

Wilfried Grossmann  
Stefanie Rinderle-Ma

Fundamentals of  
**Business  
Intelligence**

# **Data-Centric Systems and Applications**

---

## **Series Editors**

M.J. Carey  
S. Ceri

## *Editorial Board*

A. Ailamaki  
S. Babu  
P. Bernstein  
J.C. Freytag  
A. Halevy  
J. Han  
D. Kossmann  
I. Manolescu  
G. Weikum  
K.-Y. Whang  
J.X. Yu

More information about this series at  
<http://www.springer.com/series/5258>

Wilfried Grossmann • Stefanie Rinderle-Ma

# Fundamentals of Business Intelligence



Springer

Wilfried Grossmann  
University of Vienna  
Vienna  
Austria

Stefanie Rinderle-Ma  
University of Vienna  
Vienna  
Austria

ISSN 2197-9723

ISSN 2197-974X (electronic)

Data-Centric Systems and Applications

ISBN 978-3-662-46530-1

ISBN 978-3-662-46531-8 (eBook)

DOI 10.1007/978-3-662-46531-8

Library of Congress Control Number: 2015938180

Springer Heidelberg New York Dordrecht London

© Springer-Verlag Berlin Heidelberg 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer-Verlag GmbH Berlin Heidelberg is part of Springer Science+Business Media  
([www.springer.com](http://www.springer.com))

# Foreword

Intelligent businesses need Business Intelligence (BI). They need it for recognizing, analyzing, modeling, structuring, and optimizing business processes. They need it, moreover, for making sense of massive amounts of unstructured data in order to support and improve highly sensible—if not highly critical—business decisions. The term “intelligent businesses” does not merely refer to commercial companies but also to (hopefully) intelligent governments, intelligently managed educational institutions, efficient hospitals, and so on. Every complex business activity can profit from BI.

BI has become a mainstream technology and is—according to most information technology analysts—looking forward to a more brilliant and prosperous future. Almost all medium and large-sized enterprises and organizations are either already using BI software or plan to make use of it in the next few years. There is thus a rapidly growing need of BI specialists. The need of experts in machine learning and data analytics is notorious. Because these disciplines are central to the Big Data hype, and because Google, Facebook, and other companies seem to offer an infinite number of jobs in these areas, students resolutely require more courses in machine learning and data analytics. Many Computer Science Departments have consequently strengthened their curricula with respect to these areas.

However, machine learning, including data analytics, is only one part of BI technology. Before a “machine” can learn from data, one actually needs to collect the data and present them in a unified form, a process that is often referred to as data provisioning. This, in turn, requires extracting the data from the relevant business processes and possibly also from Web sources such as social networks, cleaning, transforming, and integrating them, and loading them into a data warehouse or other type of database. To make humans efficiently interact with various stages of these activities, methods and tools for data visualization are necessary. BI goes, moreover, much beyond plain data and aims to identify, model, and optimize the business processes of an enterprise. All these BI activities have been thoroughly investigated, and each has given rise to a number of monographs and textbooks. What was sorely missing, however, was a book that ties it all together and that gives a unified view of the various facets of Business Intelligence.

The present book by Wilfried Grossmann and Stefanie Rinderle-Ma brilliantly fills this gap. This book is a thoughtful introduction to the major relevant aspects of BI. The book is, however, not merely an entry point to the field. It develops the various subdisciplines of BI with the appropriate depth and covers the major methods and techniques in sufficient detail so as to enable the reader to apply them in a real-world business context. The book focuses, in particular, on the four major areas related to BI: (1) data modeling and data provisioning including data extraction, integration, and warehousing; (2) data and process visualization; (3) machine learning, data and text mining, and data analytics; and (4) process analysis, mining, and management. The book does not only cover the standard aspects of BI but also topics of more recent relevance such as social network analytics and topics of more specialized interest such as text mining. The authors have done an excellent job in selecting and combining all topics relevant to a modern approach to Business Intelligence and to present the corresponding concepts and methods within a unified framework. To the best of my knowledge, this is the first book that presents BI at this level of breadth, depth, and coherence.

The authors, Wilfried Grossmann and Stefanie Rinderle-Ma, joined to form an ideal team towards writing such a useful and comprehensive book about BI. They are both professors at the University of Vienna but have in addition gained substantial experience with corporate and institutional BI projects: Stefanie Rinderle-Ma more in the process management area and Wilfried Grossmann more in the field of data analytics. To the profit of the reader, they put their knowledge and experience together to develop a common language and a unified approach to BI. They are, moreover, experts in presenting material to students and have at the same time the real-life background necessary for selecting the truly relevant material. They were able to come up with appropriate and meaningful examples to illustrate the main concepts and methods. In fact, the four running examples in this book are grounded in both authors' rich project experience.

This book is suitable for graduate courses in a Computer Science or Information Systems curriculum. At the same time, it will be most valuable to data or software engineers who aim at learning about BI, in order to gain the ability to successfully deploy BI techniques in an enterprise or other business environment. I congratulate the authors on this well-written, timely, and very useful book, and I hope the reader enjoys it and profits from it as much as possible.



Georg Gottlob

Oxford, UK  
March 2015

# Preface

The main task of business intelligence (BI) is providing decision support for business activities based on empirical information. The term business is understood in a rather broad sense covering activities in different domain applications, for example, an enterprise, a university, or a hospital. In the context of the business under consideration, decision support can be at different levels ranging from the operational support for a specific business activity up to strategic support at the top level of an organization. Consequently, the term BI summarizes a huge set of models and analytical methods such as reporting, data warehousing, data mining, process mining, predictive analytics, organizational mining, or text mining.

In this book, we present fundamental ideas for a unified approach towards BI activities with an emphasis on analytical methods developed in the areas of process analysis and business analytics.

The general framework is developed in Chap. 1, which also gives an overview on the structure of the book. One underlying idea is that all kinds of business activities are understood as a process in time and the analysis of this process can emphasize different perspectives of the process. Three perspectives are distinguished: (1) the production perspective, which relates to the supplier of the business; (2) the customer perspective, which relates to users/consumers of the offered business; and (3) the organizational perspective, which considers issues such as operations in the production perspective or social networks in the customer perspective.

Core elements of BI are data about the business, which refer either to the description of the process or to instances of the process. These data may take different views on the process defined by the following structural characteristics: (1) an event view, which records detailed documentation of certain events; (2) a state view, which monitors the development of certain attributes of process instances over time; and (3) a cross-sectional view, which gives summary information of characteristic attributes for process instances recorded within a certain period of time.

The issues for which decision support is needed are often related to so-called key performance indicators (KPIs) and to the understanding of how they depend on certain influential factors, i.e., specificities of the business. For analytical purposes,

it is necessary to reformulate a KPI in a number of analytical goals. These goals correspond to well-known methods of analysis and can be summarized under the headings business description goals, business prediction goals, and business understanding goals. Typical business description goals are reporting, segmentation (unsupervised learning), and the identification of interesting behavior. Business prediction goals encompass estimation and classification and are known as supervised learning in the context of machine learning. Business understanding goals support stakeholders in understanding their business processes and may consist in process identification and process analysis.

Based on this framework, we develop a method format for BI activities oriented towards ideas of the L\* format for process mining and CRISP for business analytics. The main tasks of the format are the business and data understanding task, the data task, the modeling task, the analysis task, and the evaluation and reporting task. These tasks define the structure of the following chapters.

Chapter 2 deals with questions of modeling. A broad range of models occur in BI corresponding to the different business perspectives, a number of possible views on the processes, and manifold analysis goals. Starting from possible ways of understanding the term model, the most frequently used model structures in BI are identified, such as logic-algebraic structures, graph structures, and probabilistic/statistical structures. Each structure is described in terms of its basic properties and notation as well as algorithmic techniques for solving questions within these structures. Background knowledge is assumed about these structures at the level of introductory courses in programs for applied computer science. Additionally, basic considerations about data generation, data quality, and handling temporal aspects are presented.

Chapter 3 elaborates on the data provisioning process, ranging from data collection and extraction to a solid description of concepts and methods for transforming data into analytical data formats necessary for using the data as input for the models in the analysis. The analytical data formats also cover temporal data as used in process analysis.

In Chap. 4, we present basic methods for data description and data visualization that are used in the business and data understanding task as well as in the evaluation and reporting task. Methods for process-oriented data and cross-sectional data are considered. Based on these fundamental techniques, we sketch aspects of interactive and dynamic visualization and reporting.

Chapters 5–8 explain different analytical techniques used for the main analysis goals of supervised learning (prediction and classification), unsupervised learning (clustering), as well as process identification and process analysis. Each chapter is organized in such a way that we first present first an overview of the used terminology and general methodological considerations. Thereafter, frequently used analytical techniques are discussed.

Chapter 5 is devoted to analysis techniques for cross-sectional data, basically traditional data mining techniques. For prediction, different regression techniques are presented. For classification, we consider techniques based on statistical principles, techniques based on trees, and support vector machines. For unsupervised

learning, we consider hierarchical clustering, partitioning methods, and model-based clustering.

Chapter 6 focuses on analysis techniques for data with temporal structure. We start with probabilistic-oriented models in particular, Markov chains and regression-based techniques (event history analysis). The remainder of the chapter considers analysis techniques useful for detecting interesting behavior in processes such as association analysis, sequence mining, and episode mining.

Chapter 7 treats methods for process identification, process performance management, process mining, and process compliance. In Chap. 8, various analysis techniques for problems are elaborated, which look at a business process from different perspectives. The basics of social network analysis, organizational mining, decision point analysis, and text mining are presented. The analysis of these problems combines techniques from the previous chapters.

For explanation of a method, we use demonstration examples on the one hand and more realistic examples based on use cases on the other hand. The latter include the areas of medical applications, higher education, and customer relationship management. These use cases are introduced in Chap. 1. For software solutions, we focus on open source software, mainly R for cross-sectional analysis and ProM for process analysis. A detailed code for the solutions together with instructions on how to install the software can be found on the accompanying website:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

The presentation tries to avoid too much mathematical formalism. For the derivation of properties of various algorithms, we refer to the corresponding literature. Throughout the text, you will find different types of boxes. Light grey boxes are used for the presentation of the use cases, dark grey boxes for templates that outline the main activities in the different tasks, and white boxes for overview summaries of important facts and basic structures of procedures.

The material presented in the book was used by the authors in a 4-h course on Business Intelligence running for two semesters. In case of shorter courses, one could start with Chaps. 1 and 2, followed by selected topics of Chaps. 3, 5, and 7.

Vienna, Austria  
Vienna, Austria

Wilfried Grossmann  
Stefanie Rinderle-Ma



# Acknowledgements

We thank the following persons for their support and contributions to the book: Reinhold Dunkl for providing details on the EBMC<sup>2</sup> project, Simone Kriglstein for the support on the example presented in Fig. 4.3, Hans-Georg Fill for the discussions and support on ontologies, Jürgen Mangler for his help with the HEP data set, Fengchuan Fan for support on dynamic visualization, Karl-Anton Fröschl for the inspiring discussions, and Manuel Gatterer for checking the language.

Our greatest gratitude goes to our families for their unconditional support.



# Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Definition of Business Intelligence .....	1
1.2	Putting Business Intelligence into Context .....	4
1.2.1	Business Intelligence Scenarios .....	4
1.2.2	Perspectives in Business Intelligence .....	6
1.2.3	Business Intelligence Views on Business Processes .....	8
1.2.4	Goals of Business Intelligence .....	11
1.2.5	Summary: Putting Business Intelligence in Context .....	13
1.3	Business Intelligence: Tasks and Analysis Formats .....	14
1.3.1	Data Task .....	14
1.3.2	Business and Data Understanding Task .....	15
1.3.3	Modeling Task .....	17
1.3.4	Analysis Task .....	19
1.3.5	Evaluation and Reporting Task .....	20
1.3.6	Analysis Formats .....	20
1.3.7	Summary: Tasks and Analysis Formats .....	24
1.4	Use Cases .....	24
1.4.1	Application in Patient Treatment .....	25
1.4.2	Application in Higher Education .....	28
1.4.3	Application in Logistics .....	29
1.4.4	Application in Customer Relationship Management .....	30
1.5	Structure and Outline of the Book .....	31
1.6	Recommended Reading (Selection) .....	32
	References .....	32
<b>2</b>	<b>Modeling in Business Intelligence .....</b>	<b>35</b>
2.1	Models and Modeling in Business Intelligence .....	35
2.1.1	The Representation Function of Models .....	36
2.1.2	Model Presentation .....	39
2.1.3	Model Building .....	41

2.1.4	Model Assessment and Quality of Models.....	44
2.1.5	Models and Patterns.....	45
2.1.6	Summary: Models and Modeling in Business Intelligence .....	46
2.2	Logical and Algebraic Structures.....	46
2.2.1	Logical Structures .....	46
2.2.2	Modeling Using Logical Structures .....	48
2.2.3	Summary: Logical Structures.....	51
2.3	Graph Structures .....	51
2.3.1	Model Structure .....	51
2.3.2	Modeling with Graph Structures .....	54
2.3.3	Summary: Graph Structures .....	57
2.4	Analytical Structures .....	58
2.4.1	Calculus.....	58
2.4.2	Probabilistic Structures .....	61
2.4.3	Statistical Structures .....	67
2.4.4	Modeling Methods Using Analytical Structures .....	70
2.4.5	Summary: Analytical Structures.....	73
2.5	Models and Data.....	74
2.5.1	Data Generation .....	74
2.5.2	The Role of Time.....	76
2.5.3	Data Quality .....	78
2.5.4	Summary: Models and Data .....	82
2.6	Conclusion and Lessons Learned.....	82
2.7	Recommended Reading (Selection) .....	83
	References .....	83
<b>3</b>	<b>Data Provisioning .....</b>	<b>87</b>
3.1	Introduction and Goals .....	87
3.2	Data Collection and Description .....	88
3.3	Data Extraction .....	90
3.3.1	Extraction-Transformation-Load (ETL) Process .....	90
3.3.2	Big Data .....	93
3.3.3	Summary on Data Extraction .....	98
3.4	From Transactional Data Towards Analytical Data.....	98
3.4.1	Table Formats and Online Analytical Processing (OLAP) ...	100
3.4.2	Log Formats .....	104
3.4.3	Summary: From Transactional Towards Analytical Data .....	108
3.5	Schema and Data Integration .....	108
3.5.1	Schema Integration .....	108
3.5.2	Data Integration and Data Quality .....	112
3.5.3	Linked Data and Data Mashups .....	113
3.5.4	Summary: Schema and Data Integration .....	114
3.6	Conclusion and Lessons Learned .....	115

3.7	Recommended Reading .....	115
References .....		115
<b>4</b>	<b>Data Description and Visualization .....</b>	119
4.1	Introduction .....	119
4.2	Description and Visualization of Business Processes .....	120
4.2.1	Process Modeling and Layout .....	121
4.2.2	The BPM Tools' Perspective .....	122
4.2.3	Process Runtime Visualization .....	123
4.2.4	Visualization of Further Aspects .....	123
4.2.5	Challenges in Visualizing Process-Related Information .....	126
4.2.6	Summary: Description and Visualization of Business Processes .....	127
4.3	Description and Visualization of Data in the Customer Perspective .....	127
4.3.1	Principles for Description and Visualization of Collections of Process Instances .....	127
4.3.2	Interactive and Dynamic Visualization .....	131
4.3.3	Summary: Visualization of Process Instances .....	133
4.4	Basic Visualization Techniques .....	133
4.4.1	Description and Visualization of Qualitative Information .....	134
4.4.2	Description and Visualization of Quantitative Variables .....	137
4.4.3	Description and Visualization of Relationships .....	140
4.4.4	Description and Visualization of Temporal Data .....	143
4.4.5	Interactive and Dynamic Visualization .....	145
4.4.6	Summary: Basic Visualization Techniques .....	146
4.5	Reporting .....	147
4.5.1	Description and Visualization of Metadata .....	147
4.5.2	High-Level Reporting .....	149
4.5.3	Infographics .....	151
4.5.4	Summary: Reporting .....	152
4.6	Recommended Reading .....	153
References .....		153
<b>5</b>	<b>Data Mining for Cross-Sectional Data .....</b>	155
5.1	Introduction to Supervised Learning .....	155
5.2	Regression Models .....	159
5.2.1	Model Formulation and Terminology .....	159
5.2.2	Linear Regression .....	161
5.2.3	Neural Networks .....	166
5.2.4	Kernel Estimates .....	169
5.2.5	Smoothing Splines .....	171
5.2.6	Summary: Regression Models .....	172

<b>5.3</b>	<b>Classification Models</b>	173
5.3.1	Model Formulation and Terminology	173
5.3.2	Classification Based on Probabilistic Structures	177
5.3.3	Methods Using Trees	182
5.3.4	K-Nearest-Neighbor Classification	185
5.3.5	Support Vector Machines	186
5.3.6	Combination Methods	190
5.3.7	Application of Classification Methods	191
5.3.8	Summary: Classification Models	192
<b>5.4</b>	<b>Unsupervised Learning</b>	193
5.4.1	Introduction and Terminology	193
5.4.2	Hierarchical Clustering	195
5.4.3	Partitioning Methods	199
5.4.4	Model-Based Clustering	201
5.4.5	Summary: Unsupervised Learning	203
<b>5.5</b>	<b>Conclusion and Lessons Learned</b>	204
<b>5.6</b>	<b>Recommended Reading</b>	204
	<b>References</b>	205
<b>6</b>	<b>Data Mining for Temporal Data</b>	207
<b>6.1</b>	<b>Terminology and Approaches Towards Temporal Data Mining</b>	207
<b>6.2</b>	<b>Classification and Clustering of Time Sequences</b>	212
6.2.1	Segmentation and Classification Using Time Warping	214
6.2.2	Segmentation and Classification Using Response Features	217
6.2.3	Summary: Classification and Clustering of Time Sequences	220
<b>6.3</b>	<b>Time-to-Event Analysis</b>	220
<b>6.4</b>	<b>Analysis of Markov Chains</b>	224
6.4.1	Structural Analysis of Markov Chains	226
6.4.2	Cluster Analysis for Markov Chains	230
6.4.3	Generalization of the Basic Model	231
6.4.4	Summary: Analysis of Markov Chains	233
<b>6.5</b>	<b>Association Analysis</b>	233
<b>6.6</b>	<b>Sequence Mining</b>	237
<b>6.7</b>	<b>Episode Mining</b>	240
<b>6.8</b>	<b>Conclusion and Lessons Learned</b>	242
<b>6.9</b>	<b>Recommended Reading</b>	243
	<b>References</b>	244
<b>7</b>	<b>Process Analysis</b>	245
<b>7.1</b>	<b>Introduction and Terminology</b>	245
<b>7.2</b>	<b>Business Process Analysis and Simulation</b>	247
7.2.1	Static Analysis	248
7.2.2	Dynamic Analysis and Simulation	248

7.2.3	Optimization.....	251
7.2.4	Summary: Process Analysis and Simulation.....	252
7.3	Process Performance Management and Warehousing .....	252
7.3.1	Performance Management .....	252
7.3.2	Process Warehousing .....	253
7.3.3	Summary: Process Performance Management and Warehousing .....	255
7.4	Process Mining .....	255
7.4.1	Process Discovery .....	256
7.4.2	Change Mining .....	263
7.4.3	Conformance Checking.....	266
7.4.4	Summary: Process Mining.....	267
7.5	Business Process Compliance .....	268
7.5.1	Compliance Along the Process Life Cycle.....	268
7.5.2	Summary: Compliance Checking .....	270
7.6	Evaluation and Assessment .....	270
7.6.1	Process Mining .....	270
7.6.2	Compliance Checking.....	271
7.7	Conclusion and Lessons Learned.....	271
7.8	Recommended Reading .....	272
	References .....	272
8	<b>Analysis of Multiple Business Perspectives .....</b>	275
8.1	Introduction and Terminology .....	275
8.2	Social Network Analysis and Organizational Mining .....	277
8.2.1	Social Network Analysis.....	277
8.2.2	Organizational Aspect in Business Processes.....	282
8.2.3	Organizational Mining Techniques for Business Processes .....	284
8.2.4	Summary: Social Network Analysis and Organizational Mining .....	290
8.3	Decision Point Analysis.....	290
8.4	Text Mining .....	294
8.4.1	Introduction and Terminology .....	294
8.4.2	Data Preparation and Modeling .....	296
8.4.3	Descriptive Analysis for the Document Term Matrix .....	301
8.4.4	Analysis Techniques for a Corpus .....	303
8.4.5	Further Aspects of Text Mining .....	307
8.4.6	Summary: Text Mining .....	313
8.5	Conclusion and Lessons Learned.....	313
8.6	Recommended Reading .....	315
	References .....	315
9	<b>Summary .....</b>	319

<b>A Survey on Business Intelligence Tools .....</b>	329
A.1 Data Modeling and ETL Support .....	329
A.2 Big Data .....	330
A.3 Visualization, Visual Mining, and Reporting .....	334
A.4 Data Mining .....	337
A.5 Process Mining .....	338
A.6 Text Mining .....	339
References .....	340
<b>Index .....</b>	343

# Chapter 1

## Introduction

**Abstract** In this chapter, we provide definitions of Business Intelligence (BI) and outline the development of BI over time, particularly carving out current questions of BI. Different scenarios of BI applications are considered and business perspectives and views of BI on the business process are identified. Further, the goals and tasks of BI are discussed from a management and analysis point of view and a method format for BI applications is proposed. This format also gives an outline of the book's contents. Finally, examples from different domain areas are introduced which are used for demonstration in later chapters of the book.

