difference between the original matrix 3.14a and the clustering-preserved randomized matrix 3.14b is fairly small, that is, the clustering structure of the data in addition to the row and column margins explains most of the structure in the data. Also the frequencies of itemsets *AB* and *BH* explain most of the data.

3.2.7 Related Randomization Methods

Next, we discuss related work on randomization methods. Obviously, significance testing has received a large amount of attention. Excellent general sources on a variety of randomization approaches include books written by Good [Goo00], Besag [Bes04], Westfall and Young [WY93], Casella and Berger [CB01], and Edgington [Edg95].

Defining the significance of discovered patterns has attracted a lot of attention in data mining. Many papers work on the significance of association rules [SBM98, TKS02, Hä10]. The χ^2-test is used by Silverstein *et al.* [SBM98] for significance testing of correlation rules that are generalizations of association rules. Functional dependencies and logic are used for pruning out nonsignificant patterns algorithmically by Liu *et al.* [LHM99, LHM01]. Using inference to prune out nonsignificant correlations quickly was done by Xiong *et al.* [XSTK04].

Definitions and views on patterns other than frequent itemsets or association rules can be found, for example, in DuMouchel and Pregibon [DP01], and Jaroszewicz and Simovici [JS01]. The randomization is used by Ukkonen and Mannila [UM07] to study partial orders. Approaches to assess results on graphs via randomization techniques are introduced by Hanhijärvi *et al.* [HGP09], Newman *et al.* [NWS02], and Ying and Wu [YW09]. Testing the significance of patterns with clustering preserving randomization is studied by Vuokko and Kaski [VK10]. The randomization of time series is studied by Long *et al.* [LSF+01] and Schreiber [Sch98], and evaluating the quality of segmentations by Haiminen *et al.* [HMT07]. Using randomization to assess gene periodicity results is studied by Kallio *et al.* [KVO+11]. A more methodological view on pruning nonsignificant patterns using multiple hypothesis testing concepts can be found in Megiddo and Srikant [MS98].

Various null-models have been studied in many application areas. In ecology, the use of null-models in testing the significance of discoveries is quite widespread. For example, in the analysis of nestedness, there are several slightly different null-models that have been under careful study in recent years [MS07, UG07a, UG07b]. Null-models for temporal trends in biological records are studied by Wonham and Pachepsky [WP06] and in geographic range size evolution by Waldron [Wal07] and Storch *et al.* [SSR+08].

Recently, Tijl De Bie *et al.* have proposed a method for defining null-models semi-automatically [De 09, KDB10, DBKS10]. The idea is to preserve given statistics in expectation and choose the null-distribution that maximizes entropy. The principle of maximum entropy [Jay57a, Jay57b] guarantees that the null-model defined in this way makes the least additional assumptions about the underlying null-distribution. The work by Tijl De Bie has connections to the Rasch models widely used in psychometrics. Rasch model is a simple randomization model that is used to assess, for example, abilities of people or difficulties of tasks [Ras61, BF07].

Sub-sampling methods such as bootstrapping [Efr79, ET93, Man06] use randomization to study the properties of the underlying distribution, but they are not used for testing the data against some null-model as we intend in this thesis. Randomization is widely used in many other context as well, for example, in privacy preserving [AY08], in randomized algorithms for guaranteeing stable performance [MR95], and in physics to model the nature [Buc10]. For other types of approaches to significance testing of patterns, see work by Jaroszewicz [Jar08] and Webb [Web07, Web08].

CHAPTER 4

Applying Randomization

In the previous chapter we discussed various approaches for producing randomized datasets. In this chapter we show how these randomization methods can be applied in practice. For each approach we perform a case analysis and discuss the practical properties of the methods.

4.1 Assessing Patterns in Matrices

First we study assessing patterns in matrices. The randomizations methods for matrices were introduced in Sections 3.2.3 and 3.2.4 corresponding to Publications I and II. As the latter publication is a generalization of the former, here we will only apply the improved approach and method of Publication II. The implementation of the randomization method *SwapConstrained* is publicly available [Oja10].

To demonstrate the use of *SwapConstrained*, we assess the results of k-means clustering and principal component analysis on ten different datasets. The goal is to identify whether the clustering and PCA structures are explained by the row and column value distributions. The basic properties of the datasets are given in Table 4.1. The RANDOM dataset is a random matrix where the elements are independently chosen uniformly from range $[0, 1]$. It is the same matrix as used also in Publications I and II. The other datasets are real-life datasets taken from NCBI's Gene Expression Omnibus [EDL02] and UCI machine learning repository [AN11]. These datasets are not previously analyzed in any publication of this thesis.

Short descriptions of the real-life datasets are as follows: The COMMUNITY dataset contains normalized socio-economic, law-enforcement and crime data for different communities within the United States. The MOVEMENT

Dataset	Rows	Columns	Missing	Type	Attempts
RANDOM	100	100	No	Sim.	94
GDS1335	24	768	No	Sim.	51
GDS1385	15	3757	No	Sim.	44
GDS2481	32	1737	No	Sim.	64
GDS2960	101	4132	No	Diss.	126
COMMUNITIES	1994	122	0.151	Sim.	123
MOVEMENT	360	90	No	Sim.	100
PARKINSONS	195	22	No	Diss.	41
PAGEBLOCKS	5473	10	No	Diss.	37
SECOM	1567	590	0.045	Diss.	175

Table 4.1: Properties of the datasets. For each dataset the number of rows and columns are given as well as the proportion of missing values. Type and attempts give the parameters for *SwapConstrained*, that is, whether the features are similar/dissimilar and how many attempts per element is used.

dataset contains 360 time-series of the location of a moving hand, classified into different gestures. The PARKINSONS dataset contains biomedical voice measurements from people with and without Parkinson's disease. The PAGEBLOCKS dataset contains features for blocks in page layouts of different documents. The SECOM dataset contains sensor measurements from a semiconductor manufacturing process. The GDS1335 and GDS2481 datasets contain gene expression measurements for juvenile hormone III effect on anterior midgut in pine engraver. The GDS1385 dataset contains gene expression measurements for hepatocellular carcinoma induced by choline-deficient L-amino-acid-defined diet in brown rat, and the GDS2960 dataset contain gene expression measurements of cultured skin fibroblasts from patients with Marfan syndrome.

To use *SwapConstrained*, we have to first define which randomization approach is used, that is, whether the features are assumed to be similar or dissimilar (that is, measured using different scales). However, if the features are measured using the same scale but the value distributions are still very different, the approach for dissimilar features should be used. This guarantees that there exists enough swappable quartets for *SwapConstrained*. Note that we can always apply the approach for dissimilar features to any data. However, if the value distributions of the features are really similar, we can preserve the distributions more accurately in randomization by using the specific approach. In Table 4.1, the applied randomization approach is given for each dataset. The suitable approach was automatically detected by *SwapConstrained*. The number of randomization steps, attempts, is decided based on the theoretical

Dataset	Swaps	Rate	Time	Dist.	Diff.	Row err.	Col. err.
RANDOM	4.2	0.04	0.1s	1.40	1.00	0.0117	0.0118
GDS1335	4.5	0.09	0.1s	1.22	1.00	0.0025	0.0272
GDS1385	4.8	0.11	0.5s	1.35	1.00	0.0008	0.0370
GDS2481	5.1	0.08	0.6s	1.23	1.00	0.0014	0.0193
GDS2960	6.0	0.05	32s	1.41	1.00	0.0009	0.0104
COMMUNITIES	26.0	0.21	19s	0.75	1.00	0.0102	0.0491
MOVEMENT	5.1	0.05	0.4s	1.30	1.00	0.0131	0.0045
PARKINSONS	4.2	0.10	0.0s	1.15	1.00	0.0338	0.0065
PAGEBLOCKS	4.7	0.13	1.5s	1.16	0.99	0.0538	0.0010
SECOM	39.9	0.23	1m57s	1.23	1.00	0.0039	0.0084

Table 4.2: Statistics of randomizations with *SwapConstrained*. Swaps: accepted attempts per element. Rate: acceptance probability of attempted swaps. Time: running time to produce one randomized sample. Distance: normalized root mean square distance between the rank matrices. Difference: proportion of changed elements. Row and column error: average L_1-CDF rank error in row or column value distributions of randomized samples.