### 1.1 Definition of Business Intelligence

If one looks for a definition of the term Business Intelligence (BI) one will find the first reference already in 1958 in a paper of H.P. Luhn (cf. [14]). Starting from the definition of the terms “Intelligence” as “the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal” and “Business” as “a collection of activities carried on for whatever purpose, be it science, technology, commerce, industry, law, government, defense, et cetera”, he specifies a business intelligence system as “[an] automatic system [that] is being developed to disseminate information to the various sections of any industrial, scientific or government organization.” The main task of Luhn’s system was automatic abstracting of documents and delivering this information to appropriate so-called *action points*.

This definition did not come into effect for 30 years, and in 1989 Howard Dresner coined the term Business Intelligence (BI) again. He introduced it as an umbrella term for a set of concepts and methods to improve business decision making, using systems based on facts. Many similar definitions have been given since. In Negash [18], important aspects of BI are emphasized by stating that “...business intelligence systems provide actionable information delivered at the right time, at the right location, and in the right form to assist decision makers.”

Today one can find many different definitions which show that at the top level the intention of BI has not changed so much. For example, in [20] BI is defined as “an integrated, company-specific, IT-based total approach for managerial decision

support” and Wikipedia coins the term BI as “a set of theories, methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information for business purposes.”

Summarizing the different definitions, BI can be characterized by the following features:

### Features of BI

- *Task of BI:* The main task of BI is providing decision support for specific goals defined in the context of business activities in different domain areas taking into account the organizational and institutional framework.
- *Foundation of BI:* BI decision support mainly relies on empirical information based on data. Besides this empirical background, BI also uses different types of knowledge and theories for information generation.
- *Realization of BI:* The decision support has to be realized as a system using the actual capabilities in information and communication technologies (ICT).
- *Delivery of BI:* A BI system has to deliver information at the right time to the right people in an appropriate form.

Corresponding to the development in ICT and availability of data, we can distinguish different epochs in BI. The prehistory of BI mainly runs under the heading decision support systems (DSS) and is documented, for example, in [19]. The review covers the era from the 1960s up to the beginning of the twenty-first century and considers theory development in computer science, optimization, and application domains, as well as systems development like model-driven DSS (planning models or simulation), data-driven DSS (from data bases up to OLAP systems), communication-driven DSS (collaboration networks), document-driven DSS (document retrieval and analysis), and knowledge-driven DSS (expert systems).

According to Howard Dresner's definition in 1989, the term BI became popular in the 1990s and was understood mostly as data-driven decision support closely connected to the development of data warehouses, the usage of online analytical processing (OLAP), and reporting tools. In parallel to the developments in the area of data management, other analysis tools such as data mining or predictive analytics became popular. Sometimes, these were summarized under the heading *business analytics*, and one got the impression that BI is a collection of a loosely related heterogeneous set of tools supporting different tasks within a business. Hence, it was necessary to consolidate the different lines of development and to focus again on the decision support perspective.

One influential approach putting the data warehouse into the center is the Kimball methodology (cf. [12]). This methodology defines a life cycle for data warehouse solutions with dimensional modeling as the core element. The design of appropriate technical architectures supports the realization of a data warehouse. Applications like reporting and analytical models provide decision makers with the necessary information.

The software life-cycle model as a framework for integration of different aspects of BI is used in [17]. Other approaches like CRISP [4] start from the analysis process in knowledge discovery from databases. Besides such conceptual ideas, one can also frequently find pragmatic definitions, for example, in [6] it is argued that BI should be divided into querying, reporting, OLAP, alert tools, and business analytics. In this definition; business analytics is a subset of BI based on statistics, prediction, and optimization. In the book, we will follow this idea and understand BI in such a broad sense.

In the last years, data availability and analysis capabilities have increased tremendously, and new research areas for BI have emerged. In [22], a number of topics are listed under the heading *Business Intelligence 2.0*. Looking at these topics from the perspective of the four main BI characteristics stated above, one can organize these new challenges as shown in the overview box.

### Actual Challenges of BI

- *Tasks of BI:* Nowadays we can find a well-structured understanding of the business logic in almost all domain areas. This new understanding has also led to a process-oriented conceptual view, which integrates workflow considerations and process mining into BI [23]. Another aspect is that new organizational structures like decentralized organizations want to apply decision support within their environment, and, hence, ideas from collective intelligence or crowd sourcing are applied in BI.
- *Foundations of BI:* Besides the traditional data warehouse, we also have to take into account data on the Web. Such data is often not well-structured, but only semistructured such as text data. The need to integrate different data useful for decision support in a coherent way has led to models for linking data in BI. In connection with such new data, the scope of analytical methods has broadened and new tools such as visual mining, text mining, opinion mining, or social network analysis have emerged.
- *Realization of BI systems:* Today's software architectures allow interesting new realizations of BI systems. From a user perspective, Software as a Service (SaaS) constitutes an interesting development for BI systems. From a computational point of view, we have to deal with large and complex data sets nowadays. Moreover, cloud computing and distributed computing are important concepts opening new opportunities for BI applications.
- *Delivery of BI:* Mobile devices offer a new dimension for delivering information to users in real-time. However, these developments have to take into account that quality of real-time information is a new challenge for BI.

Obviously, many of the mentioned new developments cover more than one aspect of the aforementioned BI characteristics, but this classification should support the understanding that the basic definition and characteristics of BI are still valid.

Due to the importance of BI for business applications, there is a big market, and many companies offer BI solutions. These vendors create a lot of terms and acronyms and propose integrated formats for BI applications, but precise and generally accepted definitions of terms are frequently missing in the BI context. For an overview on vendors and tools, we refer to [21].

## 1.2 Putting Business Intelligence into Context

In the previous section, we characterized BI and stated its goals in a rather general way. In order to make this more precise, we want to discuss first the connection between business and BI from a management point of view. An interesting reference in this context that is worth reading is [13].

We understand the term *business* in a rather broad sense, i.e., as “any kind of activities of an organization for delivering goods or services to consumers.” These organizations may be active in different application domains, for example, an enterprise, an administrative body, a hospital, or an educational institution such as a university. Besides the different application domains, we have to be aware that decision support is needed for businesses of different size and scope. By size we understand a classification of the organization with respect to criteria such as number of employees (e.g., SMEs or big enterprises), regional dispersion (from local up to global players), number of customers, or revenues. Scope refers to the number of activities of the organization for which we look for decision support. For example in business administration, we may be interested in decision support at the global level for the enterprise or at a specific functional level (e.g., production or marketing). In medical applications, our focus may be decision support for the treatment of a specific disease or for the management of a hospital. In the administrative context, we can look for decision support for efficient organization of services or for improving customer satisfaction with the services.

### 1.2.1 Business Intelligence Scenarios

For development of a general framework of such diverse problems, we will follow ideas as outlined in [13] which organize BI activities according to principles used in business enterprises. A management level, an organizational level, a functional analytical level, and levels for data organization and acquisition are distinguished, and the role of BI in connection with *business models* is discussed. As in the case of BI, there are many definitions of the term business model (cf. [1]), but for our purpose the following rather naive understanding seems sufficient: A *business*

*model reflects the strategy of an enterprise for creating value.* There are four different scenarios that link BI to the business context, ranging from rather simple applications of decision support for a specific problem up to BI as an essential part of strategic planning [13].

### BI Scenarios

1. *Business intelligence separated from strategic management:* In this case BI is mainly concerned with the achievement of short-term targets in a division of an organization, for example, a department of an enterprise or a clinic in a hospital. Typically, results of the BI application are more or less standardized reports for a dedicated part of the business.
2. *BI supports monitoring of strategy performance:* Such a BI application is motivated by overall strategic goals and formulated in accordance with these goals. Monitoring of the performance is done by defining measurable targets. A data warehouse allowing a unified view onto the business is usually a prerequisite for such an application scenario.
3. *BI feedback on strategy formulation:* This application goes one step beyond the previous strategy and aims at an evaluation of the performance using analytical methods. In the best case, such an application can be used for the optimization of a strategy. A typical end-product in this scenario may be a balanced scorecard.
4. *BI as strategic resource:* This strategy uses the information generated by BI not only for optimization but also as an essential input for the definition of the strategy at the management level. Typical examples are customer-based marketing or development of standard operation procedures for patient treatment.

Obviously, this classification depends on the size of the organization and the scope of the business under consideration. For example, a BI application at a university department may be used as feedback on strategy formulation at the level of the department but also as a tool for monitoring the performance at the university level.

At first glance, the third and fourth strategies seem to be favorable, but in general, we have to take into account specificities of the application, how many resources can be attributed to BI, and the availability of information. For large production-oriented enterprises, the third option may be a good choice, and in service-oriented businesses the fourth strategy has yielded many success stories. But sometimes decision problems occur ad hoc, are hard to formalize, and it is not clear whether implementation of a high-level strategy is worth it in the long run. Moreover, results of such ad hoc applications may lead to standardized new BI activities at a higher strategic level.

### 1.2.2 Perspectives in Business Intelligence

After the determination of the overall BI strategy, we have to think about the structure of business activities. The description of the structure is frequently done by formulating a *business process*. We understand the term business process as a *collection of related and structured activities necessary for delivering a certain good or service to customers together with possible response activities of customers*.

Note that most definitions of business processes such as [5] omit the last part of the definition. However, we think that understanding the customer as an active decision maker inside the business process is more suited for BI. In the book, generally speaking, we will take the position that all kinds of business activities are processes, which means that activities take place within a period of time and follow some rules such as the partial ordering or the exclusion of an activity under certain conditions. However, we have to be aware that, to some extent, the incorporation of customer activities into the business process limits the application of the idea that business activities resemble the structure of purely rule-based activities. Instead of such a mechanistic consideration of business processes, BI is more concerned with the empirical realization of business process defined by *process instances*. In order to scrutinize these instances, we introduce the following three *BI perspectives* for the business process.

#### Perspectives in BI

- *Production perspective*: This perspective considers decision support for answering questions such as what kind of products should be offered to the customers and how the production should be operated. This perspective plays an important role for product development and for internal organization of the business.
- *Customer perspective*: This perspective focuses on customer behavior and aims at understanding how customers perceive products or services and how they react to this offer. The customer perspective plays an essential role in service-oriented businesses.
- *Organizational perspective*: This perspective examines the organizational background of the business process. It may refer to the organizational background for the operations in connection with the production perspective or to the influence of social networks on customer behavior.

Obviously, such perspectives depend on the application domain, the size, and the scope of the business. Practical applications usually encompass all three perspectives, but for BI applications such a division is useful for choosing appropriate information and analysis models. To some extent, this division also reflects the historical development of models and analytical methods nowadays applied in BI.

The production perspective usually requires detailed information and data about the internal organization of the business. Typically, the organizational structure of enterprises is specified and maintained in terms of organizational models that consist of organizational entities such as roles, organizational units, and actors. The organizational units are typically linked by different relations. For BI applications, the following roles are of interest<sup>1</sup>

### Roles in Context of BI

- The first role is the *process owner*, defined as the entity setting the rules governing the process. Traditionally, the process owner is defined from a production perspective, but in service-oriented businesses customers also may be process owners. Think, for example, of patients who decide about their treatment.
- The next step is to identify the *process subjects* as the entities that identify the process instances. In most cases, these process subjects are defined by the customers, but specific products or networks of people involved in the business process are also possible candidates. One can understand the process subjects as the entities triggering the initialization of a process instance by some event. For example, a patient with a certain health problem shows up at the hospital triggering a certain treatment process.
- Besides the process subjects, other people or in organizations can generate events in the process as well. We will denote these entities as *process actors*, or short as *actors*. In business administration; the actors are usually the part of the organization responsible for the production of goods or services.

The customer perspective frequently needs a much simpler view on the internal processes, because customers are usually not aware of the internal organization and only react according to their personal view on the business. On the other hand, for understanding customer behavior we need a more detailed description of personal customer characteristics like sex, age, or social status, together with their organizational embedding into the business process.

An important issue is the interaction between production and customer perspective. This interaction may be rather simple, for example, when a customer decides to buy goods in a shop for some time and quits the business relation afterwards. Other processes may be rather complex and comprise many interactions like negotiations

---

<sup>1</sup>In the business process management literature, different roles or stakeholders within the process life cycle are mentioned. See, for example, Dumas et al. [8] or Weske [25]. In this book, we will explain a selection of these roles and augment them with the roles of the process subjects.

or the usage of multiple services. A typical example from health care is the treatment of patients. Depending on the complexity of the interaction, a specific combination of the different perspectives might be required.

In summary, business processes can be analyzed along three goal perspectives: customer, production, and organization. We aim at discussing BI using all three perspectives throughout the book. In Sect. 1.4, we will show by examples how these perspectives may be realized in various application contexts.

### 1.2.3 Business Intelligence Views on Business Processes

Structural analysis of business processes, i.e., the analysis based on a model describing the business process, is, in many cases, an interesting and useful task. However, as already mentioned, BI applications are more interested in understanding the real-world process behavior.

This real-world behavior is reflected by the execution of, possibly, a multitude of instances that are created, initiated, and executed according to an often not explicitly stated process model. As a consequence, we have to think about how to exploit empirical information collected during the execution of process instances.

Depending on the effort spent on data collection, a broad spectrum of data might be available about the execution of process instances. Ideally, there exists a log of *events*, observed and stored during the execution of a process instance. For each activity, the log of a process will record its beginning by a *start event*, the completion of the activity by an *end event*, and, if necessary, also *interruption* or *resumption events*. Additionally, the time of occurrence for all these events is known (usually reflected by a time stamp), including additional attributes characterizing the activity associated with the events. Necessary for subsequent analysis is an attribute that reflects the activity label or id. Further attributes include the outcome of the activity, the people involved in the activity (mostly working on the activity), the cost of the activity, and the resources required for activity execution.

This description of data collection resembles the idea of a fully automated process which is hardly realized in practice. In particular, this is the case if events are triggered by customers. Moreover, the use of all available data for decision making is not recommendable, because one may get lost in too much detail. Hence, in BI we need specific views on the data about the instances of the business process. The overview box summarizes the different views on business processes.

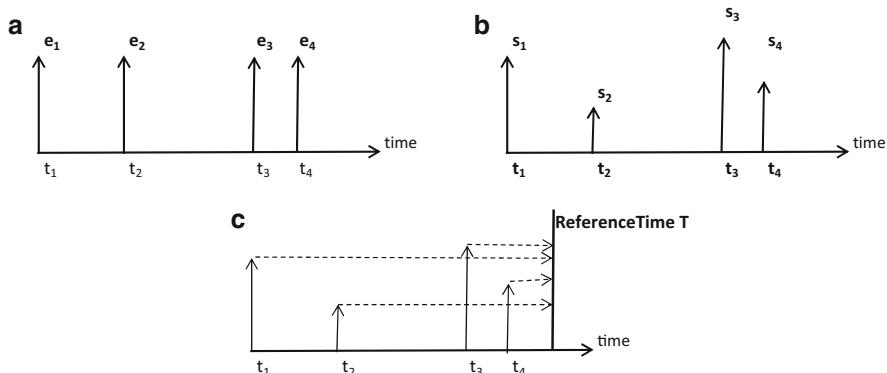
#### BI Views on Business Processes

- *Event view:* The main emphasis is on the events in the business process characterized by a time stamp for the start, a time stamp for the end, and, if necessary, also a time stamp for the resumption of the activity execution after an interruption.

- **State view:** Besides the occurrence of events the state view also considers the values of attributes, the so-called state variables, measured in connection with the events.
- **Cross-sectional view:** In this case, we investigate the history of many process instances at a certain reference time. Usually, this view considers information about events as well as the values of state variables and summarizes the information about process instances for decision making.

The event view puts the main emphasis on the rules defining the partial ordering of the business process events according to the production perspective. This ordering of the events defines the *control flow perspective* of the business process. Figure 1.1a outlines the recording of four events  $e_1, \dots, e_4$  at the corresponding time stamps  $t_1, \dots, t_4$  which defines a partial order between these events. Let us mention that, in some cases, it is rather difficult to exactly record the start and end events for the activity. In medical applications, for example, the start event for an illness is often hard to define and we have only information about the time of diagnosis.

The last remark leads us to the second view on the business process, which emphasizes the outcome of the process activities. These outcomes switch the focus from the business process to the corresponding process subjects. These subjects may be customers, delivered goods or services, or a network of business partners. The understanding of the behavior of these process subjects is based on measured quantities, so-called *state variables*. This notion suggests that the business process is treated as a dynamic system. Obviously, the values of the state variables change over time, either due to certain business process activities or due to some kind of inherent



**Fig. 1.1** BI views on business process instances. **(a)** Event view, **(b)** state view, **(c)** cross-sectional view

variability of the process subjects themselves. Think, for example, about the state of health of a person measured by some medical parameters or the change in quality of a product due to ageing. Understanding the instances of the business process as trajectories of such state variables will henceforth be called the *state view* on the business process. Figure 1.1b shows four states of one state variable,  $s_1, \dots, s_4$ , that are taken by the system at four points in time, i.e.,  $t_1, \dots, t_4$ . The different heights of the bars represent the different values of the state variable, e.g., if a patient is in the state *running a fever* the height of the corresponding bar would record temperature at a certain time. The state view is sometimes blurred with the event view, because usually we need a business process activity for recording or changing the values of the state variables. For example, in business applications, the activity of examination of a customer's account is necessary for obtaining the financial state of the customer. Event view and state view result in a data structure for observed instances which is frequently called *temporal data* or *time stamped data*.

The third view starts from the observation that in many applications our main interest is on the aggregated quantities of instances of business processes at a certain point in time. Such aggregations may be obtained from the event view or the state view. Concerning the event view, we may consider the counts of the instances of the different activities occurring within a certain observation period, the total processing time of these activities, or the consumed resources for these instances. In the case of the state view, the aggregation may refer to an average value of the state for each instance over the observation period or the actual state of the instances. We will call this view the *cross-sectional view* on the business process. The idea behind the cross-sectional view on business processes is depicted in Fig. 1.1c. Note that the view on the states is taken at some reference time  $T$ .

Event, state, and cross-sectional views on business processes can be used in combination with the three perspectives: customer, production, and organization as defined above. In the production perspective, one can put the main emphasis on the sequence of events corresponding to activities for the production of goods or services. Another possibility is to focus on state variables describing the production (e.g., the utilization of resources over a period of time), or one can look at the summary of characteristics for production in a certain period of time. In the case of the customer perspective, either the cross-sectional or state view is dominant. Events play an important role in connection with the classification of customers. A credit default, for example, is an event that defines two classes of customers. In medical applications, the state view describing the behavior of patients with respect to a certain parameter is often of main interest. In the organizational perspective, a cross-sectional view is important for many applications, in particular for the analysis of networks of customers, but the organizational structure behind specific events may also be of interest.

### 1.2.4 Goals of Business Intelligence

The starting point for BI applications are analysis goals. Such goals range from the acquisition of information about some aspects of the business process over improving the performance of the process up to understanding the implications of the process for achieving strategic goals. The goals can be formulated in two different ways. The first one is based on so-called *key performance indicators (KPIs)*. KPIs allow measuring the performance of the business with respect to some goals in any perspective of the business. In the economic context, a key performance indicator may refer to the acquisition of new customers or the improvement of the produced goods or services measured with respect to customer satisfaction. A systematic overview and classifications of KPIs can be found in [13, 21]. In the educational context, KPIs may be the drop-out of students, the costs per degree, or measurements showing the position of the institution in the community. In medical applications, KPIs may refer to the efficiency of the treatment process or to the well-being of patients. Besides such quantitative indicators, one can also define other types of indicators that are more difficult to measure. Identification of KPIs is based on the identification of a predefined business process, on the definition of requirements for the business process, and on a measurement of the results of the business in comparison with set goals. In Sect. 1.4, we will define a number of KPIs for the use cases.

#### KPIs as Goals for BI

- *Key performance indicator:* A KPI links the activities of the business to objectives by defining a measurable quantity. KPIs may refer to some aspects of the performance of the business process or to the business as a whole. One can distinguish between quantitative indicators presented as numbers, practical indicators interfacing with processes, directional indicators showing whether the organization is getting better or not, actionable indicators for controlling effects of change, or financial indicators.
- *Influential factors:* Attributes that may influence the behavior of the KPI in any BI perspective.

KPIs describe the business at a top level. However, many a time BI goes one step beyond goal formulation and measurement of performance and aims to understand how business performance can be affected by so-called *influential factors* that determine and explain the values of KPIs. The relation between KPIs and influential factors can be used later on for decision support. For example, in the case of customer relationship, we are not only interested in the number of

customers who quit the business relation but want to understand the reasons for their behavior. Possible influential factors have to be investigated from all three business perspectives. The production view of the business, for example, looks at influential factors in connection with the production of goods or services. From the customer perspective, possible influential factors are frequently based on customer attributes defining sociodemographic characteristics and attributes referring to the perception of the offered products or services. The organizational perspective can help in the identification of influential factors connected with the internal organization of the business or the influence of social networks on customer behavior.

For understanding the relation between KPIs and influential factors, we use a second formulation of goals in BI called *analytical goals*. This formulation is based on a typology of the questions with respect to possible approaches in the analysis. One can distinguish three broad types of analytical goals summarized in the overview box. The first type, the descriptive goals, occurs in all perspectives and can be based on all three views on the business process. The basic descriptive goal is reporting which is frequently a supplementary goal for achieving other analytical goals. The descriptive goals segmentation and detection of interesting behavior are frequently summarized under the heading *unsupervised learning*. Predictive goals are more ambitious; they are of main interest in the case of the customer perspective using the cross-sectional view. For predictive goals, the term *supervised learning* is frequently used. Even more ambitious are understanding goals, which are usually closely related to the production perspective using either the event view or the state view. In the subsequent chapters, we will cover various models and analysis methods for these analytical goals from the different business perspectives.

### Typology of Analytical Goals

- *Descriptive goals* generate a summary description for the instances of the business process from the different BI perspectives. Three main goals can be summarized under this heading:
  1. *Reporting*: Summarize the instances in such a way that one can use the information for decisions.
  2. *Segmentation*: Group the instances according to a similarity measure and find representative instances for these groups.
  3. *Detect interesting behavior*: Identify events during business process execution that allow the identification of important aspects of the process.
- *Predictive goals* predict the behavior of instances of the business process. Two different kinds of prediction may be distinguished:
  1. *Regression*: Find a function that allows the prediction of the output (usually a KPI) from a number of input variables (influential factors).
  2. *Classification*: Given a partition for observed instances into disjoint classes, assign a new instance to one of the classes.

- *Understanding goals* support stakeholders in understanding their business processes. Two main goals can be formulated:
  1. *Process identification:* Identify the rules that determine the relationships between the events of the process.
  2. *Process analysis:* Investigate the performance of the instances with respect to their conformance with a defined business process.

Note that the goal orientation is complementary to the life-cycle analysis of the business process along three phases: design time, run-time, and change time. At *design time*, the business process is described in terms of process models (cf. Sect. 2.3.2) defined in agreement with a KPI for certain business needs. Many a time, the background for the design is formulated as a *business plan*, and detailed formulation requires the investigation of different analytical goals. For example, if we are interested in launching a new product, analytical goals like prediction of market opportunities or detailed product description have to be achieved. The analysis of the process at *runtime* refers to data from process execution, i.e., data from process instances. Also in this case the analysis requires a precise formulation of analytical goals. Similarly, the analysis at *change time* corresponds to a specific formulation of a KPI. However, attention should be paid to the fact that traditionally this analysis along the lifetime of a process is mainly understood in connection with the production perspective. Our goal-oriented formulation of analytical goals seems more open to the other perspectives.

### 1.2.5 Summary: Putting Business Intelligence in Context

For the development of a unified umbrella for BI, we use a process-oriented definition of the term business applicable in many different domains. One can look at such a business process from different perspectives, in particular the production perspective, the customer perspective, and the organizational perspective are identified. In connection with the perspective, it is often important to identify the roles of actors within the business process; in particular, process subjects as the actors that generated instances of the business are of utmost importance in BI.

The main input for all BI activities are data about the instances of business processes. These data are generated according to a specific view on the business process. Three views are identified: the event view, the state view, and the cross-sectional view. In the production perspective, the event view is of utmost importance and in the customer perspective the cross-sectional view is dominant.

Using data as input, any BI activity starts from a certain goal. For the goal measurable quantities, so-called key performance indicators (KPIs) are defined. The KPIs have to be seen in connection with the strategic use of BI inside the business. This strategic use ranges from application of BI for achieving short-term targets with no connection to the management strategy over use of BI as a feedback for the overall management strategy up to understanding BI as a strategic resource for management decisions.

Many a time, BI applications aim for understanding the dependence of a KPI from other quantities called influential factors. This leads to the formulation of analytical goals for BI. Different analytical goals can be identified: descriptive goals, predictive goals, and business understanding goals. These analytical goals allow a formal analysis, and the results of the analysis can be used later on for decision support.

## 1.3 Business Intelligence: Tasks and Analysis Formats

Achieving the different analytical goals requires the completion of a number of tasks, including an analysis format for the execution of these tasks. In this section, we briefly describe the tasks and propose an analysis format.

### 1.3.1 Data Task

The data task is a prerequisite for all BI activities. The main goal is organization of available information about the business and its environment. Typically, the information are data about the structural properties of the enterprise and the registered customers, the transactional data from business process instances, the data describing production activities, or traces of activities in social networks. These data are collected under different data-capturing regimes and stored in different data sources using multifarious structures ranging from data with diverse temporal and spatial granularity up to semistructured text data. The major challenge is organizing the data in such a way that they can be utilized in various BI activities.

Many a time, one can start with an existing organization of the data in a data warehouse, which offers coherent data of high quality and thus supports diverse BI activities, in particular standard reporting. This is the reason why BI is often understood as an endeavor of data modeling and retrieval. However, due to the changes of the business and its environment over time, even a well-designed data warehouse cannot answer all questions. Decision support for new challenges may require a reorganization of the data or collecting additional data for special purposes. Consequently, it is necessary to have knowledge about the methods for data collection and for the augmentation of existing data with new data.

The data task relies on *data modeling techniques* encompassing different data models like ER models, UML, or semistructured data models, including methods on how to apply the models, and an IT infrastructure for data provisioning. Chapter 3 discusses topics of data provision and introduces analytical data formats useful for the different business perspectives. Moreover, issues of data integration are discussed in Chap. 3.

### 1.3.2 Business and Data Understanding Task

The starting point of business and data understanding is an initial formulation of a goal, in the best case formulated as KPIs. The business and data understanding task considers the business regarding this intended goal and develops first ideas about what part of the business is of interest in connection with the goal and what data from the repository can be used. The results of the task are a formulation of analytical goals, an excerpt of the overall business relevant for the analytical goal, and data needed for achieving the analytical goal. Moreover, a first outline of the work schedule for further activities in the BI project is defined. This needs a number of interrelated activities that are summarized in the overview box.

The first two activities are more oriented towards business understanding and specify the application environment and the business perspectives of interest. Next we have to decide about the view on the business and the data as a first step in data understanding. In BI, data traces of past instances of the business process are the main source of knowledge used for analyzing the business with respect to the goals. Although BI often has an exploratory nature, some prestructuring according to domain knowledge is necessary. Consider a medical treatment process as an illustrative example:

It is not reasonable to use all possible parameters informing about the health status of a patient for monitoring a specific treatment process. Instead, a number of potential influential factors are selected according to expert knowledge, belief, and interest. In other words, knowledge about the process is mapped to a number of variables also taking into account factors that are probably not part of the established knowledge.

#### Issues in Business and Data Understanding

- *Application environment:* This topic explores the size and the scope of the analysis goal within the overall business and determines the BI scenario (Sect. 1.2.1) for the application. Furthermore, the resources and time horizon of the project are determined.
- *Business perspective:* This point covers the investigation of the analysis goal from the different business perspectives comprising the identification of process owners, process subjects, and actors in the business process (Sect. 1.2.2).

- *BI views:* That part of existing data relevant for the goal is identified and an appropriate view on the business and the data of interest for the analysis is specified (Sect. 1.2.3).
- *Analytical goals:* A precise definition of the envisaged KPIs and influential factors is given and the intended types of analytical goals (Sect. 1.2.4) are formulated.
- *Assessment of data:* Screening of data with respect to properties of the variables and data quality. Furthermore, a number of data transformations may be necessary for editing the data in such a way that they can be used in the envisaged models necessary for achieving the analytical goal.

The choice of view on the business depends not only on knowledge and goal but also on the availability of the data. Many a time, data in the cross-sectional view can be easily accessed, whereas detailed data about process instances in the event view are only available for parts of the process. In the cases where data in the desired view are not available, one has to decide whether the use of existing data is feasible for the goal. In the worst case, it may happen that the availability of the data is the limiting factor and a new data collection may be necessary.

Using the information gathered up to now, one can give a precise formulation of the goal in terms of analytical goals. This requires a combination of domain knowledge, i.e., business understanding, and knowledge about properties of the data, i.e., data understanding.

Besides the general considerations about data, the feasibility of the envisaged analysis depends often on data peculiarities. These peculiarities are found in data assessment. Investigation of properties of the data using data description and visualization techniques gives quantitative information about individual variables together with fundamental relations between the variables. Such an analysis supports often the selection of possible influential factors. Issues of data quality refer to data generation, for example, the completeness of data or the coherence of data from different sources. These aspects will be discussed in Chaps. 2 and 3.

Often data assessment includes data transformations for obtaining the data in a form needed as input for the modeling and analysis task. One type of transformations are those necessary for obtaining a unified view on the data of the business process. For example, if some of the influential factors are recorded as data in the event view and others only in the cross-sectional view, we have to transform the data in such a way that we can use them in one model. Another type of transformation is the computation of new variables out of existing ones, for example, scores.

The *business and data understanding task* uses *business understanding techniques* and *data understanding techniques*. Business understanding techniques answer the questions about the application environment and require domain knowledge about the business and experience in project management. In the literature about project management, one can find techniques for structuring this process,

but an open mind, discussions of the problem from different perspectives, and experience are probably the most important requirements. With respect to data understanding, one can rely on techniques for data description and data visualization, which will be treated in Chap. 4. We use the term technique to emphasize that we have to rely on models for the description and visualization, methods for using such models, as well as tools that support the realization of the task.

### 1.3.3 Modeling Task

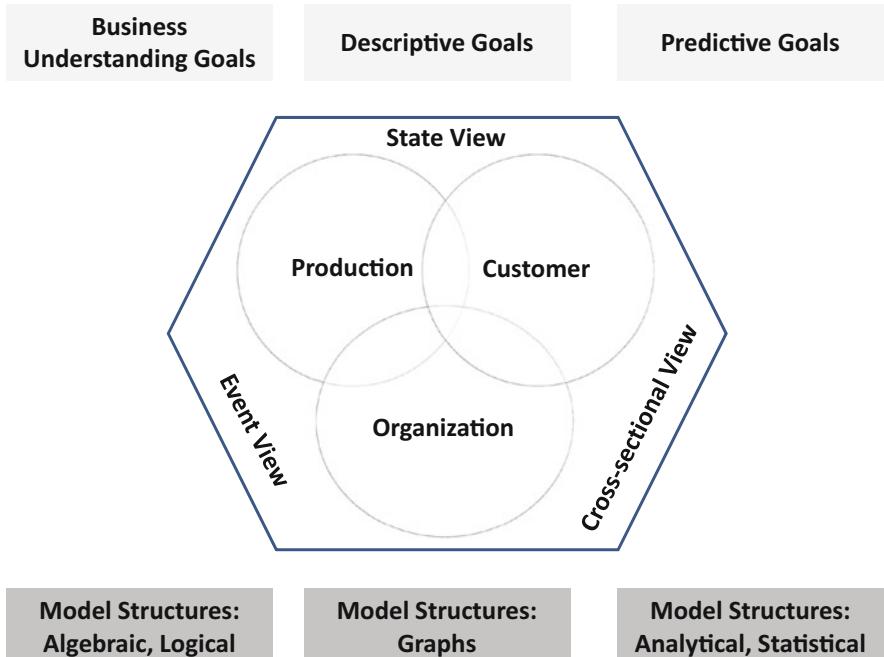
The modeling task aims at setting up an *analytical business model*, i.e., a formal model that allows precise answers for the analytical goals. Depending on the BI perspectives, views, and goals of interest, we use formal structures to build a model that enables the transformation of the analytical goals into formal questions about the properties of a model. Sometimes, the model may be rather simple and is not more than a specific query on the available data. At other times, choosing a model may be rather intricate and is by no means evident. Consider, for example, the model formulation for analyzing a KPI in connection with customer acquisition:

- One can start with a marketing-oriented approach, take the customer perspective, and identify factors that attract new customers. After the identification of these factors, one can think about the necessary internal processes meeting the customer requirements.
- Another approach is to start with the production perspective, scrutinize the production process, develop possible scenarios for changes of the production processes. Afterwards, the different scenarios are analyzed with respect to the attractiveness for customers.

This shows that different model formulations are possible depending on the business perspective and the formulation of the analytical goal. Figure 1.2 illustrates the interrelations between goals, perspectives, views, and analytical models.

The circles in the center of the hexagon represent the different BI perspectives. The intersection of the circles illustrate that we often have to cope with analytical goals that need multiple BI perspectives. The inner labels at the sides of the hexagon describe the BI views on the perspectives, for example, taking an event view to analyze the production perspective or a combination of production and organizational perspective. For example, one may take the event view to analyze the production perspective or a combination of production and organizational perspective.

Above the hexagon, we denote the BI goals, i.e., understanding goals, descriptive goals, or predictive goals as discussed in Sect. 1.2.2. The lower part of Fig. 1.2 introduces the formal structures used for transforming BI goals into properties of the model. Basically, one can distinguish between models with an algebraic



**Fig. 1.2** Overview on modeling activities

and a logical structure, models with a graph structure, and models with analytical structure. Among the algebraic-logic structures, business process models are the most prominent ones. Among the analytical structures, probability and statistics are most important and frequently used. Graph structures combine analytical and algebraic elements and play an important role in BI modeling.

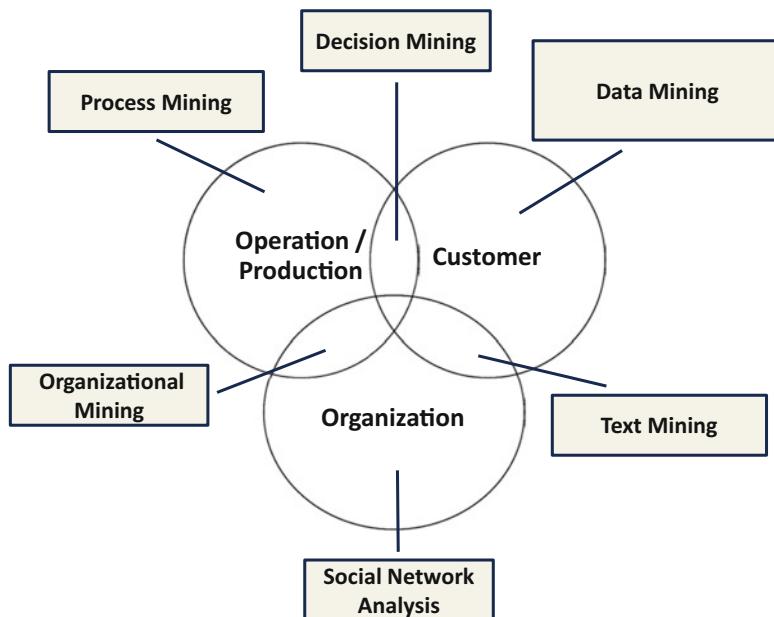
Corresponding to their many facets, successful BI applications rely on a rather large repository of possible *modeling techniques*. As we will discuss in Chap. 2, a modeling technique is based on a model structure, a method of using these structures, and tools supporting the formulation of a model. Generally speaking, Fig. 1.2 can help in structuring a model repository in such a way that one can define or find a model that fits into the different perspectives and views and tackles a certain goal. It can also help to think about frequently used approaches in different scenarios. However, leaving the standard path is always an option worth pursuing, provided available data allow for doing so.