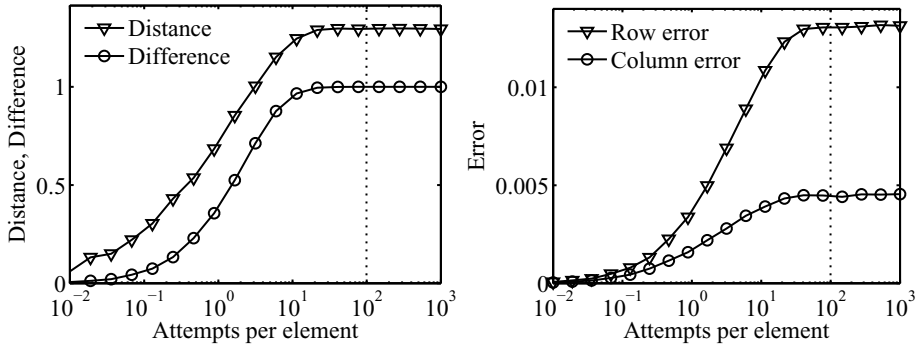
analysis presented in Publication II. With these automatically determined parameters we can apply the *SwapConstrained* method to produce 999 randomized samples. Note that we are using the backward-forward sampling approach by Besag to produce these samples, see Section 3.1.4.

The next step before calculating the empirical p-values is to assess the quality of the randomized samples, that is, how random they are and how well the row and column value distributions are preserved. In Table 4.2 we give some basic statistics of the randomized datasets. We note that on the average each element is swapped a few times. To estimate convergence we use two difference measures. The first is the normalized root mean square distance between the original matrix D and the randomized matrix $\widehat{D}$,

$$d(D,\widehat{D}) = \frac{1}{\sigma_D}\sqrt{\frac{\sum_{i,j}|D_{ij}-\widehat{D}_{ij}|^2}{nd}},$$

used in Publication I. We apply this distance measure to the rank matrices where the values are replaced with their corresponding ordinal values. For a fully permuted matrix where the row and column value distributions are not preserved at all, this distance measure is approximately $\sqrt{2}$. The second difference measure is the proportion of changed elements; we note that in all datasets approximately 100% of the elements have changed their locations.

The row and column errors are the average L_1 errors between the cumulative distribution functions of rows and columns of the original rank matrix and



(a) Convergence of distance and difference (b) Convergence of error

Figure 4.1: Convergence of (a) distance and difference, as well as (b) row and column error with *SwapConstrained* on MOVEMENT dataset. The dashed line corresponds to the used number of attempts.

the randomized rank matrix. For example, an average row error of 0.01 means that on the average the ranks are preserved with 1% accuracy in each row. In Figure 4.1 the convergence of the distances and errors are shown on the MOVEMENT dataset. We can conclude that *SwapConstrained* has produced well randomized samples preserving the row and column value distributions accurately. Nevertheless, evaluating the statistics shown in Table 4.2 is always a good practice before using the randomizations to significance testing.

The actual significance testing of the data mining results is pretty straightforward. We apply k-means clustering and principal component analysis to the whitened data where the features are normalized to zero mean and unit variance after randomization. We use the structural measures given in Sections 2.1.1 and 2.1.2. The missing values were replaced after randomization with column-wise medians. For most datasets, a meaningful number of clusters was known. For others, $k = 10$ was used. In Table 4.3 we give the significance testing results in the datasets for these two data mining methods. We notice that only the results on the artificial RANDOM are nonsignificant, while all the other results are clearly significant. Thus, the real datasets contain clustering and PCA structures that are not explained by the row and column distributions.

4.2 Identifying Interdependent Phenomena

In the previous section we studied k-means clustering and principal component analysis as separate phenomena. Next, we apply the approach introduced in

	k-means				PCA			
Dataset	k	Orig.	Rand.	p-value	$\widehat{d}$	Orig.	Rand.	p-value
RANDOM	10	8.46k	8.48k	**0.252**	5	0.17	0.17	**0.381**
GDS1335	2	14.7k	16.2k	0.001	5	0.57	0.38	0.001
GDS1385	2	42.1k	45.5k	0.001	5	0.69	0.54	0.001
GDS2481	2	44.1k	48.7k	0.001	5	0.56	0.38	0.001
GDS2960	2	356k	407k	0.001	5	0.37	0.09	0.001
COMMUNITIES	10	134k	219k	0.001	5	0.58	0.13	0.001
MOVEMENT	15	10.3k	26.4k	0.001	5	0.81	0.22	0.001
PARKINSONS	2	2.80k	3.44k	0.001	5	0.88	0.51	0.001
PAGEBLOCKS	5	23.7k	36.5k	0.001	5	0.89	0.58	0.001
SECOM	2	707k	730k	0.001	5	0.17	0.03	0.001

Table 4.3: Significance testing results for normalized k-means clustering and principal component analysis with *SwapConstrained*. For each dataset, the original structural measure, the average structural measure on 999 randomized datasets and the empirical p-value is given. The k is the number of clusters used for k-means and $\widehat{d}$ is the new dimension used for *PCA*. The p-values in boldface correspond to nonsignificant results with significance level $\alpha = 0.05$.

Section 3.2.6 for identifying patterns that are independent of the clustering structure. In Section 3.2.6 and Publication III the *ClusterSwap* method was introduced for randomizing binary matrices while preserving the clustering structure. Next, we generalize this approach to real-valued matrices and apply it to assess the principal components and pair-wise correlations.

To produce randomizations preserving the clustering structure of a matrix, we first apply some clustering algorithm to the original data to define the clustering structure to be preserved. In the experiments we use the same normalized k-means clustering as in Section 4.1. To preserve this structure in randomization, we separately randomize each of the submatrices, clusters, using the *SwapConstrained* method. This guarantees that we approximately preserve the value distribution of points and features inside each cluster. This method, *ClusterSwap*, is a direct generalization of the binary version given in Publication III. Recently, Vuokko and Kaski introduced a related clustering preserving randomization method that uses the connection between PCA and k-means to produce the randomizations [VK10]. It preserves the general clustering structure of the data whereas *ClusterSwap* preserves a specific given clustering.

To test that *ClusterSwap* really produces randomized matrices preserving the original clustering structure, we assess the significance of the k-means clustering using *ClusterSwap* when the same clustering is preserved. In Table 4.4

		k-means			PCA			
Dataset	k	Orig.	Rand.	p-value	$\widehat{d}$	Orig.	Rand.	p-value
RANDOM	10	8.46k	8.46k	**0.512**	5	0.17	0.17	**0.782**
GDS1335	2	14.7k	14.8k	**0.059**	5	0.57	0.43	0.001
GDS2481	2	44.1k	44.2k	**0.052**	5	0.56	0.35	0.001
COMMUNITIES	10	134k	135k	**0.136**	5	0.58	0.43	0.001
PARKINSONS	2	2.80k	2.79k	**0.746**	5	0.88	0.67	0.001
PAGEBLOCKS	5	23.7k	23.7k	**0.657**	5	0.89	0.81	0.001

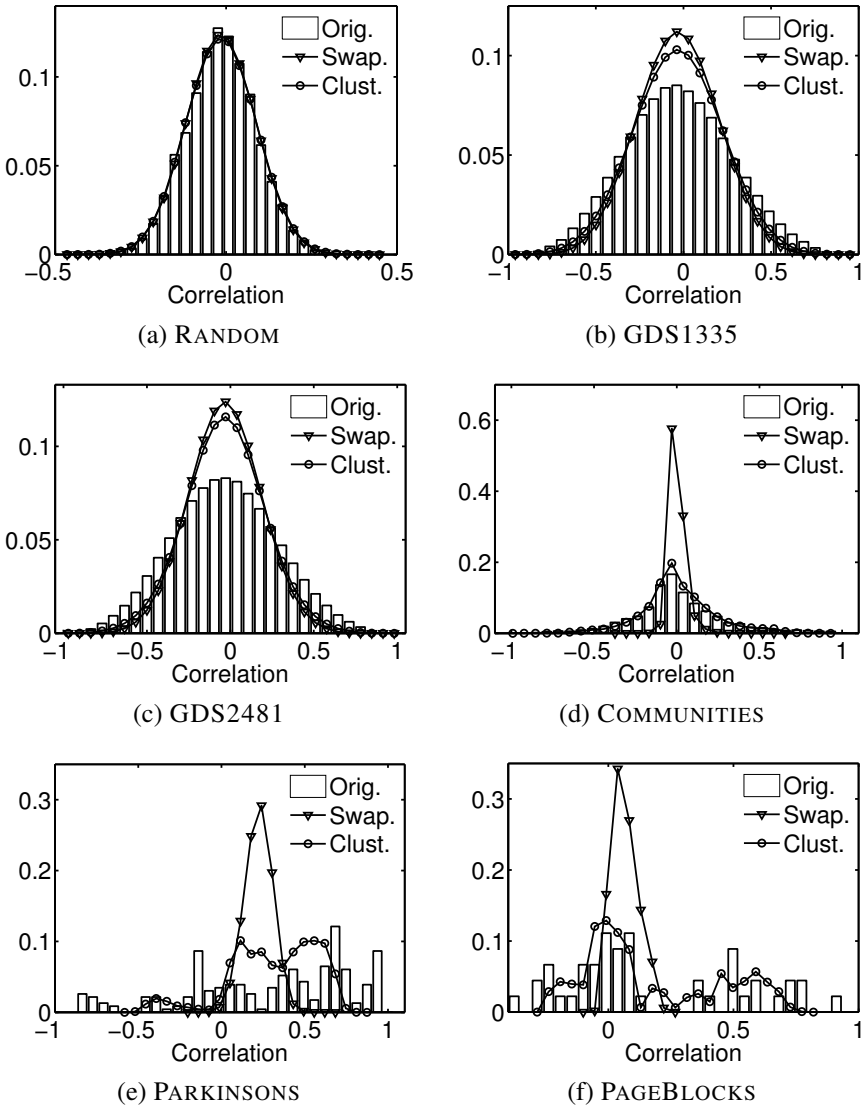
Table 4.4: Significance testing results for normalized k-means clustering and principal component analysis with *ClusterSwap* using 999 randomized samples. The columns are as in Table 4.3. Empirical p-values in boldface correspond to nonsignificant results with significance level $\alpha = 0.05$.

we give significance testing results for a small, random subset of the datasets using *ClusterSwap*. We observe that the k-means clustering results are explained by *ClusterSwap* as expected. None of the PCA results are explained by the clustering structure. However, with *ClusterSwap* randomization most of the PCA structural measures are closer to the original ones as with *SwapConstrained*, that is, the clustering structure explains some of the observed PCA structure.

Until this point we have studied only single large patterns like the clustering structure of the data. Often such patterns are found to be significant as it can suffice that there is some small part in the data that is not random. Next, we apply the approach to study multiple smaller patterns. In such case it is more likely to obtain both significant and nonsignificant patterns.

We apply the *SwapConstrained* and *ClusterSwap* randomizations to find the significant pair-wise correlations between the columns. In Publication II a similar setting was studied but then only *SwapConstrained* randomization and column permutation were used. We calculate all the pairwise correlations in the original datasets and compare them to the distribution of pairwise correlations in 99 randomized datasets with both *SwapConstrained* and *ClusterSwap*. In Figure 4.2, we present the distribution of pair-wise correlations in the original datasets and in randomized datasets with *SwapConstrained* and *ClusterSwap*. We notice that *ClusterSwap* preserves the pair-wise correlations better than *SwapConstrained* as expected. On PARKINSONS and PAGEBLOCKS datasets we notice that *ClusterSwap* has preserved the complex distribution of the correlations surprisingly well although only the structures with two and five clusters are preserved, respectively.

In Table 4.5 we show the number of significant pairwise correlations us-



(a) RANDOM (b) GDS1335

(c) GDS2481 (d) COMMUNITIES

(e) PARKINSONS (f) PAGEBLOCKS

Figure 4.2: Distributions of pairwise correlations between the columns in six datasets for the original data and for randomized data with *SwapConstrained* and *ClusterSwap*. Triangles: data randomized with *SwapConstrained* while preserving the row and column value distributions. Circles: data randomized with *ClusterSwap* while additionally preserving the clustering structure with k clusters where k is given in Table 4.4.

	Orig.	*SwapConstrained*			*ClusterSwap*		
Dataset	Med.	Med.	B-H thresh.	#Sign.	Med.	B-H thresh.	#Sign.
RANDOM	0.00	0.00	— / 0.32	0	0.00	— / 0.32	0
GDS1335	-0.00	0.00	0.69 / 0.69	3360	0.00	0.88 / 0.88	117
GDS2481	-0.00	0.00	0.68 / 0.68	23474	0.00	0.84 / 0.84	2289
COMMUNITIES	0.01	0.03	0.19 / 0.19	1671	0.02	0.74 / 0.74	179
PARKINSONS	0.44	0.26	0.41 / 0.41	120	0.37	0.74 / 0.73	56
PAGEBLOCKS	0.08	0.08	0.39 / 0.14	16	0.08	0.78 / 0.73	3

Table 4.5: Significance testing results for pairwise correlations using two different randomization approaches: *SwapConstrained* and *ClusterSwap*. For each dataset, the median pairwise correlation between the columns in the original dataset and in randomized datasets are given. The number of significant pairwise correlations (#Sign.) is calculated using Benjamini-Hochberg approach with *SwapConstrained* and *ClusterSwap*. The B-H threshold contains the smallest significant and the largest insignificant pairwise correlation.

ing *SwapDiscretized* and *SwapConstrained*. To correct for multiple comparisons, we apply the Benjamini-Hochberg approach described in Section 2.2.5. We restrict the false discovery rate below 0.05. Firstly, the dataset RANDOM does not contain any significant pair-wise correlations when significance is estimated by using either method. In the other five datasets, the clustering structure explains about 75–95% of the pair-wise correlations that were significant when only the row and column value distributions were preserved with *SwapConstrained*. This example shows well how adding constraints to the randomizations can explain the significance of some other seemingly remote patterns.