Besides modeling techniques, the modeling task requires specific data preparation techniques. For example, in the case of analysis goals referring to text data, different techniques can be used for transforming such unstructured data to structured data which allow the application of algorithms. Such transformations will be considered in connection with the analysis methods.

### 1.3.4 Analysis Task

Having defined a model, one needs algorithms to compute a solution for the analytical goal within the model. In BI, these algorithms are usually denoted by the term *mining*, stressing that we are searching for a solution concerning a frequently not very well-defined problem. Another frequently used term is *Machine Learning* which has its origin in Artificial Intelligence and was formally defined in [16] as computer programs with the ability to learn to solve a task. Learning is understood as improving the performance of a program using experience from past executions. This experience is obtained from examples but the number of examples is not necessarily large. The term data mining is originally defined as the analysis step in the process of the knowledge discovery in data bases which has a more exploratory nature. Today, the two terms are often used synonymously, but in BI applications mining is predominant and we will use it throughout this book.

Different types of mining have been proposed; Fig. 1.3 shows an overview on these types in connection with the different BI perspectives. Note that this includes mining algorithms that frequently occur in connection with overlapping BI perspectives. At the intersection of the perspectives production and customer, for example, decision mining has been suggested in the literature.



**Fig. 1.3** Overview on mining algorithms along the BI perspectives

Besides the BI perspective, the choice of the algorithms depends on the view and the envisaged analytical goal. This combination provides us with an organization of the book chapters that deal with the analysis task in BI, i.e., Chaps. 5–8. Chapter 5 is devoted to data mining algorithms and tools for the cross-sectional view on different BI perspectives, mainly the customer. Chapter 6 features data mining for temporal data that are particularly suited for analyzing the state view applied to analytical goals occurring in the customer and the production perspective. Chapter 7 presents process analysis techniques and specifically takes the event view on the production perspective. Chapter 8 considers techniques that address questions of organizational mining and questions at the intersections of two perspectives taking different views. For example, in the case of text mining, analytical goals typically arise at the intersection of the customer and the organizational perspective, using a cross-sectional view and focusing on text data. For this, we could be interested in learning the opinion of customers about a certain person or product expressed at some social platform like Twitter.

Similar to the case of modeling, we need for analysis a repository of *analysis techniques* that stores the knowledge about the algorithms, the methods of using these algorithms, and the tools that support the application. These methods represent the knowledge about the required view, the analysis goal, and the data structure for the algorithm.

### 1.3.5 Evaluation and Reporting Task

The evaluation and reporting task has to view the analysis results from two different perspectives. The first one is evaluation of the results in context of the analytical goal and the second one is evaluation from a global business perspective, i.e., understanding the results of the analysis in the context of the business. The main goals are the interpretation of the results in reference to domain knowledge and coming to a decision of how to proceed further. Usually, the evaluation task employs reporting techniques that are similar to data description and visualization techniques. Depending on the intended audience of the report, different types of reporting can be distinguished. We will sketch some ideas in Chap. 4.

### 1.3.6 Analysis Formats

In the same way as we understand all business activities as a process, we can look at BI activities as a process and define a structure for organizing the different tasks. Such structures are subsumed under the term *analysis format* using ideas from life-cycle models for software development or from knowledge discovery in data bases. There is a basic distinction between cyclical and linear formats. We think that

cyclical formats are more useful in BI, because, in practice, covering the different perspectives often requires a sequence of models and a combined evaluation of the results. Another argument in favor of a cyclic model is that a first analysis frequently detects new and unexpected features that need further investigation.

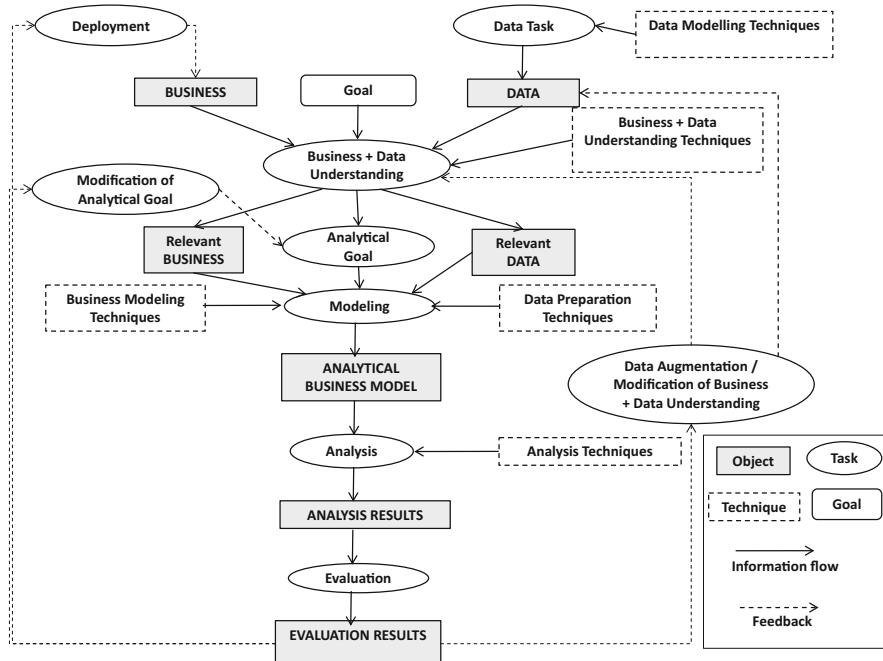
An influential and widely used method format is the Kimball method [12]. It emphasizes the data task and starts with the definition of analysis goals and business understanding. The main focus is on the deployment activity, i.e., the integration of the results of BI activities into a data warehouse. The business analytics aspect is only treated as a secondary aspect, and the integration of unstructured data sources needs some extensions.

A number of analysis formats for the knowledge discovery process have been proposed as data mining formats. One can distinguish between academic-oriented efforts, cf. [10], and application-oriented approaches like CRISP [4], which is nowadays some kind of standard for data mining applications. Other formats are SEMMA or KDD [2]. CRISP mainly focuses on the cross-sectional view on the business process but adaptations to the state view are possible. Our definition of tasks is closely oriented towards CRISP. The main difference is that we formulate a closer connection between business understanding and data understanding. In addition, we approach the modeling and analysis task in a way that allows coping with the event view on the business process and the application of process mining techniques.

Recently, the L\* format [24] has been proposed as a method for business process analysis and mining, which emphasizes explicitly the idea of the application of different models and analysis techniques. The L\* method starts with the event view of a business process, mainly seen from the production perspective. After planning and justifying, which corresponds to business and data understanding, the data (in an event log format) is extracted and a process model is formulated and analyzed. Based on this model, the organizational perspective and the customer perspective are investigated using different analytical methods. Such investigations use a state view and a cross-sectional view on the business process. In a deployment phase, the derived process models can be implemented and operationally supported.

Combining the ideas of CRISP and L\*, we propose the *iMine* analysis format that supports the integrated application of data and process mining. *iMine* hereby stands for integrated mining.

The *iMine* workflow is depicted in Fig. 1.4. The format allows different types of analysis cycles. If the analysis goal is, for example, to discover the real-world processes from data sources, i.e., to conduct process mining, the data used will be in the event view and the modeling techniques and the *analytical techniques* will be process-oriented. If the analysis goal is obtaining knowledge about customers' preferences, the data formats will be rather found within classical, multidimensional table structures corresponding to the data of a warehouse application. In addition, the modeling techniques will be oriented towards statistics and the analysis techniques will provide algorithms for cross-sectional analysis, for example, clustering or association techniques (data mining).



**Fig. 1.4** The *iMine* method

### The *iMine* Method: Basic Ingredients

- **Objects:** Rectangles with full borders represent the objects that are analyzed or produced during the BI process along the three goal perspectives.
- **Tasks:** Ellipses represent the BI tasks that have to be conducted by the BI analyst.
- **Techniques:** Rectangles with dashed border represent techniques for the different BI tasks. Techniques consist of repositories for procedures, methods for using the procedures, and tools for implementation.
- **Information flow:** Solid arrows represent the flow of information.
- **Feedback:** Dashed arrows refer to possible feedback loops.
- **Analysis goals:** Analysis goals are depicted as rectangles with rounded corners.

The *iMine* method starts from the available *data* that has been provided by the *data task* employing *data modeling techniques*. *Data* and *business* together with a certain *analysis goal* provide the input for the task of *business and data understanding*. Thereby, *business* and *goal* together represent that part of reality for which we aim for decision support. The output of this task are *analytical goals* together with *relevant business* and *relevant data*, possibly as a certain fragment or view on the original *business* and *data* along the different BI perspectives customer,

production, and organization. Note that frequently the initially formulated goal needs a number of different analytical goals. For example, a first analytical goal may be a descriptive goal, and afterwards a predictive goal is formulated. Based on the results from *business and data understanding*, an *analytical business model* is built in the *modeling task* using *business modeling techniques* and *data preparation techniques*. Data preparation techniques mainly use transformations mentioned in Sect. 1.3.2, either for unification of the business process views or for generating new data like scores. The *analytical business model* is then analyzed by different *analysis techniques*, leading to *analysis results*. These results are then evaluated yielding *evaluation results*.

In the best case, the evaluation indicates that the analysis results are satisfactory for deployment. This means that we have obtained a result which should be implemented for further business decisions (see the feedback loop at the very left in Fig. 1.4 from *evaluation results* over *deployment* task to the *business*). Such an implementation obviously depends on the chosen strategy about using BI within the business model as discussed in Sect. 1.2.1. The decision might be realized by means of a standardized report, a procedural rule for certain customers, or a reorganization of the business process that was subject to analysis. Corresponding to the strategy, we have to adapt the data and the business itself. Nowadays, it is common to provide a dedicated area for analysis tasks inside a data warehouse which is called *analytical sandbox*. If the results derived in the analytical sandbox are ready for deployment, the operative data might be augmented for usage in the business.

Many a time, further investigations are necessary as defined by the other feedback loops. The feedback loop on the left side of Fig. 1.4 coming from *evaluation results* over a *modification of analytical goal* to the *analytical goal* initiates a new analysis cycle with a new or modified analytical goal and a new understanding of business and data.

Such a new goal may be the consequence of the evaluation of the results. For example, we start with a descriptive goal and learn from the analysis results that the analysis of a predictive model may be reasonable, leading to a respective modification of the analytical goal. In more detail, starting from a scatter plot of a number of variables (descriptive), it can be concluded that certain regression models as predictive models may be feasible.

Another possibility is that the new analysis cycle has been planned already in advance. This frequently occurs for goals at the intersection of the perspectives. If we want to understand, for example, a certain business process, we start with process mining techniques for obtaining the structure of the process. In a second round of analysis, decision trees are applied for understanding the decision in the process. In Fig. 1.3, this corresponds to the intersection of the perspectives production and customer referred to as *Decision Mining*.

The feedback loops *data augmentation* and *modification of business and data understanding* on the right side in Fig. 1.4 refers to further investigations. It takes the *evaluation results* as input for the task of *data augmentation and adaptation of business and data understanding* which provides the input for the adapted *data* and the task of *business and data understanding*. This means that we define a new

analytical (sub)-goal with a different view on the data and the business processes. In some cases, this may imply new data collection activities and a new analytical business model.

### 1.3.7 Summary: Tasks and Analysis Formats

This section presented an introduction into the different BI tasks. The starting point for any BI activity is a business goal that is frequently formulated in terms of KPIs. In the *business and data understanding task*, the framework of further analysis is defined. This requires background knowledge of the application domain and knowledge about possible models and analysis techniques. As a result of this task, one obtains a problem-oriented view on the available data and the business. Analytical goals of interest are defined and a first model for the analysis is formulated. Depending on the goal and the business perspectives of interest, the formulation of the model uses different analytical structures. These structures may be logical oriented, graph oriented, or statistical oriented. Each structure supports different techniques for the analysis task. The term *mining* is used as an umbrella term for the different analysis techniques. The last step is the evaluation and reporting task, which summarizes the findings and gives an interpretation in terms of the original goal.

Frequently, one analysis sequence is not sufficient for achieving the goal and a cyclic analysis format is defined. The format combines ideas of the CRISP format for data mining and the L\* format for process mining. In that sense, it is open for application of analysis methods for data in the different views of the business.

## 1.4 Use Cases

In this section, we introduce five use cases presented throughout the book. They stem from four different application domains, i.e., patient treatment, higher education, logistics, and customer relationship management (CRM). A description of the use cases is given by the following template, which defines the input for an analysis cycle of the *iMine* format outlined in Sect. 1.3. Each use case has a number of goals, which implies the execution of a number of cycles.

#### Use Case Description Template

- **Business Case:**
- **Goals:**
- **Data Task:**

Afterwards, the business and data understanding task for the use cases is summarized in a template along the lines of the overview box in Sect. 1.3.2.

#### Business and Data Understanding Template

- **Application Environment:**
- **Business Perspectives:**
- **BI Views:**
- **Analytical Goals:**
- **Assessment of Data:**

For the different analytical goals templates for the modeling task, the analysis task and the evaluation and reporting task are presented in Chaps. 4–8.

### 1.4.1 Application in Patient Treatment

Health care is a very demanding application area and has been researched by various fields of computer science during the last years. We present two use cases from health care, i.e., skin cancer treatment and pre-eclampsia in pregnancy. The first one is based on a real-life data whereas the second use case is based on a realistic setting, but with simulated data. We opted for these two settings in order to demonstrate different aspects of analysis.

#### The EBMC<sup>2</sup> Project: Description

- **Business Case:** The Evidence-based Medical Compliance Cluster project (EBMC<sup>2</sup>) (see <http://ebmc2.univie.ac.at/>) aims at the analysis of skin cancer treatment processes. The project has been conducted as joint funding and effort between the Medical University of Vienna and the University of Vienna, more precisely, the Department of Dermatology, the Center of Medical Statistics, Informatics, and Intelligent Systems, and the research groups Data Analytics and Computing, Knowledge Engineering, and Workflow Systems and Technology.
- **Goals:** Analysis goals refer to the treatment of patients as well as to the performance of the institution (hospital). With respect to the treatment of patients the following KPIs are of interest:
  - Survival time of patients
  - Compliance of patients with preventive medical check-upsFor performance of the institution, we will consider the following KPIs:
  - Compliance of the institution with the international melanoma guidelines
  - Organization of internal work processes

- **Data Task:** The data sources are data repositories at the local department, the Austrian Cancer Registry, and data from the Austrian Social Insurance System. Data are different with respect to temporal granularity, the quality of information and its completeness, as well as its structure (structured and semistructured data). Hence, we need a flexible data model that allows the integration (i.e. linking of data) according to the analysis goals of interest. Such linking also allows additional quality considerations.

The following template shows the details for the business and data understanding task.

#### EBMC<sup>2</sup> Use Case: Business and Data Understanding

- **Application Environment:** With respect to size, we look at a certain department of a hospital and a specific illness (melanoma) over a longer period of time. This involves many different activities by the hospital and different reactions of the patients. The reference to the global environment is of utmost importance such as the international comparison of comparable institutions and their embedding into the public health-care system. In this case, the BI scenario mainly monitors the strategy performance of the department.
- **Business Perspective:** Corresponding to the different goals, we consider all business perspectives. The process owners are the Department of Dermatology, the process subjects are patients, and the process actors are defined by the staff of the hospital.
- **BI Views:** All three BI views are used depending on the analytical goals. For measuring the effectiveness of the check-up and the treatment, we use the cross-sectional view, for issues of survival time the state view, and for the process-oriented analytical goals the event view.
- **Analytical Goals:** Corresponding to the initial formulation of KPIs a number of analytical goals can be formulated.
  - Effectiveness of preventive medical check-ups
  - Regression model for survival time of patients
  - Compliance analysis for the treatment process
  - Organizational mining for hospital staff
- **Assessment of Data:** Each analytical goal needs a specific data excerpt, which is obtained by data integration and transformations of the data, for example, cross-sectional summaries of the event view. Data description has to be done for each data excerpt, and data quality has to be checked as well. Important issues are completing missing information and improving temporal resolution.

The second use case in medical context considers a specific problem of complication in pregnancy.

#### Pre-eclampsia Use case: Description

- **Business Case:** Pre-eclampsia is a complication in pregnancy caused by multiple factors. In order to detect pre-eclampsia, weight, blood pressure and proteinuria of women are monitored during pregnancy.
- **Goals:** In this use case, we consider the following goals:
  - Representation of the monitoring process by quantitative measures
  - Rules for deciding about the need for hospitalization of a persons
- **Data Task:** This use case was mainly considered for evaluating the *DPA<sub>TimeSeries</sub>* methodology as presented in [9] for decision point analysis. Time series of measurements for weight, systolic and diastolic blood pressure, and proteinuria were generated for 300 cases containing 8 % of cases with pre-eclampsia.

For the business and data understanding task, the specification is given in the following template.

#### Pre-eclampsia Use case: Business and Data Understanding

- **Application Environment:** This application is general without reference to a specific business environment. The scope of the process is rather simple and defined by six types of events: proteinuria check, blood pressure check, weight check, hospitalize patient, homecare, giving birth. The checks are repeated from the 20th week of pregnancy onwards. The BI scenario is BI separated of strategy performance, but the results of the analysis can be used later on for defining new business strategies for monitoring pregnancy.
- **Business Perspective:** The perspective in this use case is on the customer who is owner of the process and process subject. Furthermore, medical practitioners are passive actors who decide about hospitalization.
- **BI View:** This is a typical example of data in the state view, i.e. we have time series about the medical parameters.
- **Analytical Goal:** The KPI is defined by hospitalization in dependence on the temporal behavior of the medical parameters. Correspondingly the analytical goals are descriptive goals (reporting) and predictive goals (classification rules allowing a decision about hospitalization).
- **Assessment of Data:** Due to the control through simulation, the data is complete allowing the evaluation of algorithms. For reporting purposes, summaries for the instances are calculated.

### 1.4.2 Application in Higher Education

The Higher Education Processes (HEP) project features a realistic data set on business processes that reflects the realization of different undergraduate courses at a university. These courses incorporate different stakeholders and a fair amount of process subjects (i.e., students) and participants (e.g., tutors, lecturers).

#### HEP Use Case: Description

- **Business Case:** As part of the Higher Education Processes (HEP) project (see <http://www.wst.univie.ac.at/communities/hep/index.php?t=main>, [15]) undergraduate teaching courses at the Faculty of Computer Science, University of Vienna, were observed based on the usage of the teaching platform CeWEbs [7].
- **Goals:** The following goals are of main interest:
  - Derivation of a reference process model exploiting additional compliance constraints, e.g., setting out the relation of milestones and submissions
  - Conformance between real-life teaching processes and a given reference model
  - Utilization of the forum by students and the relation of students' performance to activities in the forum
- **Data Task:** Data are collected from four distinct services, i.e., forum, submission, registration, and code evaluation, on the service-oriented learning platform CeWEbs. Logs from annually offered undergraduate courses over a period of 3 years (course conducted every year) are available. In total, there were 330 students and 18,511 events. For the use case, data were collected in the .csv format and anonymized.

The specifications for the business and data are shown in the following template.

#### HEP Use Case: Business and Data Understanding

- **Application Environment:** The business case under consideration is defined for a faculty. The scope of the processes is complex due to constraints that hold for an undergraduate teaching process. An example of a constraint with high enforcement level is “For each milestone, no upload must take place after the corresponding milestone deadline” [15]. Finally, we acquired the reference teaching process based on interviews with process participants, e.g., lecturers. The business scenario is monitoring the process and provides feedback on the business strategy of the faculty. In the long run, the results can also be used as a strategic resource for improving the teaching process.
- **Business Perspectives:** All three business perspectives occur in this use case. The process owner is the faculty and the customers and process subjects are the students. Other actors are the tutors and lecturers.

- **BI Views:** The event view will be used for the business understanding goals and the cross-sectional view for the goals in connection with student performance.
- **Analytical Goals:** Corresponding to the goals, the following analytical goals can be formulated:
  - Understanding of the business process
  - Conformance analysis for the process instances
  - Segmentation of students into groups with different utilization of the forum
- **Assessment of Data:** Data are extracted from the service-oriented blended learning environment and integrated along the analysis goals. After data purging and cleaning, data are represented in a log-oriented format. Using data transformations, additional attributes are generated in the cross-sectional view.

### 1.4.3 Application in Logistics

Contrary to the above examples taken from patient treatment and higher education, the logistics use case targets the transportation of containers, i.e., the subjects are not human. The use case is based on a realistic setting but uses synthetic data that was produced by simulation.

#### Logistics Use Case: Description

- **Business Case:** The container transportation case has been adapted from a realistic process that was described and implemented in [3]. It describes the process of loading a vehicle at the origin and starting to move towards its destination. During the movement of the container, temperature is constantly monitored. If the temperature exceeds a certain threshold for some time, the vehicle has to move back to its origin. Otherwise, it continues to the destination where the containers are unloaded.
- **Goals:** Two main goals are identified:
  - Derivation of a model for the transportation process
  - Derivation of a decision rule on when to return the container to their origin before they were delivered
- **Data Task:** Simulated data of the transportation process and the time series measurements of container transportation are used.

#### Logistics Use Case: Business and Data Understanding

- **Application Environment:** Similar to the pre-eclampsia use case, there is no detailed specification of the business environment defined. We consider a general problem for logistics and the scope of the business only refers to a small

number of activities. The application scenario uses BI separated from business strategy for a specific subprocess of a possibly larger process, but results can be used later on.

- **Business Perspective:** The logistic company is the process owner, the containers are the process subjects, and further actors are personnel of the company. Since the containers are understood as customers involved in the transportation process, the main perspective is the customer perspective.
- **BI View:** We use the event view for the first goal and the state view for the second goal.
- **Analytical Goals:** For achieving the goals, we formulate two analytical goals: The first one is process identification and the second one is classification of the different process instances in such a way that an economically favorable decision strategy for returning to the origin can be formulated. Influential factors is the temperature of the container.
- **Assessment of Data:** Due to the fact that the data were simulated, data are already in an adequate format and are of good quality.

#### ***1.4.4 Application in Customer Relationship Management***

This use case is a typical application in customer relationship management (CRM). It is based on a real use case but data and attributes have been modified.

##### **CRM Use Case: Description**

- **Business Case:** A company with outlets in different cities offers customers a variety of services. Customers are registered by a loyalty card. In order to improve services according to customer needs and learning about the image of the company, a survey was carried out. About 2,300 customers participated in the survey. For each customer, age, sex, duration of the business relationship, and usage type (either private or business user) is known. A user profile for long-term customers is defined by three indicators: an indicator for sales, an indicator for the intensity of usage, and an indicator for the duration of customer relationship. Usage of the services is known from the transaction database.
- **Goals:** The following KPIs are of interest for the company:
  - Sales of customers in dependence of their usage profile
  - Usage of the different services
  - Customer profiles with respect to usage of the different services
- **Data Task:** Three data sources are used: a customer data base (loyalty card), a transactional data base, and a survey data base.

Business and data understanding is summarized in the following template.

#### CRM Use Case: Business and Data Understanding

- **Application Environment:** The use case considers a medium-sized company with a large number of activities in the different services, which do not have many dependencies. The application scenario uses BI as feedback for strategy formulation (different services) and probably also as strategic resource (bundling and redefining services).
- **Business Perspective:** Corresponding to the sales orientation, the main perspective is the customer perspective. The owner of the business processes is the company and the process subjects are the customers. Additionally personnel in the outlets are actors in the sales process.
- **BI Views:** All available data are cross-sectional data.
- **Analytical Goals:** In that case, the KPIs define the analysis goal estimation, segmentation, and classification. Further analytical goals are description of customers and detecting interesting behavior.
- **Assessment of Data:** For obtaining cross-sectional data, all transactional data were aggregated at different temporal resolutions (monthly, quarterly, annual). The description of the different sales variables and the identification of outliers are done as well as the description of the survey about user satisfaction.

## 1.5 Structure and Outline of the Book

Chapter 2 discusses different approaches towards the modeling task in BI applications. Chapter 3 presents an overview and details on the data provisioning task, including an excursion on big data. In Chap. 4, the tasks data description, visualization, and reporting are discussed. Chapter 5 introduces data mining techniques for cross-sectional data. Different techniques for the analysis of temporal data are presented in Chap. 6. Chapter 7 discusses techniques for the analysis of process data followed by the introduction of analysis techniques for multiple BI perspectives in Chap. 8. The book closes with a summary and discussion in Chap. 9.

Throughout all chapters, tools are recommended, described, and applied. The focus is on open-source tools in order to be able to set up BI applications at no cost. A survey on tools can be found in the Appendix.

Further on, the concepts presented in this book will be illustrated and deepened by means of selected exercises. These exercises are available on the book's website:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

## 1.6 Recommended Reading (Selection)

As it was argued in the historical overview, there are many different approaches towards business intelligence, and an abundance of literature is available ranging from an elementary introduction up to textbooks at an advanced formal level. Hence, any recommendation is a matter of taste and the understanding of the term business intelligence. Starting from the business context, we aim in this book for an understanding of BI that encompasses data warehousing business analytics and process analysis. For an introduction into BI from a management perspective, we recommend Laursen and Thorlund (2010). If one is mainly interested in the data warehouse perspective, Kimball and Ross (2010) is a standard reference. An extensive exposition of the data mining approach towards BI can be found in Han and Kamber (2011). This book covers in depth the different data structures summarized as different BI views. An excellent reference from a process-oriented point of view is van der Aalst (2012), which covers the different perspectives of BI.

- Laursen G, Thorlund J (2010) Business analytics for managers: taking business intelligence beyond reporting. J. Wiley & SAS Business, New York
- Kimball R, Ross M (2010) The Kimball Group reader: relentlessly practical tools for data warehousing and business intelligence (1). J. Wiley, New York
- Han J, Kamber M (2011) Data mining: concepts and techniques. Morgan Kaufmann Series in Data Management Systems, Waltham MA
- van der Aalst WMP (2011) Process mining—discovery, conformance and enhancement of business processes. Springer, Heidelberg

## References

1. Amit R, Zott Ch (2012) Strategy in changing markets: new business models—creating values through business model innovation. *MIT Sloan Manag Rev* 53(3):41–49
2. Azevedo A, Santos MF (2008) KDD, SEMMA and CRISP-DM: a parallel overview. In: Weghorn H, Abraham AP (eds) IADIS'08: European conference data mining. IADIS Publications, pp 182–185, Pedreira, Portugal
3. Bassil S, Keller RK, Kropf P (2004) A workflow-oriented system architecture for the management of container transportation. In: Desel J, Pernici B, Weske M (eds) BPM'04: international conference on business process management. Lecture notes in computer science, vol 3080. Springer, Heidelberg, pp 116–131
4. Chapman PJ, Clinton J, Kerber R, Khabaza T, Reinartz T, Shearer C, Wirth R (2000) CRISP-DM 1.0 Step-by-step data mining guide. <http://www.the-modeling-agency.com/crisp-dm.pdf>. Accessed 20 December 2014
5. Davenport TH (1992) Process innovation: engineering work through information technology. Reissued edition. Harvard Business Press, Boston
6. Davenport TH (2006) Competing on analytics. *Harv Bus Rev* 84(1):98–107
7. Derntl M, Mangler J (2004) Web services for blended learning patterns. In: Looi K, Sutinen E, Sampson DG, Aedo I, Uden L, Kähkönen E (eds) ICALT'04: international conference on advanced learning technologies. IEEE, New York, pp 614–618

8. Dumas M, La Rosa M, Mendling J, Reijers HA (2013) Fundamentals of business process management. Springer, Berlin/Heidelberg
9. Dunkl R, Rinderle-Ma S, Grossmann W, Fröschl KA (2014) Decision point analysis of time series data in process-aware information systems. In: Nurcan S, Pimenidis E, Pastor O, Vassiliou Y (eds) CaISE Forum: joint proceedings of the CAiSE 2014 Forum and CAiSE 2014 Doctoral Consortium, CEUR workshop proceedings 1164, CEUR-WS.org, pp 33–40
10. Džeroski S (2007) Towards a general framework for data mining. In: Džeroski S, Struyf J (eds) KDID'07: knowledge discovery in inductive databases. Lecture notes in computer science, vol 4747. Springer, Heidelberg, pp 259–300
11. Han J, Kamber M (2011) Data mining: concepts and techniques. Morgan Kaufmann series in data management systems. Morgan Kaufmann, Waltham, MA
12. Kimball R, Ross M (2010) The Kimball Group Reader: relentlessly practical tools for data warehousing and business intelligence, vol 1. Wiley, New York
13. Laursen G, Thorlund J (2010) Business analytics for managers: taking business intelligence beyond reporting. Wiley & SAS Business, New Jersey
14. Luhn HP (1958) A business intelligence system. IBM J Res Dev 2(4):314–319
15. Ly LT, Indiono C, Mangler J, Rinderle-Ma S (2012) Data transformation and semantic log purging for process mining. In: Ralyté J, Franch F, Brinkkemper S, Wrycza S (eds) CaISE'12: international conference on advanced information systems engineering. Lecture notes in computer science, vol 7328. Springer, Heidelberg, pp 238–253
16. Mitchell T (1997) Machine learning. McGraw Hill, New York
17. Moss LT, Atre S (2003) Business intelligence roadmap: the complete project lifecycle for decision-support applications. Addison-Wesley Professional, Boston
18. Negash S (2004) Business intelligence. Commun Assoc Inf Syst 13(1):177–195
19. Power DJ (2007) A brief history of decision support systems. DSSResources. COM, <http://dssresources.com/history/dsshitory.html>. Accessed 20 Dec 2014
20. Rausch P, Sheta AF, Ayesh A (eds) (2013) Business intelligence and performance management: theory, systems and industrial applications. Springer, Berlin/Heidelberg
21. Roebuck K (2011) Business intelligence (BI): high-impact strategies—what you need to know: definitions, adoptions, impact, benefits, maturity, vendors. Emereo, ISBN: 9781743046289
22. Trujillo J, Maté A (2012) Business intelligence 2.0: a general overview. In: Aufaure M-A, Zimanyi E (eds) Business intelligence. Lecture notes in business information processing, vol 96. Springer, Heidelberg, pp 98–116
23. van der Aalst WMP (2011) Process mining—discovery, conformance and enhancement of business processes. Springer, New York/Berlin
24. van der Aalst WMP et al (2012) Process mining manifesto. In Daniel F, Barkaoui K, Dustdar S (eds) Business process management workshops. Lecture notes in business information processing, vol 99. Springer, Heidelberg, pp 169–194
25. Weske M (2012) Business process management: concepts, languages, architectures. Springer, Berlin/Heidelberg

# Chapter 2

## Modeling in Business Intelligence

**Abstract** Models play a central role in Business Intelligence for achieving analysis goals. Depending on the business perspective, the view on the business process, the analysis goals, and the available data, the term model assumes different meanings. This chapter starts with a section dedicated to an overview of different formal approaches to modeling and ideas about model building in Sect. 2.1. Sections 2.2–2.4 present details about the model structures already mentioned in Chap. 1. Section 2.5 discusses data from a modeling point of view. In particular, we emphasize the role of time and data quality.

### 2.1 Models and Modeling in Business Intelligence

As for other scientific disciplines, the terms *model* and *modeling* are used many times in the context of Business Intelligence, and one can find an abundance of model types, for example, business process models, organizational models, regression models, classification models, graphical models, data models, or software models. A closer look at the definitions of these models shows that the notion is ambiguous, although the general intention is similar. Due to the broad range of models in BI, we first discuss models and modeling from a rather general point of view, based on ideas in [13].

The intention behind all modeling activities is the explanation of empirical objects and phenomena in a logical and objective way, which allows transfer into practical action. In Chap. 1, we have already introduced the *iMine* method as a problem-solving strategy in BI. *iMine* can be understood as a top-level *operational model* in BI for obtaining knowledge about a business process and for achieving the analysis goals. Such an operational model is of utmost importance, because a “theory of BI” in the syntactic sense as a set of formal propositions allowing falsifiable or testable predictions seems not realistic. In the best case, we can define a theory of BI from a semantic point of view, which understands the term theory as a family of coordinated well-defined models. However, such a statement ignores many modeling applications in BI that use models as complements to general theories which are too complex to handle or as some kind of preliminary theories (see [13]).

Consequently, we take a pragmatic position and introduce different models for discovering relations (not necessarily causal ones) and rules for business processes.

Each of these models represents some part of the business process and allows a precise formulation of interesting questions that can be analyzed further by using analytical techniques. An example would be to represent the process participants in the HEP use case (cf. Sect. 1.4.2) as constituents of a model, i.e., within a social network, in order to analyze the relationship *working together* among these participants. In other words, a model is based on a *representation function* of a target system, in our case, a business process. Taking this position, two pivotal questions have to be answered:

1. How can we realize the representation function?
2. How should we present or formulate this representation?

In the following, we will discuss these two questions and present some ideas about model building and criteria for model assessment. Furthermore, we will briefly discuss the difference between models and patterns.

### 2.1.1 *The Representation Function of Models*

From a general point of view, three different approaches to the representation function are distinguished in [13], all of which occur in BI:

#### Approaches Towards Representation

- *Models of phenomena*: Phenomena, defined as features of a certain business process, interesting from an analysis point of view, are represented in such a way that within the representation, questions about reality can be formulated and analyzed.
- *Models of data*: Data of process instances are represented in such a way that representation allows the discovery of interesting facts about the business process that has generated the instances.
- *Models of theories*: Resembling the definition of a model in mathematical logic, this approach first defines a formal structure, and the model is understood as interpretation of this formal structure.

All three approaches occur in BI, and we will briefly explain these three different approaches.

#### Models of Phenomena

As stated above, we interpret the term *phenomenon* as an umbrella term as elaborated in [13]. This is a more general usage of the term than the usual understanding of phenomenon as observable occurrences [49]. Such models are an abstraction focusing on some phenomena and defining a picture of reality with

these features. This picture allows the formulation of precise questions, which can be answered using analytical techniques. It is not so important for us whether such a representation is defined according to a positivist view (models as homomorphism of real-world phenomena) or according to a more constructivist view (models as a construction of a modeler).

Many models mentioned at the beginning of this section can be seen as models of phenomena. The differences are mainly consequences of how we define the representation, which perspective of the business process is of main interest, and which view on the data is available. For example, business process models are models of phenomena for the production perspective and the event view. The main feature of interest is the control flow describing admissible sequences of events defined according to some normative rules. Using such a representation, we can ask different questions about the execution of the process. Such types of models are often called *idealized models*. The relation between model and reality is often compared with the relation between a person and a caricature.

Rather similar in the definition of the representation are models for the organizational perspective. The cooperation and communication between different actors in the business process can be represented using ideas from other disciplines. For example, we take ideas from physics about the attraction of bodies in dependence of distance and apply this principle in an organizational model. Such representations are sometimes called *analogical models*. However, distinguishing between idealized and analogical representations is many a time not easy.

In the customer perspective, the definition of the representation in idealized or analogical way is often not possible. The reason is that frequently customers have no precise understanding about the internal business process and react more to signals and events they receive from the process. Consequently, the definition of a model mainly relies on the observable properties. For example, in the case of churn management, we observe that a person decides to quit the connection with a provider, i.e., we observe the phenomenon of canceling the contract. For the definition of a model, we postulate a relation between the cancellation of the contract, sex and age of the customer, and the price of the product. Using this assumption, we represent the relation in a statistical model by some function and add a random term for capturing other possible influences. Models using such ideas for representation are called *phenomenological models*. In this case, the view on the business process is often the cross-sectional view.

The advantage of models of phenomena is that they enable the combination of the theoretical considerations behind the model with the observed reality. A typical example is the question about the conformance of the instances of the business process with the model, for example, in medical applications, the conformance of patients with the planned treatment process. The focus on the comparison of the model with a usually large number of observed process instances distinguishes vague BI from operations research, which rather considers questions about optimal behavior of processes.

## Models of Data

As already mentioned at the end of the discussion of models of phenomena, data play an essential role in BI for the comparison of models and reality. Models of data go one step further. They abstain from a theoretical model for the phenomena and want to “learn” such a model from the observed data. In the analysis goals introduced in Sect. 1.2.4, such models play an essential role in all perspectives of the business process. In the case of the production perspective, we are many a time confronted with the fact that there is no precise definition of the control flow of the business process, but we want to identify the control flow from the observed instances. In the case of the customer perspective, models of data occur in connection with analysis goals like understanding why customers quit, how long it takes until the business relation ends, or why a credit defaults. Contrary to the phenomenological model discussed above, we do not know which of the attributes influence the target variable, but we want to learn from the data which model is most appropriate. In the organizational perspective, it may be of interest to learn the different roles of the actors in a business process. In order to learn the different roles, we can use a segmentation model.

These examples show that models of data play an essential role in BI. The usual approach for obtaining models of data is to start with some preprocessing of the existing data in order to achieve “clean” data. It should be noted that preprocessing itself relies on models of data and is an essential part of the data understanding task discussed in connection with the iMine method (cf. Sect. 1.3.4). The second step consists in defining candidate models including the most adequate model for the analytical goal, which is selected according to a set of certain criteria. Sometimes, models of data can be found by using techniques for data description and data visualization. More frequently, candidate models are based on models of phenomena, for example, statistical models in case of the cross-sectional view or the state view. After the selection of a model, one has to check its performance with new data. The importance of models of data is also emphasized by the term data mining for the different analytical techniques.

## Models of Theories

Models of theories are mainly used for representing knowledge about the domain of a BI application. Domain knowledge is usually formulated in terms of important concepts of the domain and logical dependencies between these concepts. The concepts together with their relations define a formal system which describes the scheme for a database or, expressed in a more general way, an ontology. If we understand such a formulation as theory of the business, the data instances are a realization of this formal system and define a model. Data models resemble the understanding of the term model used in logic, and details of this approach will be sketched in Sect. 2.2.2. A more practical point of view of data models, which are an indispensable requirement for all BI activities, is treated in detail in Chap. 3.

### 2.1.2 Model Presentation

As soon as we have decided about the representation of the business process, we have to choose an appropriate formulation for the model by combining our understanding of the business process and the available data in such a way that the questions defined by the analytical goals can be analyzed. With respect to presentation, one can distinguish between iconic models and linguistic models. In BI, linguistic models are dominant, and many expositions about business processes, for example, [34], start the presentation of the model with the definition of a *model language*. This is done by means of syntax, semantics, and notation, including a *modeling method* for describing usage of the language.