4.3 Studying Dependent Features in Classification

Next, we study the performance and properties of classification algorithms on those of the previous datasets that contain class labels. We apply the two tests from Publication V, introduced in Section 3.2.1, to assess whether the classifier has found a real connection between the data and the class labels and whether the classifier is using the dependencies between the features to improve the classification accuracy.

Applying these two tests is easy and straightforward. First, we train a classifier on the original dataset and calculate the classification error. Then we produce randomized datasets, and repeat the same procedure on them. We use a decision tree classifier and 1-nearest neighbor classifier, described in Sec-

Decision tree

		Original	Test 1		Test 2	
Dataset	Classes	Err. (Std)	Err. (Std)	p-value	Err. (Std)	p-value
RANDOM	10	0.49 (0.02)	0.50 (0.06)	**0.439**	0.53 (0.05)	**0.227**
GDS1335	2	0.25 (0.00)	0.30 (0.14)	**0.450**	0.25 (0.03)	**0.781**
GDS1383	2	0.33 (0.00)	0.50 (0.23)	**0.327**	0.35 (0.05)	**0.601**
GDS2481	2	0.16 (0.00)	0.51 (0.18)	0.030	0.18 (0.01)	**0.244**
GDS2960	2	0.23 (0.04)	0.48 (0.07)	0.002	0.22 (0.04)	**0.491**
MOVEMENT	15	0.32 (0.02)	0.93 (0.01)	0.001	0.53 (0.03)	0.001
PARKINSONS	2	0.15 (0.02)	0.27 (0.02)	0.001	0.12 (0.02)	**0.834**
PAGEBLOCKS	5	0.03 (0.00)	0.10 (0.00)	0.001	0.03 (0.00)	**0.989**
SECOM	3	0.10 (0.00)	0.11 (0.01)	**0.350**	0.11 (0.01)	**0.257**

Table 4.6: Classification errors and empirical p-values obtained with decision tree classifier for Test 1 and Test 2. The empirical p-values are calculated over 999 randomized samples. Classification on the original data is repeated ten times. Ten-fold cross-validation is used for datasets with more than 50 rows, for smaller datasets leave-one-out cross-validation is used. The average values and standard deviations of the classification errors are given. The p-values in boldface correspond to nonsignificant results with significance level $\alpha = 0.05$.

tion 2.1.5. As the classification error, we use the leave-one-out cross-validation error for small datasets, namely GDS1335, GDS1383 and GDS2481, and the 10-fold cross validation error for the other datasets. Only the COMMUNITIES dataset was left out from the previously studied datasets as it does not contain labels. In the RANDOM dataset we assigned the points uniformly in random into 10 classes.

To assess the performance of the classifiers, we produced 999 randomized datasets with both Test 1 and Test 2, that is, by permuting the labels or by permuting the features inside each class. In Table 4.6 we give the significance testing results for the decision tree classifier and in Table 4.7 for the 1-nearest neighbor classifier. Compared to the results in Publication V there are now more datasets where the classifiers have not found significant class structures, that is, the p-value of Test 1 is high. The reason is that the original class labels especially in the GDS1335 and GDS1385 datasets are fairly meaningless.

The MOVEMENT dataset is now the only dataset where the p-value of Test 2 is small with both classifiers, i.e., both classifiers are clearly using the dependencies between the features to increase the classification accuracy. We also applied the naive Bayes classifier to all of these datasets and obtained the expected results: the dependencies between the features were not used in any

1-nearest neighbor

		Original	Test 1		Test 2	
Dataset	Classes	Err. (Std)	Err. (Std)	p-value	Err. (Std)	p-value
RANDOM	10	0.56 (0.03)	0.50 (0.06)	**0.847**	0.50 (0.05)	**0.841**
GDS1335	2	0.04 (0.00)	0.29 (0.12)	0.004	0.83 (0.02)	0.001
GDS1383	2	0.27 (0.00)	0.49 (0.15)	**0.121**	0.47 (0.11)	**0.064**
GDS2481	2	0.53 (0.00)	0.52 (0.10)	**0.614**	0.50 (0.00)	**1.000**
GDS2960	2	0.15 (0.01)	0.49 (0.06)	0.001	0.00 (0.00)	**1.000**
MOVEMENT	15	0.14 (0.01)	0.93 (0.02)	0.001	0.43 (0.02)	0.001
PARKINSONS	2	0.04 (0.01)	0.37 (0.03)	0.001	0.04 (0.02)	**0.541**
PAGEBLOCKS	5	0.04 (0.00)	0.19 (0.00)	0.001	0.06 (0.00)	0.001
SECOM	3	0.09 (0.00)	0.12 (0.02)	**0.092**	0.15 (0.07)	**0.135**

Table 4.7: Classification errors and empirical p-values obtained with 1-nearest neighbor classifier for Test 1 and Test 2. The p-values are calculated over 999 randomized samples. Classification on the original data is repeated ten times. Ten-fold cross-validation is used for datasets with more than 50 rows, for smaller datasets leave-one-out cross-validation is used. The average values and standard deviations of the classification errors are given. The p-values in boldface correspond to nonsignificant results with significance level $\alpha = 0.05$.

of the datasets. This happens because naive Bayes explicitly assumes that the features are independent. To conclude, these two tests provide simple descriptive features for the classifier performance.

4.4 Assessing Queries in Multi-Relational Databases

Compared to the approaches discussed in the previous sections, applying the randomization on multi-relational databases is not as straightforward. The different randomization strategies for multi-relational databases were presented in Section 3.2.5, corresponding to Publication IV. These approaches are still more conceptual ideas than practical tools. Nevertheless, next we apply the approaches for studying a database BOOK-CROSSING [ZMKL05] that has not been analyzed previously in any publication of this thesis.

The BOOK-CROSSING dataset consists of ratings given by users for books they have read, list of authors for the books, and some demographic information for the users. In Table 4.8 we give a list of the binary relations in BOOK-CROSSING dataset that are used in this section. We interpret that a user has read a certain book if he or she has rated the book. From the original dataset we have removed users that have not read any books, books that have not been

Relation	Description	Rows	Columns	Ones / row	Ones / col
UB	User×Book	51525	222584	14.11	3.27
CU	Country×User	186	51525	242.57	1
AU	Age×User	99	51525	520.45	1
BA	Book×Author	222584	85128	1	2.61
BT	Book×Title length	222584	256	1	869.47

Table 4.8: Summary of binary relations in the BOOK-CROSSING dataset.

	Mean (Std)	p-value
Original	10.94	
$\mathsf{sw}(\mathrm{UB})\bowtie\mathrm{BA}$	12.97 (0.00)	0.001
$\mathrm{UB}\bowtie\mathsf{sw}(\mathrm{BA})$	13.82 (0.01)	0.001

Table 4.9: Significance testing results for Hypothesis 4.1, "People prefer familiar authors", on input relations $\mathrm{UB}\bowtie\mathrm{BA}$. Mean and std are the average and standard deviation of Statistic 4.1 in the original input data (first row) and in the different randomizations using 999 randomized samples. The randomization $\mathrm{UB}\bowtie\mathsf{sw}(I_\mathrm{B})\bowtie\mathrm{BA}$ is equivalent to $\mathrm{UB}\bowtie\mathsf{sw}(\mathrm{BA})$.

read by anyone and users that have not given the demographic information.

Next, we study three simple hypotheses on this dataset to demonstrate the use of the randomization approach.

Hypothesis 4.1. *People prefer familiar authors.*

As a structural measure we use the following statistics.