A review of models used in BI shows that in fact many presentation languages exist. The reason for this diversity are the distinct perspectives of the business process, the possible views on the data, and the analytical goals as shown in Fig. 1.2. Let us consider some typical examples:

- If we are interested in the description of data and the retrieval of instances with certain attributes, a data language like UML would be a useful tool, rather independent of business perspective and view on the data.
- If we are interested in understanding the logic of the control flow of the business process, an appropriate choice would be a language for representing the business process from a production perspective using data in the event view.
- If our main interest is understanding the dependency of customer behavior on attributes of the process and of the customers itself, a language oriented towards mathematics and statistics is useful, in particular in the case of data in the cross-sectional view.
- If we are interested in organizational issues like the cooperation and communication between the actors, a graph language which describes social networks would be helpful.

Depending on the language, there are different syntactic rules for formulating the statements in this language, and the semantics of the language attributes meaning to the statements, rather independent of the domain. For example, in a business process modeling language, a start and an end event carries a clear meaning, a linear regression function carries meaning in a statistical language, or a distance between objects carries a well-defined meaning in a graph language. Let us call this kind of semantics *model language semantics*. This also allows formulation of different *model elements* that can be used later on for building models for a domain problem. Moreover, using this model semantics, one can define a number of so-called *generic questions* about the model elements and analysis techniques, which were developed in the context of this language support answering these questions. Let us consider some examples for generic questions:

- In the case of a business process model, such a generic question may be whether two events can occur simultaneously in the process.
- In the case of a statistical model, questions about the strength of the relationship between two attributes can be answered by calculation of a correlation.

- In the case of a graph model, questions about the reachability of one node from the other can be answered by using algorithms for finding paths.

Due to the fact that these languages have a long tradition in different scientific disciplines, language-specific notations exist, which has been proven useful for denoting the model elements and for answering generic questions. Moreover, similar generic questions can be formulated in different languages, and the answers for such generic questions may be different. For example, if we are interested in answering a question about the relation between different attributes, we can use model elements of different model languages as the following options show:

- In a data language, we can use a table as model element and answer the question for the different combinations of the attribute values. A query language can be used as analysis technique.
- In a mathematical-statistical language, we can use a linear function as model element and formulate the relation as an equation.
- In a visualization language, we can use a scatterplot as model element and visualize the relation according to analytical techniques for data visualization.

This shows that using models formulated in different languages is of utmost importance in BI applications. For a unified description, we introduce the term *model structures* which shall comprise the language, the model elements, and the generic questions. In Sects. 2.2–2.4, we will discuss important model structures together with frequently used models in detail.

### Model Structures

1. *Model language*: A model language is based on some fundamental elements and a model syntax which defines how expressions are built in the language. The model language semantics describes the meaning of the expressions within the language. A notation is used for communication about the expressions in the language.
2. *Model elements*: A model element is a certain expression in the model language useful for describing facts about the business process. Model elements have its own model semantics rather independent from applications in a domain.
3. *Generic questions*: A generic question is a question formulated in the model semantics about properties of model elements. Generic questions can be answered using specific analysis techniques.
4. *Model structure*: A model structure is a model language together with model elements and generic questions. A common feature of all model structures is that the usage of the model structure is supported by software tools.

Taking a strictly formalized deductive view on model structures, it could be argued that all these model structures can be transformed into one unified language. Such an approach is useful if we are interested in the representation of the model

language as implementation, but we doubt whether this idea is useful for BI applications. In practice, the introduction of interesting model structures independent of an implementation is more useful, provided the model structure has shown formal correctness within its scope. It allows a more user-friendly formulation of real-world problems and increases flexibility. Besides model formulation, this approach is also advantageous for analysis, because each model language allows the formulation of different kinds of generic questions, and there exist language-specific analysis techniques for answering these generic questions. In that sense, we prefer the approach in [30] that concerning the modeling method, distinguishes between modeling procedures for formulating the model and algorithms and mechanics for analyzing questions within the model. What is called algorithms and mechanics in [30] is summarized in Chap. 1 under the term analytical techniques. Their application to different models is the main issue in subsequent chapters.

### 2.1.3 ***Model Building***

As already stated in Sect. 1.3.3, we can understand modeling as a mapping of some part of the domain semantics of the business process into the model structure. This happens in such a way that the available data enable formal analysis of questions about the business process. The term *domain semantics* encompasses the definition of domain concepts (i.e., a terminology) and a number of interesting questions about the domain concepts. In computer science, the mapping of the domain semantics into a model structure is frequently denoted by the term *conceptual modeling*. The result is a representation of reality that can be used for the operational handling of the questions of interest, formulated as KPIs or as analytical goals. Different questions use different parts of domain knowledge, and within one application, different model structures can be useful. Essentially, this was the idea behind the definition of the different BI perspectives in Chap. 1. Let us explain the idea of domain semantics by the four use cases outlined in Sect. 1.4.

#### Domain Semantics in Use Case 1

- **EBMC<sup>2</sup> use case:** In health care, the semantics of the business process centers around concepts such as diagnosis, treatment, patient properties, or available therapies. If we are interested in questions about the conformance of patient treatment with medical guidelines, a model structure capturing the events described by the guideline will be appropriate, including a method for measuring the distance between guideline processes and actual processes. If we are interested in questions about survival times of patients, a model structure explaining the survival time in dependence of treatment variables and personal attributes of patients will be a good choice for capturing domain semantics in a model.

#### Domain Semantics in Use Cases 2, 3, and 4

- **HEP use case:** The semantics of the higher education processes is based on concepts such as students, lectures, exercises, and tests. Behavior is described by means of the fulfillment of assignments and by the communication in the forum. In the first case, it is again a model structure for the event view that enables the mapping of the domain problem into a modeling language. For the second question, a cross-sectional view on the entries in the forum and mapping the relations onto a graph may be a starting point for the analysis. If our analysis goal is the description of the students' performance, a model structure for the tabulation of success in dependence of some other variables will be appropriate, together with graphical display.
- **Logistics use case:** The semantics of the business process is defined by concepts such as driver, container, and an explicit state variable container temperature. If our KPI is asking for the decision for successful delivery of the container depending on its temperature, the model structure must capture events, and the values of the variable container temperature.
- **CRM use case:** In this application, the semantics of a sales process is based on the concept of products, customers, services, customer profiles, and legal conditions. Due to the fact that we have only cross-sectional data at hand, the business process is of minor importance. We can map the questions about customer behavior in the KPIs on well-known statistical models for segmentation and regression. Again, for reporting, we mainly use a presentation of the data in a data model.

The examples show that in some cases, the necessary domain semantics is a rather general semantics of information flow of a business process. In other cases, detailed knowledge of the domain and customer behavior are essential for the model. Depending on the KPI and the analytical goal of interest, we have to use different model structures, even in one application domain. Taking this complexity of the modeling task into account, it is not surprising that it is frequently compared to a form of art. To facilitate successful model building, the following topics have to be considered by a *modeling method*:

#### Topics of a Modeling Method

1. *Definition of model configuration:* A model configuration is an admissible expression in a certain model structure that allows the formulation and answering of the analysis goal as a question about the properties of the model configuration.
2. *Connection between model configuration and observations:* The model configuration has to refer to reality using data describing the business process and data about the instances as input and output. This reference has to be established regarding available data and the different views on the business process.

3. *Definition of model variability:* Usually, data about the instances are blurred either due to noise or due to statistical variability. Hence, it has to be decided how this variability shall be incorporated into the model. This is of utmost importance in the case of models for the customer perspective.

We define a *modeling method* as a description of how to build a model by means of some model structures. The support of a modeling method is done by tools, which, together with the modeling method, define a *modeling technique*. The three topics of modeling require a combination of logical and algebraic considerations for the definition of the model configuration with probabilistic and statistical considerations for capturing variability. The importance of the topics depends on the type of model and the analysis goal. Typically, in the production perspective, when the aim is building models for different business divisions, e.g., sales or accounting, variability plays only a minor role. In the case of the customer perspective, however, variability is a core issue, and the definition of the configuration is many a time rather easy. The connection between the model configuration and data is always an important topic, and sometimes data are the limiting factor for useful model configurations. For example, if, for comparison purposes, a model requires data in the event view but only data in the cross-sectional view are available, a connection may be difficult and one may choose a simpler model better matching with the data.

A popular way for the description of modeling is the definition of a *meta-model*. Here, we understand the term meta-model as a formalism of modeling that describes how basic constituents are combined in a logical or algebraic way and how to connect these models to blurred observations. Usually, the term meta-model considers only the logical or algebraic part of modeling. This is many a time sufficient in the case of models regarding the business process from the production perspective. In such cases, simulation frequently considers variability as an analytical technique. However, in BI, we have to deal with the customer perspective as well which needs additional stochastic components, because the motivation of decision of the customers is frequently not known. This means that the meta-model capturing the different BI perspectives must be rather general.

Let us mention that a general approach for model building was developed under the heading *pattern theory*. The name pattern theory indicates its root in pattern recognition and should not be confused with the term pattern in software engineering. An introduction showing how to use such a general modeling technique in various application domains can be found in [15]. Our application for BI is rather simple from a mathematical point of view and has been discussed in [31]. An application for modeling business processes in the nuclear safety inspection process can be found in [1]. The implementation of the model was done in ADOxx<sup>®</sup><sup>1</sup> as meta-modeling platform.

---

<sup>1</sup><http://www.adoxx.org/live/home>.

### 2.1.4 Model Assessment and Quality of Models

In BI, the design of a model usually includes a number of decisions of the modeler, and one has to assess a model according to some objective criteria. Such assessment criteria are also called *quality criteria* of a model. Different communities have defined various criteria for model quality depending on the used model structures and the analytical questions answered by the model. In the case of information models with emphasis on the production perspective, the *Guidelines of Modeling* [40] define the following criteria (adapted from [34]):

#### Quality Criteria for Business Process Models

1. *Correctness*: The model should be syntactically correct within the used model structure, and it combines the model semantics with the domain semantics in an appropriate way.
2. *Relevance*: The model has to comply with its intended function. In BI, relevance usually depends on the ability to explain past observations and to predict future observations.
3. *Economic efficiency*: There should be a reasonable trade-off between the model complexity necessary for achieving the implied needs and the cost of usage of the model in practice.
4. *Clarity*: The model should be understood by the users of the model and should have some aesthetic appeal.
5. *Comparability*: The model should fit into the overall framework of an analysis of a business process. In particular, if we use different models for instances of the same business process, the results should be consistent.

In social sciences and psychology, the quality of models is considered from a more empirical and statistical point of view emphasizing the quality of the measurement process for the data.

#### Quality Criteria for Empirical Models

1. *Objectivity*: A model is objective if the results are independent of the person using the model and of the methods of measurement used for obtaining the data.
2. *Reliability*: A model is reliable if the results can be reproduced. Reliability is connected to repeatability and to precision, which means that under similar conditions, the results of the model would be rather similar.
3. *Validity*: The model is useful from a practical point of view. Validity is in close connection to accuracy which is defined by the closeness of the results from the model to reality. One can distinguish between content validity (results represent the phenomenon under consideration to a high degree), criterion validity (there

is a high correlation between results of the model and other external properties), and construct validity (one can derive new results from the model).

The importance of validity and how to assess the validity of models will be discussed in connection with analytical techniques for predictive goals in Chap. 5.

### 2.1.5 *Models and Patterns*

In connection with models, some applications use the term *pattern*. In BI, a pattern refers to the description of local behavior of the business process, whereas a model represents the global behavior. Such a distinction can be made for all perspectives and views of the business process. Let us illustrate this with some examples for the different BI views.

- In the case of the event or the state view, a model will cover the course of all relevant events, whereas a pattern would be interested only in the occurrence of some well-defined events. For example, in medical applications, a pattern could be the co-occurrence of some events like the intake of different medications or the occurrence of different types of cancer. In business applications, a typical pattern is that a customer uses certain types of services. Many a time, the main question of interest is how frequently such patterns occur.
- In the case of the cross-sectional view, a model typically aims at describing the relationship between variables in terms of a function, whereas a pattern would focus on the occurrence of a specific value for some variables, which usually occur only for few instances of the business process, i.e., records in the data.

From the analysis point of view, patterns are of interest because sometimes important events in the process are connected with such local behavior and we can learn more about the process. Techniques for description and visualization considered in Chap. 4 are often useful for detecting patterns in data. In Chap. 6, we will consider methods for finding so-called association patterns. A more systematic treatment of patterns can be found in [22].

In this book, we will not strictly distinguish between models and patterns and refer only in the analysis techniques whether the technique is more oriented towards models or patterns. One reason is that the term pattern is used in computer science and software engineering with a slightly different meaning denoting a reusable template (see, e.g., [6]).

### 2.1.6 Summary: Models and Modeling in Business Intelligence

Finding answers for analytical goals is based on models. Hence, modeling is an essential task in BI. Besides models of phenomena, based on idealization, analogy, or definition of an equation, BI uses frequently a modeling approach characterized by the term *models of data*. This means that we want to learn the relation between a quantity of interest, in many cases a KPI, and the explanatory variables, so-called influential factors, from the empirical data. Besides these approaches, another approach towards modeling called models of theories is used. It resembles the understanding of the term model in predicate logic which defines the model as an interpretation of a formal theory.

Formulation of a model requires a modeling language with a specific syntax, semantics, and notation. Moreover, such model languages allow the definition of a number of generic model elements and provide algorithms and procedures for answering questions about properties of a model. We introduced the term model structure for the model language together with some generic model elements and the algorithms.

Any useful model has to fulfill a number of quality criteria. Quality criteria were discussed from a process-oriented modeling perspective as well as from an empirical modeling perspective.

Besides modeling in the sense described above, patterns play in many BI applications an important role. In BI, the term pattern is used for describing local structures, for example, the co-occurrence of two events, or an interesting value of a variable outside the standard range.

## 2.2 Logical and Algebraic Structures

In BI, logical and algebraic structures are mainly used for the description of the domain semantics. They are useful for defining a knowledge base, a knowledge organization system, an XML schema, or archetypes (a term popular in medical informatics) and form the background for data provisioning treated in Chap. 3. In Sect. 2.2.1, we describe logical model structures and in Sect. 2.2.2, two important models.

### 2.2.1 Logical Structures

#### Language

From the many different logical structures, *propositional logic* and *predicate logic* are the most important ones in BI applications. A good introduction into logic as a tool for knowledge representation can be found in [43].

Propositional logic is concerned with the formal building of propositions from elementary propositions using the well-known operators AND ( $\wedge$ ), OR( $\vee$ ), NOT ( $\neg$ ). Using these operators, one can define the logical implication if... then ( $\Rightarrow$ ) and equivalence if... and only if ( $\Leftrightarrow$ ). The semantics of a model in propositional logic is based on the assignment of truth values to propositions and the propagation of the truth values according to the well-known rules for logical operations.

More interesting for applications is first-order predicate logic which enlarges the structure of propositional logic by defining individual constants (names), individual variables as place holders for constants, functions operating on constants or variables, and predicates defining properties for the individual constants. Functions and predicates have a fixed arity defining the number of allowed arguments. In the case of predicates, unary predicates represent properties,  $n$ -ary predicates relations between the constants. The quantifiers for all ( $\forall$ ) and there exists ( $\exists$ ) complete the list of basic syntactic elements.

#### HEP Use Case: Predicate Logic Excerpt

In the HEP use case, we can define the following constants: John, Dee, Martha, Height, and Peter, Knight as names of students, Advanced Topics in Meta-Modeling and Applied Business Intelligence as names of courses, and passed and not passed as possible values of a grade. As variables, we define student and course, and an example of a function could be `grade(student, course)` assigning the grade in a specific course to each student. A predicate `attendsBT` can be used for defining the property that a student attends the course Applied Business Intelligence.

Using these syntax elements, first of all, one defines terms regarding individual constants, individual variables, and functions. In the next step, one generates atomic formulas by a predicate symbol, followed by a number of terms in brackets for which the predicate is applicable. The third step is building so-called well-formed formulas by the application of propositional calculus and quantification of atomic formulas. In case of our example outlined above, a possible well-formed formula could be:

$$\exists(\text{student})(\forall(\text{course})) \text{grade}(\text{student}, \text{course}) = \text{passed} \quad (2.1)$$

meaning that there exists a student who passed all courses.

### Model Elements and Generic Questions

The model elements in predicate logic are defined according to the method used in the example above, i.e., by mapping the domain semantics into predicate calculus in the following way: First, we assign values in a certain domain to the individual constants and call it interpretation  $I$ ; next, operations are assigned for the function symbols; afterwards, unary predicates are used for defining the properties of the constants and  $n$ -ary predicates for the relations between constants. Now, the important step is to assign truth values to the expressions. We start by assigning

truth values to atomic formulas and call them facts. Using these facts, we obtain truth values for all well-formed formulas if we assign each free individual variable to any possible individual constant and calculate the truth values using the rules of propositional calculus. If such an interpretation results in the truth value TRUE for all possible assignments of the free variables, we call the interpretation a model.

The generic question answered by the model is whether a certain assertion defined by well-formed formulas is correct for some data described by the model. This immediately leads to the retrieval of properties of the business process from a database of the process instances.

### 2.2.2 *Modeling Using Logical Structures*

The definition of a logical model gives us the opportunity to describe the domain structure as a set of logical propositions in an intuitive way. In this section, we will briefly consider ontologies and frames as examples of two frequently used models. Data models are treated in Chap. 3.

#### Ontologies

Nowadays, ontologies are a popular approach for the specification of a model for the domain semantics. Many definitions exist of this term. One frequently used definition was formulated by Gruber in 1993 stating that an ontology is a “specification of a conceptualization” [17]. This definition is open for many different interpretations, and there are many ways to make this definition more precise (see, e.g., [19]). In [18], the term specification is explained by the “definition of a presentational vocabulary (classes, relations, …), which provide meaning.” This definition of vocabulary can be understood as a specification of a data model for a domain at the semantic level close to predicate logic and independent from the data structure.

A paper worth reading about the definition of the term ontology and its usage is [23]. In this paper, the differences between an ontology, XML schemas, knowledge bases, and knowledge organization systems are elaborated, and different dimensions for evaluating ontologies are defined. In particular, the following six dimensions are identified:

#### Dimensions of an Ontology

- *Expressiveness*: From simple relations between concepts up to logical theories.
- *Size of the relevant community*: The ontology has to be documented in such a way that it is understood by the intended community.
- *Dynamics of the domain*: Domains with frequently changing concepts need a versioning strategy and frequently prohibit strict axiomatized formulation in detail.

- *Number of conceptual elements in the domain:* Large ontologies are difficult to visualize and are hard to adapt.
- *Degree of subjectivity in the conceptualization:* Domains with concepts of high subjectivity require a consensus mechanism for the definitions of the concepts.
- *Average size of the specification:* The number of attributes necessary for precise specification of the concepts.

Modeling a domain by an ontology has a number of practical advantages [23].

### Reasons for Using Ontologies

- *Communication:* Ontologies facilitate communication at different levels (between systems, between humans, between humans and systems).
- *Inference:* Ontologies can be used for computational inference (representation of plans and analysis of internal structures).
- *Organization and reuse:* Ontologies support the reuse and organization of knowledge.

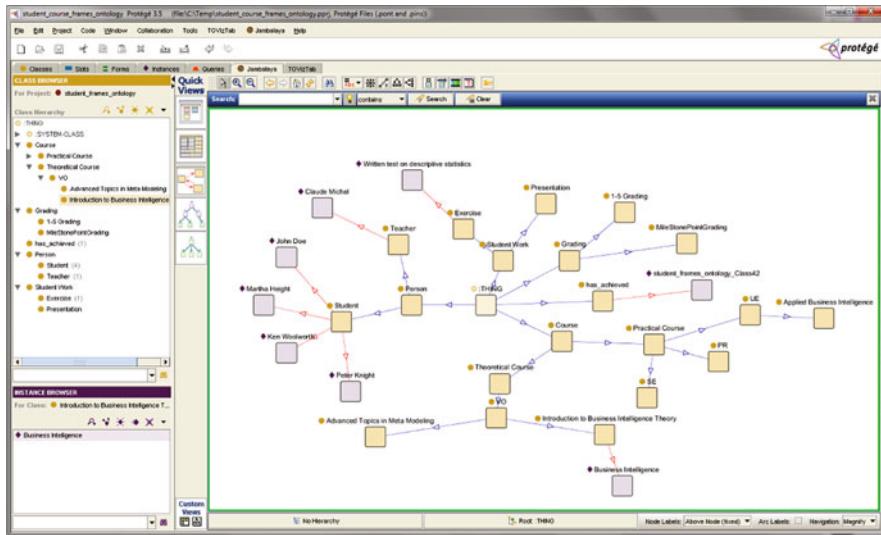
For example, in medical applications, such advantages are of interest for the exchange of data via electronic health records (EHR) of patients. In business applications, a typical implementation is the interchange of electronic data between business partners.

Ontologies exist for many application domains. A list of ontologies in different domains can be found in [50]. Besides domain ontologies, there also exist ontologies applicable in different domains, so-called upper ontologies.

The representation of an ontology may be achieved in different ways. Well-known approaches are terminological systems and taxonomies. More elaborated systems use description logic for the definition of ontologies, which also allow data and fact retrieval [3]. The realization of description logic in OWL (Web Ontology Language, [27]) is the background of the Semantic Web. In order to represent the terminology, it uses the so-called TBox (vocabulary as a logical theory). The formulation of an assertion is done in the ABox, which is afterwards checked by the language. OWL uses the open-world assumption. That means that anything can be entered in the knowledge base (Tbox) unless it violates some constraints. This is a useful assumption for the World Wide Web.

### Frames

Frames represent knowledge in an object-oriented style. Knowledge about an object is represented by a number of slots. The slots may refer to attributes of the objects, to rules, or to procedures concerning the way how the value of the slot is obtained. Contrary to OWL, frames use the closed-world assumption, which means that a statement is true if its negation cannot be proven within the system. A classical example is a statement like “all birds can fly.” Under the closed-world assumption, such a statement would be correct provided there are no nonflying birds like



**Fig. 2.1** Ontology for higher education use case (using Protégé)

penguins in the system. Under the open-world assumption, the contrary statement “there exist nonflying birds” could be entered. The closed-world assumption is useful in many applications where we have to handle partial knowledge. A short comparison of OWL and frames can be found in [47].

#### The HEP Use Case: Ontology Excerpt

An ontology modeled with Protégé OWL may look at the top level as depicted in Fig. 2.1. The ontology is based on an extended version of the terminology introduced in the example on predicate logic in Sect. 2.2.1. An overview of the classes is given on the left side of the figure. The large pane features the structure of the ontology. In the center is the system class `Thing` from which all the other classes are derived. The class `Persons` contains the class `Student` and also the class `Teacher` with different instances. The instances of the classes are represented as slots. For example, `Claude Michel` is an instance of the class `Teacher`. Similarly, the class `Courses` is divided into `Theoretical Courses` and `Practical Courses`. Instead of the simple grading used in Sect. 2.2.1, we use more sophisticated grading schemes. Additionally, the class `Student Work` for different types of student activities is introduced. As an example of a predicate, `has achieved` is used for describing the students’ results in more detail.

A well-known open-source tool for editing ontologies is Protégé.<sup>2</sup> It supports modeling ontologies via a Web client or a desktop client. Ontologies can be developed in a variety of formats such as OWL or XML schemas.

### 2.2.3 ***Summary: Logical Structures***

Logical structures are an appropriate tool for representation of the relations between the concepts of a BI application. The connection between model structure semantics and the domain semantics essentially depends on the understanding of the concepts and the relations by the modeler. One can represent the conceptual system via a logical theory using ontology tools (OWL, XML schemas) or in a more naive sense. The choice depends on the intended usage. If the main interest is a model for a specific application within a small community, it is probably easier to develop a local knowledge base for the application. Nevertheless, broader knowledge of the business analyst may be helpful for thinking about applications at a larger scale.

## 2.3 **Graph Structures**

Graph structures play an important role in many BI models and are useful for modeling relationships in a business process. The semantics of graph structures offers a number of constructs for analyzing properties of such relationships. Again, we first describe the structure of graph models followed by different modeling methods.

### 2.3.1 ***Model Structure***

#### **Language**

The basic syntax of graph structures is defined by *nodes*<sup>3</sup> and *edges*. If we provide no additional specification, we will use the notation  $G = (V, E)$  where  $V$  denotes the set of nodes and  $E$  the set of edges. Edges may be directed or undirected, which gives them different semantic interpretations. Undirected edges  $e = (v_1, v_2)$ ,  $v_1, v_2 \in V$  simply represent a relation between the nodes which are connected, whereas directed edges  $e = (v_1, v_2)$  allow an interpretation as temporal or logical (causal) ordering or dependencies between the vertices  $v_1, v_2 \in V$ . The nodes

---

<sup>2</sup><http://protege.stanford.edu/>.

<sup>3</sup>Also referred to as *vertices*.

defining a directed edge  $e = (v_1, v_2)$  are called parent or predecessor ( $v_1$ ) and child (descendants) or successor ( $v_2$ ).

An additional syntactic element are labels for the edges and the nodes. Labels for the edges can be interpreted in a generic way as costs or capacities and are useful in many domain models. In the case of nodes, labels may refer to the *degree*, i.e., the number of edges at a specific node. In directed graphs, we distinguish between *in-degree* and *out-degree*, which counts the edges ending or starting at the node. Furthermore, labels can be used for the characterization of different types of nodes and edges according to special roles inside the graph including a start node, sometimes called source, or an end node called also terminal.

A key issue for applying graphs as BI models is the appropriate notation. For communication purposes, an iconic notation for displaying the graph as figure is, in most cases, a good choice. For computation, the *adjacency matrix* is often recommendable. An adjacency matrix for graph  $G = (V, E)$  is an  $|V| \times |V|$  matrix where columns and edges are denoted by the node labels. An entry  $e_{(i,j)}$  for  $v_i, v_j \in V$  is defined as

$$e_{(i,j)} := \begin{cases} 1 & \text{if } \exists(v_i, v_j) \in E; \\ 0 & \text{otherwise.} \end{cases} \quad (2.2)$$

In case of labeled edges, it is useful to enter the value of the label instead of 1. The matrix representation, has also the advantage that a number of questions can be answered by using matrix calculus of linear algebra. Besides this representation there exist a number of more efficient implementations for graph notations. As for the following considerations, studying this issue is of minor importance; the interested reader is referred to [29].

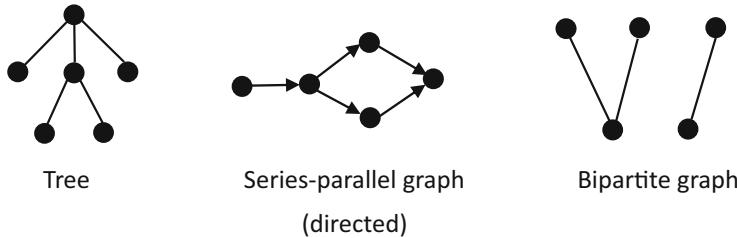
### Model Elements and Generic Questions

In practical applications, we frequently encounter special graph structures, i.e., graphs that possess certain properties that might be interesting for later analysis. For the definition of special graphs, the concept of a *path* is of utmost importance. A path  $p$  is defined by a number of nodes connected by edges as follows:

$$\begin{aligned} p &:= (v_1, v_2, \dots, v_k), v_i \in V, i = 1, \dots, k \wedge \forall v_j, v_{j+1}, \exists(v_j, v_{j+1}) \in E, \\ j &= 1, \dots, k - 1. \end{aligned} \quad (2.3)$$

The number  $k - 1$  is the *length* of the path. Two paths are called *vertex disjoint* if they have only the start and end node in common. A path which ends at its starting node (i.e.,  $v_1 = v_k$ ) is called a *circle*. A circle with length equal to 1 is called a *loop*.

A graph is *k-connected* if, for every pair of its vertices, it is possible to find  $k$  vertex-disjoint paths connecting these vertices. A special case of connected graph is the *complete graph*, which contains edges between all pairs of nodes so that  $|E| = |V| * |V|$ .



**Fig. 2.2** Examples of generic graph structures

Of utmost importance in BI are trees, series-parallel networks, and bipartite graphs. Figure 2.2 depicts examples of generic graph structures, i.e., a tree, a (directed) series-parallel graph, and a bipartite graph.

A *tree* is a directed graph which contains one root node with an in-degree of 0 and a number of leafs or terminal nodes having an out-degree of 0. Specifically, trees contain neither circles nor loops. If every node in the tree has at most two descendant nodes, we call the tree a *binary tree*.

*Series-parallel graphs* play an important role in describing the control flow of a business process. Such graphs can be generated from a graph with two nodes,  $s$  and  $t$ , and one edge  $(s, t)$  by two operations called parallel composition and series composition. The parallel composition is defined as disjoint union of two series-parallel graphs by merging the source nodes and terminal nodes. The series composition connects the terminal node of the first graph with the source node of the second graph. For a visualization of series composition, parallel composition, and some formal studies on series-parallel graphs, see, for example, [7].

A *bipartite graph*  $G = (V, E)$  is characterized by the fact that the set of nodes is partitioned into two disjoint sets  $V_1$  and  $V_2$ , i.e.,  $V = V_1 \cup V_2 \wedge V_1 \cap V_2 = \emptyset$  and only such edges exist that connect a node from  $V_1$  with a node from  $V_2$  (and vice versa for directed edges).

Graphs induce a rich model semantics, which allows the formulation of a number of generic analysis questions, useful for mapping a domain-oriented question about business processes into a generic graph theoretic problem. Well-known problems are the problem of finding the shortest path in a labeled graph, the problem of deciding whether a graph is connected or not, the construction of a spanning tree in a connected graph, the problem of finding a maximum flow in a series-parallel network, or the problem of finding the best matching of nodes in a bipartite graph. Text books about graph theory [29], operations research [25], or algorithms [41] treat these topics from different perspectives and present algorithms for solving such problems.

### 2.3.2 Modeling with Graph Structures

As soon as we have defined a graph structure, we can define a number of useful modeling concepts by interpreting the graph semantics in the context of the business process. Here, we introduce the Business Process Model and Notation (BPMN) and Petri nets for models in the production perspective. The first builds upon directed series-parallel graphs, the second one on bipartite graphs. In Chap. 6, we will see how connected graphs can be applied for analyzing the organizational perspective of the business process, and in Chap. 5, we will apply trees and layered networks for analysis of analytical goals in the customer perspective.

#### Business Process Model and Notation (BPMN)

BPMN has developed as de facto standard for modeling business processes recently. The current version is BPMN 2.0 and is maintained by the Object Management Group.<sup>4</sup> The strengths of BPMN include the provision of a rich set of events that can be captured within BPMN process models as well as its perspective to model cross-organizational or business-to-business (B2B) processes.

It would exceed the scope of this book to provide the complete specification of BPMN 2.0. Hence, we give an overview on the basic concepts that helps to understand the examples used in this publication and refer to the BPMN 2.0 specification document [9] instead.

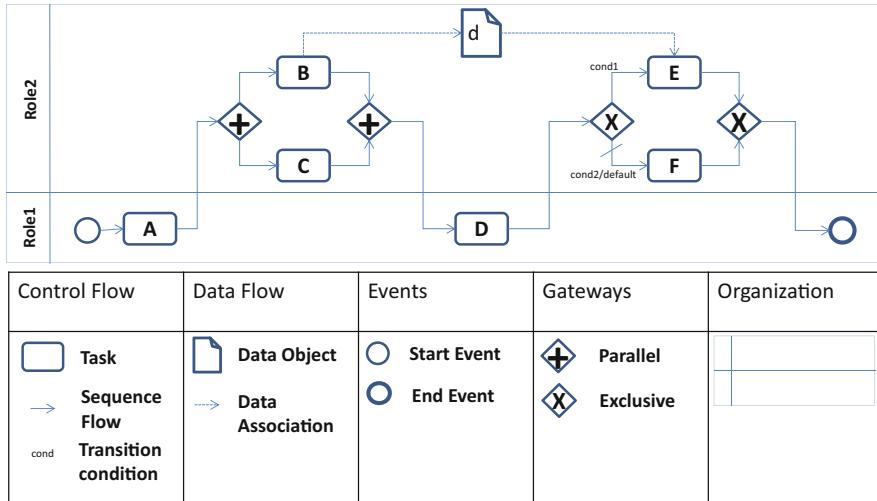
At its heart, a BPMN model is a directed series-parallel graph  $G = (V, E)$  consisting of a set of *activities* (reflected by nodes  $V$ ) that are linked by a set of *sequence flow connectors* (reflected by edges  $E$ ). Consider the BPMN model depicted in Fig. 2.3 that consists of nodes and connecting edges and unfolds a combination of node series, e.g., (Start, A), and parallel constructs, e.g., the construct comprising nodes labeled with B and C.

Nodes can be of different types. Labeled nodes represent *activities*, and their label reflects the activity name, for example, A. Activities can be of different types, too. A basic activity type is, for example, *task* which denotes an activity that is performed by a human actor or an application component during runtime. The activity types are distinguished by using different symbols. The process model depicted in Fig. 2.3, for example, consists of 6 activities (tasks).

In order to model routing constructs such as parallel or alternative branchings, BPMN offers another type of nodes, i.e., so-called *gateways*. In this book, we will use *parallel* and *exclusive* gateways. Both gateway types are shown in the process model in Fig. 2.3: the usage of the parallel gateway means that activities B and C can be executed in parallel, whereas by using the exclusive gateway, we express that activities E and F are executed alternatively, i.e., either E or F is executed for a given process instance, depending on the conditions of the process. Note that BPMN offers further gateways such as inclusive or complex gateways. Using exclusive

---

<sup>4</sup><http://www.bpmn.org/>.



**Fig. 2.3** Example process model in BPMN notation; selected elements in BPMN 2.0

gateways can be used to model cycles in the BPMN model. Note that this constitutes an extension to the basic series-parallel graph structure.

Finally, BPMN offers a rich set of events that are thrown or caught during process execution. In the following, we solely use events *start* and *end* which indicate that the process is started or finished, respectively.

Altogether, the activities, gateways, as well as the start and end events are elements that constitute the *control flow* of the business process reflected by the corresponding BPMN model in Fig. 2.3.

On top of the control flow, we can model the (business process) *data flow* of the business process within the corresponding BPMN model. The basic constructs are *data objects* and *data associations*. Process data is written and read from the data objects. If an activity writes a data element, a data association is drawn from this activity to the corresponding data object, and if an activity reads a data object, we draw a data association from the data object to the activity. In the process model shown in Fig. 2.3, activity B writes data object d and activity E reads d, respectively.

In order to capture the organizational perspective, BPMN employs the so-called *swim lane* notation. For each organizational role that is active in the process, a (swim) lane is drawn and the associated activities are placed within this lane. Lanes can be bundled within *pools* that reflect organizational units. The example depicted in Fig. 2.3 includes two lanes for Role1 and Role2, where Role1 is associated to activities A and D and Role2 is associated to activities B, C, E, and F.

For further details on BPMN with focus on visualization of process models, see Sect. 4.2.

## Petri Nets

Petri nets are bipartite graphs that are useful to describe business processes. Basically, distinct classes of Petri nets can be distinguished that differ with respect to their expressiveness. In this book, we use Petri nets as formally defined in [36].

### Definition 2.1 (Petri Net, Based on [36])

A Petri net  $P$  is defined as  $P := (T, P, E)$  with  $T$  being a set of *transitions*,  $P$  being a set of *places*, and  $E \subseteq (T \times P) \cup (P \times T)$  being a set of edges where  $E \neq \emptyset$  and  $T \cap P = \emptyset$ .

Further, let  $\bullet t := \{p \in P \mid \exists(p, t) \in E\}$  be the preset of a transition  $t \in T$  and  $t\bullet := \{p \in P \mid \exists(t, p) \in E\}$  be the post-set of  $t \in T$ , respectively.

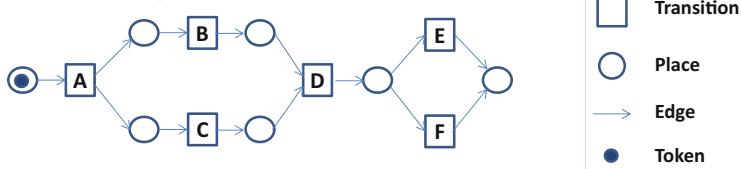
Transitions are the active elements of the net and are typically used to model process activities. Places are passive elements and represent states the business process can be in. Note the correspondence to the event view (event produced by executing transitions) and to the state view (places representing execution states the business process can be in) as introduced as BI views in Sect. 1.2.3.

From the definition of the node set  $V = P \cup T$ ,  $P \cap T = \emptyset$ , and the edge set  $E$ , we can see that a Petri net is a bipartite graph  $G = (V, E)$ . Specifically, it is only allowed to connect transitions with places and places with transitions.

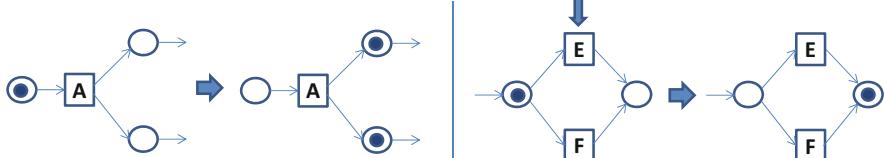
Figure 2.4 depicts an example process model represented by a Petri net. The Petri Net consists of six transitions, A, B, C, D, E, and F, and seven places. All edges connect transitions with places and vice versa.

The dynamic behavior of Petri nets is realized by so-called *tokens* that are consumed and produced by transition executions. More precisely, a transition  $t$  is *enabled*, i.e., it can *fire*, if all places in the preset of  $t$  carry a token and if all places in the post-set of  $t$  do not carry a token. Then, if  $t$  fires, all tokens from places in the preset are consumed, and, for each place in the post-set, one token is produced, respectively.

Process Model Represented as Petri Net:



Firing of Transitions (Examples):



**Fig. 2.4** Example process model in Petri net notation

When compared to the process model in BPMN notation (cf. Fig. 2.3), we can see that Petri nets (in their basic class) do not offer explicit gateway constructs. In fact, they formulate, for example, parallel or exclusive gateways using transitions and places.