Statistic 4.1. *The average number of different authors whose books each user has read.*

The value of the statistic in the original dataset is 10.94, that is, people have read books on the average from about 11 different authors. For comparison, the number of books read by an average user is 14.11. In Table 4.9 we give significance testing results for the hypothesis by applying the randomization in two different places. By theory we know that randomization $\mathrm{UB}\bowtie\mathsf{sw}(I_\mathrm{B})\bowtie\mathrm{BA}$ is equivalent to $\mathrm{UB}\bowtie\mathsf{sw}(\mathrm{BA})$, since *BA* contains exactly one 1 in each row. Both randomizations regard the original result as significant. However, randomizing the connection between the books and the authors brings the value of the statistics much closer to the number of books read by an average user.

Hypothesis 4.2. *People in a given country read books with long/short title.*

		$\mathsf{sw}(\mathrm{CU})\bowtie\mathrm{UB}\bowtie\mathrm{BT}$		$\mathrm{CU}\bowtie\mathsf{sw}(\mathrm{UB})\bowtie\mathrm{BT}$		$\mathrm{CU}\bowtie\mathrm{UB}\bowtie\mathsf{sw}(\mathrm{BT})$	
Country	Orig.	Mean (Std)	p-val.	Mean (Std)	p-val.	Mean (Std)	p-val.
USA	31.78	31.31 (0.07)	0.001	32.82 (0.10)	0.001	37.66 (0.39)	0.001
Canada	31.30	31.29 (0.27)	**0.469**	32.82 (0.28)	0.001	37.66 (0.49)	0.001
Germany	31.07	31.31 (0.31)	**0.212**	32.80 (0.32)	0.001	37.63 (0.55)	0.001
UK	30.10	31.30 (0.35)	0.001	32.82 (0.35)	0.001	37.68 (0.89)	0.001
Italy	29.85	31.33 (0.73)	0.019	32.81 (0.82)	0.001	37.50 (1.57)	0.001
Australia	29.84	31.32 (0.43)	0.001	32.81 (0.48)	0.001	37.63 (1.56)	0.001
Spain	29.17	31.32 (0.40)	0.001	32.84 (0.43)	0.001	37.66 (0.75)	0.001
France	26.27	31.30 (0.63)	0.001	32.82 (0.61)	0.001	37.66 (0.87)	0.001

Table 4.10: Significance testing results for Hypothesis 4.2, "People in a given country read books with long/short title", on input relations $\mathrm{CU}\bowtie\mathrm{UB}\bowtie\mathrm{BT}$ for countries having at least 1000 users. The original values of Statistic 4.2, with mean and std over 999 randomized samples are given. The randomization $\mathrm{CU}\bowtie\mathsf{sw}(I_{\mathrm{U}})\bowtie\mathrm{UB}\bowtie\mathrm{BT}$ is equivalent to $\mathsf{sw}(\mathrm{CU})\bowtie\mathrm{UB}\bowtie\mathrm{BT}$, and the randomization $\mathrm{CU}\bowtie\mathrm{UB}\bowtie\mathsf{sw}(I_{\mathrm{B}})\bowtie\mathrm{BT}$ is equivalent to $\mathrm{CU}\bowtie\mathrm{UB}\bowtie\mathsf{sw}(\mathrm{BT})$.

Statistic 4.2. *For a given country, the average of the average title lengths of the books read by each user, measured in the number of characters.*

The significance testing results for Hypothesis 4.2 are given in Table 4.10. Here we are using the two-tailed empirical p-value (2.2). We notice that by permuting the countries, equalling $\mathsf{sw}(\mathrm{CU})$, the average title length of the books read by users in Canada and Germany is not surprising. All the other results are not explained by any simple structure in the data. It is worth noticing that almost half of the users in the BOOK-CROSSING dataset are from USA (32411) and that Canada (4977) and Germany (4120) form the next largest user groups. Also, most of the book titles are in English. The average values of the statistic in the randomized datasets are close to the following values: The average title length of all books is 37.65, the average title length of a read book is 33.00, and the average title length of the books that an average user has read is 31.31 (this equals Statistic 4.2 when the country is left out). In this case the randomizations have broken practically all connections and produced results close to some preknown statistics.

Hypothesis 4.3. *People in different age groups read books written by different authors.*

Statistic 4.3. *The L_1 distance between the distribution of authors that an average user in the given age group and an average user not in the given age group has read.*

		$\mathsf{sw}(\mathrm{AU}) \bowtie \mathrm{UB} \bowtie \mathrm{BA}$		$\mathrm{AU} \bowtie \mathsf{sw}(\mathrm{UB}) \bowtie \mathrm{BA}$		$\mathrm{AU} \bowtie \mathrm{UB} \bowtie \mathsf{sw}(\mathrm{BA})$	
Age	Orig.	Mean (Std)	p-val.	Mean (Std)	p-val.	Mean (Std)	p-val.
–19	1.02	0.60 (0.02)	0.001	0.68 (0.00)	0.001	0.95 (0.00)	0.001
20–29	0.62	0.47 (0.01)	0.001	0.42 (0.00)	0.001	0.55 (0.00)	0.001
30–39	0.49	0.47 (0.01)	0.020	0.40 (0.00)	0.001	0.46 (0.00)	0.001
40–49	0.57	0.53 (0.01)	0.005	0.45 (0.00)	0.001	0.54 (0.00)	0.001
50–59	0.72	0.58 (0.01)	0.001	0.52 (0.00)	0.001	0.67 (0.00)	0.001
60–	0.94	0.71 (0.02)	0.001	0.70 (0.00)	0.001	0.91 (0.00)	0.001

Table 4.11: Significance testing results for Hypothesis 4.3, "People in different age groups read books written by different authors", on input relations $\mathrm{AU} \bowtie \mathrm{UB} \bowtie \mathrm{BA}$. The original values of Statistic 4.3, with mean and std over 999 randomized samples are given. The randomization $\mathrm{AU} \bowtie \mathsf{sw}(I_{\mathrm{U}}) \bowtie \mathrm{UB} \bowtie \mathrm{BA}$ is equivalent to $\mathsf{sw}(\mathrm{AU}) \bowtie \mathrm{UB} \bowtie \mathrm{BA}$, and the randomization $\mathrm{AU} \bowtie \mathrm{UB} \bowtie \mathsf{sw}(I_{\mathrm{B}}) \bowtie \mathrm{BA}$ is equivalent to $\mathrm{AU} \bowtie \mathrm{UB} \bowtie \mathsf{sw}(\mathrm{BA})$.

The last hypothesis uses a more complicated statistic as the previous two. The significance testing results for this hypothesis are in Table 4.11. Now all the age groups seem to read books from significantly different authors than the other age groups using all three randomizations. However, randomizing the table BA has preserved the structure usually the most, that is, most of the result is due to the fact that people read certain books and not certain authors.

To conclude, by randomizing in multi-relational databases we can find what is the structural impact of each participating relation to the final result. However, making solid inferences from these multiple p-values is not straightforward. Nevertheless, we can say that practically all hypotheses studied in this section were significant and not explained by the basic properties of the relations. Only the average title length of the books read by Canadians and Germans were explained by the relation between the countries and the users.

CHAPTER 5

Conclusions and Discussion

In this thesis, we discussed randomization methods that can be used to assess the significance of data mining results on various types of data. In randomization based significance testing, a result is considered to be interesting if it is unlikely to obtain as good result on randomized data that share some basic properties with the original data. To apply the randomization approach, we have to first define what are the basic properties that should be preserved in randomization. Patterns that are explained by these properties are seen as uninteresting.

We introduced specific randomization approaches and methods for four different data mining scenarios, analyzed their properties theoretically and performed experiments on real and artificial datasets. First, we studied randomizing numerical and nominal matrices where the row and column value distributions are approximately preserved. Next, we described a general iterative randomization approach for finding new patterns that are not explained by any previous result. We also considered how to randomize in multi-relational databases and how to assess the structural impact of each relation. Finally, we introduced a simple permutation method for evaluating classifier performance, namely whether the feature dependencies are used to improve the classification accuracy.

The randomization of numerical and nominal matrices is the most studied problem in this thesis. In randomizing such data, we preserve the row and column value distributions approximately. If the features are measured using different scales, the feature-wise rank distributions of the observations are preserved instead. It is important to notice that by using the randomization approach we are specifically testing whether the obtained results depend on the preserved statistics. Obtaining a low or high p-value does not mean that the

result is good or bad, respectively, but merely whether it is explained by the row and column value distributions. In extreme cases, for example, the row and column value distributions can explain the whole data (a matrix consisting of the same value), or some data mining patterns can always be explained by these statistics (the highest value in the matrix). Thus, the user has to always separately consider whether the test is suitable for the specific application.

Finding mutually independent patterns from a single dataset using different types of data mining algorithms is a challenging task. The problem with the iterative randomization approach is that it is not very practical in its current form. We introduced a general idea and gave precise methods for a few different cases. Using the general Metropolis-Hastings approach by adding more and more constraints becomes easily impractical. Thus, applying the approach in new settings needs further research. Nevertheless, the general approach is attractive for identifying interdependent data mining patterns.

In multi-relational databases, there are many possible ways to produce the randomized datasets for significance testing. We can randomize each table in the database separately as well as each connection between the tables. Each of these tests provides information about the structural impact of the randomized relation. Compared to the single matrix case where we proposed one generally interesting test, handling these multiple tests and empirical p-values can be more demanding. We studied using the binary swap randomization in multi-relational databases. Applying the new *SwapConstrained* method in the setting is worth further research.

The new test for assessing whether the classifier is exploiting the dependencies between the features to improve the classification accuracy is relatively simple and effective. However, there are two fairly similar approaches that are not studied in this thesis and are worth further research. In the current approach, we randomize both the training and test data together; this tests whether the classifier algorithm can perform as well on data that do not have any dependencies between features. Another approach would be to randomize only the test data and keep the original classifier; this might be able to test better whether the obtained classifier is really using the dependencies between the features. This approach would be faster but has the problem that the test data should be large enough so that it is meaningful to be randomized. The last approach would randomize only the training data; this would test whether we can find a classifier that is certainly not using the dependencies between the features but still performs as well or better as the original classifier. However, the current approach of randomizing the training and test data together is also commonly used with the traditional permutation test, for example, in Golland *et al.* [GLMP05].

The requirement that the same data mining method should be applied also

on the randomized data is not as straightforward as it sounds. This is also evident from the previous discussion on how to randomize the labeled data for identifying if the classifier is using the feature dependencies. The conclusion is that it all depends on what we really want to test. For example, when assessing the significance of k-means clustering, we now repeat same k-means clustering on the randomized data. However, sometimes we would really like to know whether the specific obtained clustering structure is also evident on random data and not whether the random data contains some clustering structure that is as good as the original. To alleviate this problem, we could use the approach presented by Webb [Web07] where we divide the original data into two parts and use the first half to find some interesting patterns and the second half to assess their significance by calculating the original and randomized structural measures on that half with the fixed data mining pattern.

One significant problem with statistical tests in general is that a *statistically* significant difference is totally different from a *practically* significant difference, see, for example, Siegfried [Sie10]. For example, a difference of 0.001 between the average values of two sets can be statistically significant although it would not have any practical importance. This is related to the concept of *statistical power*, that is, the probability that a test will correctly regard the result as significant [Goo00]. The more observations, the larger the statistical power is. That is, if the two sets contain each one million observations, even a small difference in the average values seems statistically significant. Thus, the user has to always consider if the statistical significance corresponds to a practical significance in the specific application. The statistical power is a complex theme that is not really discussed in this thesis.

Another issue is when we assess a single data mining result that summarizes the whole data, such as the clustering structure or intrinsic dimension with PCA. Often with such global structural measures it can happen that the result is significant, although there is only a small part of the data that makes the whole original structural measure slightly better than on a fully random data. In such cases, it can be more useful to assess smaller patterns in the data to find out which parts of the data really convey some significant information and which parts are just random.

In this thesis, we formulated several randomization approaches for different types of data that provide useful tests for general data mining applications. These approaches have their limitations as discussed above. The tests are not suitable for all data mining scenarios, and hence the user has to carefully consider before drawing conclusions from the obtained significance testing results. Therefore, future work should focus on generalizing the randomization approach to new types of data mining problems. Nevertheless, the experiments performed in this thesis show that the new approaches are useful on

various applications, producing meaningful information that can be used to further improve or understand the obtained results. Evaluating the reliability of the results guarantees that the conclusions and recommendations are based on solid information—the new methods developed in this thesis provide the needed tools for various data mining applications.

Bibliography

[AdFDJ03] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43, 2003.

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD'93: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216. ACM, 1993.

[AMS+96] 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*, pages 307–328. American Association for Artificial Intelligence, 1996.

[AN11] Arthur Asuncion and David J. Newman. UCI machine learning repository, 2011. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[AV07] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *SODA'07: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. SIAM, 2007.

[AY08] Charu C. Aggarwal and Philip S. Yu, editors. *Privacy-Preserving Data Mining: Models and Algorithms*. Springer, 2008.

[BBV06] Ivona Bezáková, Nayantara Bhatnagar, and Eric Vigoda. Sampling binary contingency tables with a greedy start. In

SODA'06: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 414–423. SIAM, 2006.

[BC89] Julian Besag and Peter Clifford. Generalized Monte Carlo significance tests. *Biometrika*, 76(4):633–642, 1989.

[BC91] Julian Besag and Peter Clifford. Sequential Monte Carlo p-values. *Biometrika*, 78(2):301–304, 1991.

[BDX04] Stephen Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing Markov chain on a graph. *SIAM Review*, 46(4):667–689, 2004.

[Ber06] Pavel Berkhin. A survey of clustering data mining techniques. In Jacob Kogan, Charles Nicholas, and Marc Teboulle, editors, *Grouping Multidimensional Data*, chapter 2, pages 25–71. Springer, 2006.

[Bes04] Julian Besag. Markov chain Monte Carlo methods for statistical inference. `http://www.ims.nus.edu.sg/Programs/mcmc/files/besag_tl.pdf`, 2004.

[BF07] Trevor G. Bond and Christine M. Fox. *Applying the Rasch Model: Fundamental Measurement in the Human Sciences*. Lawrence Erlbaum, 2nd edition, 2007.

[BG04] Yoshua Bengio and Yves Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5:1089–1105, 2004.

[BH95] 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.

[BH02] Pierre Baldi and G. Wesley Hatfield. *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling*. Cambridge University Press, 2002.

[BND04] Ulisses Braga-Neto and Edward R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.

[Bon35] Carlo E. Bonferroni. Il calcolo delle assicurazioni su gruppi di teste. In *Studi in Onore del Professore Salvatore Ortu Carboni*, pages 13–60, 1935.

[Buc10] Mark Buchanan. Enter the matrix: the deep law that shapes our reality. *New Scientist*, April 2010.

[BV01] Alvis Brazma and Jaak Vilo. Gene expression data analysis. *Microbes Infect*, 3(10):823–829, 2001.

[CB01] George Casella and Roger L. Berger. *Statistical Inference*. Duxbury Resource Center, 2001.

[CC96] 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.

[CC03] George W. Cobb and Yung-Pin Chen. An application of Markov chain Monte Carlo to community ecology. *The American Mathematical Monthly*, 110(4):265–288, 2003.

[CDHL05] Yuguo Chen, Persi Diaconis, Susan P. Holmes, and Jun S. Liu. Sequential Monte Carlo methods for statistical analysis of tables. *Journal of the American Statistical Association*, 100(469):109–120, 2005.

[Con98] William J. Conover. *Practical Nonparametric Statistics*. Wiley Series in Probability and Statistics. Wiley, 3rd edition, 1998.

[DBKS10] Tijl De Bie, Kleanthis-Nikolaos Kontonasios, and Eirini Spyropoulou. A framework for mining interesting pattern sets. In *UP'10: Proceedings of the ACM SIGKDD Workshop on Useful Patterns*, pages 27–35. ACM, 2010.

[De 09] Tijl De Bie. Explicit probabilistic models for databases and networks. *CoRR*, abs/0906.5148, 2009.

[DG95] Persi Diaconis and Anil Gangolli. Rectangular arrays with fixed margins. In David Aldous, Persi Diaconis, Joel Spencer, and J. Michael Steele, editors, *Discrete Probability and Algorithms*, volume 72, pages 15–41. Springer, 1995.

[DHS00] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2000.

[DP01] William DuMouchel and Daryl Pregibon. Empirical Bayes screening for multi-item associations. In *KDD'01: Proceedings*

of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 67–76. ACM, 2001.

[Dye03] Martin Dyer. Approximate counting by dynamic programming. In *STOC'03: Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 693–699. ACM, 2003.

[Edg95] Eugene S. Edgington. *Randomization Tests*. Marcel Dekker, Inc., 3rd edition, 1995.

[EDL02] Ron Edgar, Michael Domrachev, and Alex E. Lash. Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002.

[Efr79] Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[ET93] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[FH51] Evelyn Fix and Joseph L. Hodges. Discriminatory analysis: Nonparametric discrimination: Consistency properties. Technical Report Project 21-49-004, Report Number 4, USAF School of Aviation Medicine, Randolf Field, Texas, 1951.

[FKH07] Michael P. Fay, Hyune-Ju Kim, and Mark Hachey. On using truncated sequential probability ratio test boundaries for Monte Carlo implementation of hypothesis tests. *Journal of Computational and Graphical Statistics*, 16(4):946–967, 2007.

[FPP07] David Freedman, Robert Pisani, and Roger Purves. *Statistics*. W. W. Norton & Company, 4th edition, 2007.

[FW98] Eibe Frank and Ian H. Witten. Using a permutation test for attribute selection in decision trees. In *ICML'98: Proceedings of the 15th International Conference on Machine Learning*, pages 152–160. Morgan Kaufmann, 1998.

[GCSR03] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, 2nd edition, 2003.

[Gey91] Charles J. Geyer. Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of 23rd Symposium on the Interface*, pages 156–163. Interface Foundation, Fairfax Station, 1991.

[GF03] Polina Golland and Bruce Fischl. Permutation tests for classification: Towards statistical significance in image-based studies. In *IPMI'03: Proceedings of the 18th International Conference on Information Processing and Medical Imaging*, pages 330–341. Springer, 2003.

[GLMP05] Polina Golland, Feng Liang, Sayan Mukherjee, and Dmitry Panchenko. Permutation tests for classification. In *COLT'05: Proceedings of the 18th Annual Conference on Learning Theory*, pages 501–515. Springer, 2005.

[GMMT07] 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):14, 2007.

[Goo00] Phillip Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer Series in Statistics. Springer, 2nd edition, 2000.

[HAD03] Tailen Hsing, Sanju Attoor, and Edward R. Dougherty. Relation between permutation-test p values and classifier error estimates. *Machine Learning*, 52(1-2):11–30, 2003.

[Ham10] Greg Hamerly. Making k-means even faster. In *SDM'10: Proceedings of the 10th SIAM International Conference on Data Mining*, pages 130–140. SIAM, 2010.

[Has70] W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[HGP09] Sami Hanhijärvi, Gemma C. Garriga, and Kai Puolamäki. Randomization techniques for graphs. In *SDM'09: Proceedings of the 9th SIAM International Conference on Data Mining*, pages 780–791. SIAM, 2009.

[HMS01] David J. Hand, Heikki Mannila, and Padhraic Smyth. *Principles of data mining*. MIT Press, 2001.

[HMT07] Niina Haiminen, Heikki Mannila, and Evimaria Terzi. Comparing segmentations by applying randomization techniques. *BMC Bioinformatics*, 8(1):171, 2007.

[Hol79] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[Hot33] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[HY01] David J. Hand and Keming Yu. Idiot's Bayes—not so stupid after all? *International Statistical Review*, 69(3):385–398, 2001.

[Hä10] Wilhelmiina Hämäläinen. *Efficient search for statistically significant dependency rules in binary data*. PhD thesis, University of Helsinki, 2010.

[Jar08] Szymon Jaroszewicz. Interactive HMM construction based on interesting sequences. In *LeGo'08: Local Patterns to Global Models Workshop at the the European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 82–91. Springer, 2008.

[Jay57a] Edwin Thompson Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.

[Jay57b] Edwin Thompson Jaynes. Information theory and statistical mechanics II. *Physical Review*, 108(2):171–190, 1957.

[Jen92] David Jensen. *Induction with randomization testing: decision-oriented analysis of large data sets*. PhD thesis, Washington University, 1992.

[JS01] Szymon Jaroszewicz and Dan A. Simovici. A general measure of rule interestingness. In *PKDD'01: Proceedings of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 253–265. Springer, 2001.

[KDB10] Kleanthis-Nikolaos Kontonasios and Tijl De Bie. An information-theoretic approach to finding informative noisy tiles in binary databases. In *SDM'10: Proceedings of the 10th SIAM International Conference on Data Mining*, pages 153–164. SIAM, 2010.

[KVO+11] 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.

[LHM99] Bing Liu, Wynne Hsu, and Yiming Ma. Pruning and summarizing the discovered associations. In *KDD'99: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 125–134. ACM, 1999.

[LHM01] Bing Liu, Wynne Hsu, and Yiming Ma. Identifying non-actionable association rules. In *KDD'01: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 329–334. KDD, 2001.

[Llo57] Stuart P. Lloyd. Least squares quantization in PCM. Technical report, Bell Laboratories, 1957.

[Llo82] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[LPW06] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2006.

[LSF+01] Chris Long, John Suckling, Jalal Fadili, Gemma Calvert, O 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(2):61–78, 2001.

[LV07] John A. Lee and Michel Verleysen. *Nonlinear Dimensionality Reduction*. Information Science and Statistics. Springer, 1st edition, 2007.

[Man06] Bryan F. J. Manly. *Randomization, Bootstrap and Monte Carlo Methods in Biology*. Chapman & Hall/CRC, 3rd edition, 2006.

[Mas51] Frank Massey. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[MH06] Denton Morrison and Ramon Henkel, editors. *The Significance Test Controversy: A Reader*. Aldine Transaction, 2006.

[MIM10] Malik Magdon-Ismail and Konstantin Mertsalov. A permutation approach to validation. In *SDM'10: Proceedings of the 10th SIAM International Conference on Data Mining*, pages 882–893. SIAM, 2010.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[MRR+53] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Mici Teller, and Edward Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[MS98] Nimrod Megiddo and Ramakrishnan Srikant. Discovering predictive association rules. In *KDD'98: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 274–278. AAAI Press, 1998.

[MS07] Jeffrey E. Moore and Robert K. Swihart. Toward ecologically explicit null models of nestedness. *Oecologia*, 152(4):763–777, 2007.

[MSP05] Annette M. Molinaro, Richard Simon, and Ruth M. Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[MT06] Ravi Montenegro and Prasad Tetali. Mathematical aspects of mixing times in Markov chains. *Foundations and Trends in Theoretical Computer Science*, 1:237–354, 2006.

[NCS02] 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.

[NWS02] Mark Newman, Dunkan J. Watts, and Steven Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 1):2566–2572, 2002.

[OG09] Markus Ojala and Gemma Garriga. Permutation tests for studying classifier performance. In *ICDM'09: Proceedings of the 9th IEEE International Conference on Data Mining*, pages 908–913. IEEE, 2009.

[Oja10] Markus Ojala. Implementation of SwapConstrained. `http://www.cis.hut.fi/mrojala/randomization/`, 2010.

[OVK+08] Markus Ojala, Niko Vuokko, Aleksi Kallio, Niina Haiminen, and Heikki Mannila. Randomization of real-valued matrices for assessing the significance of data mining results. In *SDM'08: Proceedings of the 8th SIAM International Conference on Data Mining*, pages 494–505. SIAM, 2008.

[Pea01] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[PM00] Dan Pelleg and Andrew Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML'00: Proceedings of the 17th International Conference on Machine Learning*, pages 727–734. Morgan Kaufmann, 2000.

[Qui93] J. Ross Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, 1st edition, 1993.

[Ras61] Georg Rasch. On general laws and the meaning of measurement in psychology. In *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability, IV*, pages 321–334. Berkeley: University of Chicago Press, 1961.

[RC04] Christian Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, 2nd edition, 2004.

[RG02] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. McGraw-Hill Higher Education, 3rd edition, 2002.

[Rys57] Herbert J. Ryser. Combinatorial properties of matrices of zeros and ones. *Canadian Journal of Mathematics*, 9:371–377, 1957.

[SBM98] Craig Silverstein, Sergey Brin, and Rajeev Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1):39–68, 1998.

[SC08] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, 2008.

[Sch78] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[Sch98] Thomas Schreiber. Constrained randomization of time series data. *Physical Review Letters*, 80(10):2105–2108, 1998.

[Sie10] Tom Siegfried. Odds are, it's wrong: Science fails to face the shortcomings of statistics. *Science News*, 177(7):26+, 2010.

[SSR+08] David Storch, Arnost L. Sizling, Jiri Reif, Jitka Polechova, Eva Sizlingova, and Kevin J. Gaston. The quest for a null model for macroecological patterns: geometry of species distributions at multiple spatial scales. *Ecology Letters*, 11(8):771–784, 2008.

[TI97] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. SIAM, 1997.

[TKS02] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *KDD'02: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 32–41. ACM, 2002.

[UG07a] Werner Ulrich and Nicholas J. Gotelli. Disentangling community patterns of nestedness and species co-occurrence. *Oikos*, 116(12):2053–2061, 2007.

[UG07b] Werner Ulrich and Nicholas J. Gotelli. Null model analysis of species nestedness patterns. *Ecology*, 88(7):1824–1831, 2007.

[UM07] Antti Ukkonen and Heikki Mannila. Finding outlying items in sets of partial rankings. In *PKDD'07: Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 265–276. Springer, 2007.

[VK10] Niko Vuokko and Petteri Kaski. Testing the significance of patterns in data with cluster structure. In *ICDM'10: Proceedings of the 10th IEEE International Conference on Data Mining*, pages 1097–1102. IEEE, 2010.

[Wal45] Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[Wal07] Anthony Waldron. Null models of geographic range size evolution reaffirm its heritability. *The American Naturalist*, 170(2):221–231, 2007.

[Web07] Geoffrey I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.

[Web08] Geoffrey I. Webb. Layered critical values: A powerful direct-adjustment approach to discovering significant patterns. *Machine Learning*, 71(2–3):307–323, 2008.

[WKRQ+07] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2007.

[WP06] Marjorie J. Wonham and Elizaveta Pachepsky. A null model of temporal trends in biological invasion records. *Ecology Letters*, 9(6):663–672, 2006.

[WY93] Peter H. Westfall and S. Stanley Young. *Resampling-based multiple testing: examples and methods for p-value adjustment*. Wiley, 1993.

[XSTK04] Hui Xiong, Shashi Shekhar, Pang-Ning Tan, and Vipin Kumar. Exploiting a support-based upper bound of Pearson's correlation coefficient for efficiently identifying strongly correlated pairs. In *KDD'04: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 334–343. ACM, 2004.

[XW05] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[YW09] Xiaowei Ying and Xintao Wu. Graph generation with prescribed feature constraints. In *SDM'09: Proceedings of the 9th SIAM International Conference on Data Mining*, pages 966–977. SIAM, 2009.

[ZMKL05] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic

diversification. In *WWW'05: Proceedings of the 14th International Conference on World Wide Web*, pages 22–32. ACM, 2005.

DISSERTATIONS IN INFORMATION AND COMPUTER SCIENCE

TKK-ICS-D13 Yuan, Zhijian.
Advances in Independent Component Analysis and Nonnegative Matrix Factorization. 2009.

TKK-ICS-D14 Hirsimäki, Teemu.
Advances in Unlimited-Vocabulary Speech Recognition for Morphologically Rich Languages. 2009.

TKK-ICS-D15 Heikinheimo, Hannes.
Extending Data Mining Techniques for Frequent Pattern Discovery: Trees, Low-Entropy Sets, and Crossmining. 2010.

TKK-ICS-D16 Hermelin, Miia.
Multidimensional Linear Cryptanalysis. 2010.

TKK-ICS-D17 Savia, Eerika.
Mutual Dependency-Based Modeling of Relevance in Co-Occurrence Data. 2010.

TKK-ICS-D18 Liitiäinen, Elia.
Advances in the Theory of Nearest Neighbor Distributions. 2010.

TKK-ICS-D19 Lahti, Leo.
Probabilistic Analysis of the Human Transcriptome with Side Information. 2010.

TKK-ICS-D20 Miche, Yoan.
Developing Fast Machine Learning Techniques with Applications to Steganalysis Problems. 2010.

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.

Data mining is an interdisciplinary research area that develops general methods for finding interesting and useful knowledge from large collections of data. This thesis addresses from the computational point of view the problem of assessing whether the obtained data mining results are merely random artefacts in the data or something more interesting. In randomization based significance testing, a result is compared with the results obtained on randomized data. The randomized data are assumed to share some basic properties with the original data. To apply the randomization approach, the first step is to define these properties. The next step is to develop algorithms that can produce such randomizations. Results on the real data that clearly differ from the results on the randomized data are not directly explained by the studied properties of the data. In this thesis, new randomization methods are developed for four specific data mining scenarios. The randomization methods introduced in this thesis are useful in various data mining applications.

ISBN 978-952-60-4323-4 (pdf)
ISBN 978-952-60-4322-7
ISSN-L 1799-4934
ISSN 1799-4942 (pdf)
ISSN 1799-4934

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 99/2011

Markus Ojala

Randomization Algorithms for Assessing the Significance of Data Mining Results

Aalto University