In the example shown in Fig. 2.4 (at the left side), transition A serves as a parallel gateway. A is enabled at its preset place and carries a token, whereas the two post-set places do not carry a token. When A is firing, the token in the preset place is consumed, and, for each of the places in the post-set, one token is produced. The semantics is that by firing A, two threads of control are activated, and subsequent transitions can fire in arbitrary order. Contrary to A, the pre-set place of transitions E and F serves as an exclusive gateway (to the right side). As it carries a token, both subsequent transitions E and F are enabled. This means that either E or F fires, but not both of them. Based on the model, it cannot be decided which transition will fire (non-determinism [35]). Within a process-oriented implementation, this decision would be based on a decision rule and process data.

A special class of Petri nets that are particularly suitable for modeling and executing business processes are *WorkFlow nets (WF Nets)* as introduced in [45]. They offer certain modeling criteria and syntactic extensions such as explicit logical connectors for modeling, for example, parallel or alternative branchings.

### Event-Driven Process Chains (EPCs)

EPCs are widely used in practice and are the core process meta-model employed by the ARIS<sup>®</sup><sup>5</sup> tool family. As for Petri nets, EPCs offer two basic modeling elements, i.e., *functions* (cf. transitions) and *events* (cf. places), that are connected in an alternating way using control edges. By contrast to Petri nets, EPCs offer explicit logical connectors for denoting alternative and parallel branching. For more details, we refer to [34].

For further details on modeling and visualizing processes with Petri nets and EPCs, respectively, we refer to Sect. 4.2.

#### 2.3.3 *Summary: Graph Structures*

Graph theory offers a powerful tool for modeling in BI. The semantics of a graph can be translated into domain semantics about relations between objects of the business process in an intuitive way. For graphs, one can define a number of generic model structures which are of interest in BI. In particular, series-parallel networks, bipartite graphs, and trees are important structures. Moreover, one can formulate a number of generic questions about properties of graphs, for example, finding the shortest path, a spanning tree with minimum costs, or the maximum capacity of a network. Such questions are of interest in many applications. In this section, we introduced BPMN

---

<sup>5</sup><http://www.softwareag.com/corporate/products/aris/default.asp>.

and Petri nets as two applications which are of utmost importance in the case of modeling the production perspective.

Finally, let us mention the dual representation of graphs as an important advantage of graph models. On the one hand, the visualization of graphs is very well suited for communication; on the other hand, the matrix representation supports the application of linear algebra for algorithmic solutions.

## 2.4 Analytical Structures

Analytical structures are the most important model structures for BI regarding data in the cross-sectional view. Modeling combines concepts of (real-valued) functions, probability, and statistics. Accordingly, we first present the basics on these three structures and then show how they are combined in modeling. We assume that the reader is familiar with mathematics, probability, and statistics at the level of introductory courses for applied computer science or business administration. There are numerous textbooks covering these topics. Let us mention only [26] as a classical introduction into probability and statistics using only basic mathematics, Mendenhall et al. [33] as a traditional textbook in probability and statistics, and [11] for an approach more oriented toward computational statistic.

### 2.4.1 Calculus

#### Language

The concept of a function is probably the most powerful modeling tool from mathematics. It allows the representation of relations between different variables in a quantitative way. In the following, we prefer the traditional mathematical term *variables* instead of the term *attributes* used in computer science. For example, if we want to make a statement about the development of the state of health during a treatment, we quantify the state of health by some measurement and define a function which shows its development over time. In the case of business applications, an example for using functions is modeling customer preferences for a certain product in dependence of its properties and customer characteristics like age or sex.

The basic syntax and notation of mathematical functions is rather simple. We use the generic symbol  $f : X \rightarrow Y$  for denoting a function from a domain  $X$  into a range  $Y$ . In most applications, the domain is a subset of the  $p$ -dimensional vector space  $\mathbb{R}^p$  and the range is a subset of the real numbers  $\mathbb{R}$ . We will use the notation

$$y = f(\mathbf{x}) = f(x_1, \dots, x_p) \quad (2.4)$$

to show the dependence of the result of the application of the function on the vector  $\mathbf{x} = (x_1, x_2, \dots, x_p)'$ . In general, we understand vectors  $\mathbf{x}$  as column vectors and the transposed vector  $\mathbf{x}'$  as row vectors. The arguments in the domain are called *independent variables*, *input*, or *explanatory variables*. The result of the application of the function is called the *dependent variable*, *output*, or *response*. In the first example mentioned above, health status is the dependent variable that should be explained by the independent variables `time` and parameters describing therapy decision. In the second case, we have the dependent variable `customer preference` and independent variables `price` and a number of `customer attributes`.

Real-valued functions allow the application of classical function calculus, i.e., we can do arithmetic operations with functions and also compose functions provided the domain and range of the functions allow composition.

### Model Elements and Generic Questions

In the following, we briefly introduce a number of important model elements.

#### *Elementary Functions*

In one variable, some well-known elementary functions are used as basic models for describing relations. In particular, polynomials, exponential function ( $\exp(x)$ ), the logarithm ( $\log(x)$ ), or trigonometric functions ( $\sin(x)$  and  $\cos(x)$ ) will be used. All these functions are continuous and differentiable functions.

#### *Vector Calculus*

In the case of  $p$ -independent variables, the values  $(x_1, \dots, x_p)$  define a row vector. This allows application of vector calculus, in particular addition and scalar multiplication of a vector, definition of linear combinations, and definition of the concept of linear dependency of vectors. Furthermore, we use the concept of the length or the *norm of a vector* defined by

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^p x_i^2}. \quad (2.5)$$

An important generic question is the *distance* between two vectors defined as the norm of the difference of two vectors

$$d(\mathbf{x}, \mathbf{z}) = \sqrt{\sum_{i=1}^p (x_i - z_i)^2} = \|\mathbf{x} - \mathbf{z}\|. \quad (2.6)$$

This distance is called Euclidean distance and is used in many applications. Besides the Euclidean distance, other distances are used in modeling for special purposes and will be introduced in connection with different analytical techniques.

### Linear Functions

For modeling the relation between a response variable and a number of explanatory variables, linear functions play an important role. Given a vector  $\mathbf{w}$  of coefficients, the *linear function* is defined by the inner product:

$$f(\mathbf{x}) = \mathbf{w}'\mathbf{x} = w_1x_1 + \cdots + w_px_p. \quad (2.7)$$

The coefficients  $w_1, w_2, \dots, w_p$  define the importance or weight of the variables  $x_1, x_2, \dots, x_p$  for the output. Generalizations of the inner product are linear functions from a  $p$ -dimensional space into a  $k$ -dimensional space, which can be written in matrix notation in the form  $f(\mathbf{x}) = W\mathbf{x}$ , where  $W$  is a  $k \times p$  matrix of coefficients.

### Projections

A special case of linear functions are orthogonal projections of a vector  $\mathbf{x}$  onto a  $k$ -dimensional subspace. The orthogonal projection of a vector  $\mathbf{x}$  onto another vector  $\mathbf{w}$  is defined by

$$p_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}' \frac{\mathbf{w}}{||\mathbf{w}||} \quad (2.8)$$

In Chap. 4, we will use projections of observations onto a number of orthogonal directions for displaying observations in a lower dimension.

Projections are also the basic model element in regression. A  $p \times k$  matrix  $A$  with  $k < p$  and  $k$  linear-independent columns defines a  $k$ -dimensional subspace of  $\mathbb{R}^p$ . The projection onto this subspace is defined by the matrix  $P = A(A'A)^{(-1)}A'$ . Here,  $(A'A)^{(-1)}$  denotes the inverse matrix, i.e.,  $(A'A)^{(-1)}(A'A) = I$ , with  $I$  being the identity matrix.

### Kernels

Kernels play an important role for a number of analysis techniques, in particular, in BI applications for nonstandard data objects like graphs. We will present here only the basic ideas for vector-valued observations and refer the interested reader to a more formal exposition in [46].

The first application of kernels is in connection with measuring the similarity between observation objects. A number of analysis methods for vectors of observations are based on the calculation of the distance or similarity between two observations using the Euclidean distance. This distance can be calculated easily by the inner product of the vectors:

$$d(\mathbf{x}, \mathbf{y})^2 = \mathbf{x}'\mathbf{x} + \mathbf{y}'\mathbf{y} - 2\mathbf{x}'\mathbf{y}. \quad (2.9)$$

If one wants to transform this linear method for calculating similarity to a nonlinear measurement of similarity, a convenient way is the so-called *kernel trick*. This means that we define a function  $k(\mathbf{x}, \mathbf{y})$  for measurement of the similarity

between the two vectors. Using the kernel allows the application of the standard computations with no additional costs. A frequently used kernel for vector-valued observations is the *radial basis kernel*

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{d(\mathbf{x}, \mathbf{y})^2}{2\sigma^2}\right) \quad (2.10)$$

where  $\sigma$  is a parameter and  $d$  is the Euclidean distance. This kernel defines a measure of similarity: the kernel decreases with the distance of the vectors and attains the value 1 in case of  $\mathbf{x} = \mathbf{y}$ . We will use these ideas in Chap. 5.

The second application of kernels is the regularization of observation objects. Usually, the data represent only a finite number of observations from a function  $f(\mathbf{x})$ . This allows only a discrete representation of the function. Kernels allow a smooth representation of the function defined by the observations and will be considered in Sect. 5.2.4.

### *Optimization of Functions*

Besides the generic question about the distance between objects, questions about specific properties of functions are of interest. A frequently occurring question in modeling is finding *optimal values* of a function  $f(\mathbf{x})$  which describes a phenomenon. We use the notation  $\min_{\mathbf{x}}\{f(\mathbf{x})\}$  or  $\max_{\mathbf{x}}\{f(\mathbf{x})\}$  for the values of the minimum and the maximum of the function. The values where the function attains its minimum and maximum are denoted by  $\arg \min(f(\mathbf{x}))$  and  $\arg \max(f(\mathbf{x}))$ . Depending on the assumptions about the function, calculus offers analytical techniques for answering such questions. Minimizing and maximizing functions under constraints in the context of applications is a major topic of operations research. The interested reader is referred to [25] for an application-oriented exposition of this topic.

## 2.4.2 *Probabilistic Structures*

Probabilistic structures are important for capturing model variability as discussed in Sect. 2.1.3. Let us briefly review the notation and some important basic facts.

### **Language**

Probabilistic structures are of interest if we want to analyze business processes which show some random behavior. For example, a decision about the next event in a sequence of events is not based on a deterministic rule but on a random selection between a number of alternatives.

### *Basic Concepts of Probability*

The starting point for the development of the language of probability is uncertainty about the occurrence of an event  $E$ . We can define the probability that event  $E$

occurs as a number in the interval  $[0, 1]$  where  $P(E) = 0$  indicates that  $E$  is an impossible event and  $P(E) = 1$  that  $E$  is a certain event. Alternatively, we can use *odds* to define the probability. Odds state the chance for occurrence of event  $E$  against the chance that  $E$  does not occur. They are defined either in the form  $\text{odds}(E) = a : b$  or, more frequently, simply as a certain number  $a$  assuming that  $b = 1$ . The relation between probability and odds is defined by the equations

$$\text{odds}(E) = \frac{P(E)}{1 - P(E)}, \quad P(E) = \frac{\text{odds}(E)}{\text{odds}(E) + 1} \quad (2.11)$$

For example, if we are interested in the event that a customer quits the business relation, we can state that the event  $E = \{\text{customer quits}\}$  has the chance  $P(E) = 0.2$ . Equivalently, we could say that the odds are  $\text{odds}(E) = 0.2/(1 - 0.2) = 1 : 4 = 0.25 : 1 = 0.25$ , i.e., the customer quits in one out of five cases.

In BI, events of interest are the values of observed attributes of process instances. As in Sect. 2.4.1, we prefer the term variable instead of attribute corresponding to the idea that the values of attributes vary for the instances.

For modeling the random variation of a phenomenon, we use the concept of *random variable*. Random variables are denoted by capital letters  $X$  and the values of the random variable by  $x$ . The notation  $X = x$  is used for describing the event that the random variable  $X$  takes the value  $x$ . If there is only a finite number of possible outcomes, we can assign probabilities to all possible outcomes and apply the basic calculus of probability.

Of central interest in BI are random variables taking real numbers as values. For example, in business applications, we are interested in the event that the number of customers buying a certain product is above 1,000, or in medical applications, we are interested in the event that the blood pressure of a person takes values between 100 and 140. For describing such events of numerical random variables, we use the concept of the *distribution* of a random variable. The *distribution function*  $F(x)$  defines the probability that the random variable takes values less or equal to  $x$ . In formal notation, we write:  $F(x) = P(X \leq x)$ .

For the characterization of the distribution function, we use the concept of *quantiles*. For each value  $p$ ,  $0 \leq p \leq 1$ , the quantile  $q(p)$  is defined as the smallest value for which the distribution function takes the value  $p$ . In mathematical notation, we write  $F(q(p)) = p$ . Of special interest is the quantile for  $p = 0.5$  which is called *median*. The median splits the range of the values of the variable in two parts, such that 50 % of the values are smaller or equal to the median and 50 % of the values are larger than the median. Other important quantiles are the *quartiles* splitting the range of values of the variable into four parts: The first quartile defines the range for 25 % of the smallest values, the second quartile is the median, and the third quartile defines the value for which 25 % of the values are larger.

The distribution function characterizes the distribution of the random variable and allows computation of the probability of the event that a random variable  $X$  takes values in an interval ( $a < X \leq b$ ) by the equation

$$P(a < X \leq b) = F(b) - F(a). \quad (2.12)$$

Many a time, we want to know how likely the occurrence of the event  $E = \{X = x\}$  is. For answering these questions, two cases have to be distinguished. The first one concerns *discrete distributions* where the random variable can only have values out of a well-defined finite or countable set of numbers, often interpreted as counts of occurrences. In this case, it is meaningful to assign a probability  $p(x)$  to each possible value  $x$  of the random variable, i.e., for the event  $E = \{X = x\}$ . The second case regards *continuous distributions* where the possible values of the variable  $X$  may be any number. In such cases, it is not meaningful to talk about the probability of the event taking a specific value, but only about the probability that the value is within a certain interval  $[a, b]$ . In this case, we formally define a function  $p(x) = F'(X)$  and call it *probability density*. The values of  $p(x)$  are not strictly probabilities, but they give us information about the *likelihood* of the value  $x$ . As usual in applications, we will use the notation  $p(x)$  and the term likelihood for the discrete and the continuous case.

Besides the quantiles, many other characteristics of distributions are used in practice. Of special interest are the moments. We will use only the mean and the variance. The *mean* is defined as the expected (average) value of the random variable. Usually it is denoted by  $\mu$ . In the case of discrete probabilities, it corresponds to the average value of the random variable. The *variance* is defined as the expected squared deviation of the values of the random variable from the mean. The square root of the variance is called *standard deviation*. The symbolic notation is  $\sigma^2$  for the variance and  $\sigma$  for the standard deviation.

### *Joint Distribution, Independence, and Bayes Theorem*

In practical applications, we have to consider not only one variable but a number of variables. For describing the distribution of two and more variables, we introduce the concepts of joint distribution, conditional distribution, and independence.

The *joint distribution* of two random variables is defined by the joint probability distribution function according to the same principles as the distribution function of one variable:

$$F(x, y) = P(X \leq x, Y \leq y). \quad (2.13)$$

This function gives us the probability that the random variable  $X$  takes values less than or equal to  $x$ , and the random variable  $Y$  takes values less than or equal to  $y$ . The interesting question is the relation between the joint distribution of the two variables  $(X, Y)$  and the distribution functions of the two components  $F_X(x)$  and  $F_Y(y)$ , which, in this context, are called *marginal distributions*. The general relation between joint distribution and marginal distributions is obtained by the *conditional distribution* which is defined as the distribution of one variable given the value of the other variable.

### *Example 2.1 (Calculation of Conditional Distribution)*

For understanding the concept of conditional distribution, let us consider a simple example of customers using a forum. Customers are classified according to age as

Joint Probabilities			
Usage Pattern	Age Group		marginal
	young	old	
high	0.2	0.1	0.3
moderate	0.3	0.2	0.5
inactive	0.1	0.1	0.2
marginal	0.6	0.4	1.0

Conditional Probabilities given Usage			
Usage Pattern	Age Group		marginal
	young	old	
high	0.67	0.33	1
moderate	0.6	0.4	1
inactive	0.5	0.5	1
marginal	0.6	0.4	1.0

Conditional Probabilities given Age			
Usage Pattern	Age Group		marginal
	young	old	
high	0.33	0.25	0.3
moderate	0.50	0.50	0.5
inactive	0.17	0.25	0.2
marginal	1.00	1.00	1.0

**Fig. 2.5** Joint probabilities and conditional probabilities

young customers and old customers, and let us assume that 60 % of the customers are young and 40 % are old. This age classification defines the first random variable  $X$ . With respect to their activity in the forum, customers are classified as highly active (30 %), moderately active (50 %), and inactive (20 %). The activity pattern defines the second variable  $Y$ .

In practice, we are not only interested in the marginal distributions of the two variables but also in the probabilities of the combinations of age groups and activity patterns. The joint probabilities are shown in the inner cells of the left table in Fig. 2.5. The marginal distributions  $p_x(x)$  and  $p_y(y)$  are shown in the margin cells. Note that the joint probabilities correspond to total percentages in the table. If we are interested in the age distribution given the different usage patterns we have to switch from total percentages to row percentages. These row percentages are called the conditional distributions  $p(x|y)$  of age groups given the usage pattern  $y$  and are shown in the table in the center. The right table shows the conditional distribution  $p(y|x)$  of the usage pattern given the age groups. Obviously, the probabilities in the table at the right are obtained by normalizing the column probabilities in such a way that they sum to 1.

These intuitive calculations can be translated into the well-known formula for the conditional probability of an event  $A$  given an event  $B$ :

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \quad (2.14)$$

Figure 2.5 shows different relations between the events defined by the age groups and the events defined by the usage patterns. For example, in the case of the events  $A = \{\text{Usage Pattern} = \text{moderate}\}$  and  $B = \{\text{Age Group} = \text{young}\}$ , we obtain  $P(A \cap B) = 0.3 = 0.5 \cdot 0.6 = P(A) \cdot P(B)$ . We call such events *independent events*. In the case of the events  $C = \{\text{Usage Pattern} = \text{high}\}$  and  $B = \{\text{Age Group} = \text{young}\}$ , this relation does not hold, and we call the events *dependent events*.

The generalization of this example leads to the relation between conditional distribution, joint distribution, and marginal distribution formulated in terms of likelihoods:

$$p(y|x) = \frac{p(x, y)}{p_x(x)} \quad p(x|y) = \frac{p(x, y)}{p_y(y)}. \quad (2.15)$$

Obviously, these formulas only make sense if the denominator is not 0.

The generalization of the concept of independent events leads to the concept of *independent random variables*. The two random variables  $X$  and  $Y$  are independent if the conditional distribution coincides with the marginal distribution, i.e.,  $p(x|y) = p_x(x)$ . Equivalently, we can characterize independence by the fact that the joint distribution is the product of the marginal distributions, i.e.,  $p(x, y) = p_x(x) \cdot p_y(y)$ .

The idea that conditional distributions correspond to row percentages or column percentages in tables and the relation of these percentages to the joint distribution have an important consequence known as the *Bayes theorem*, formally stated by the formula

$$p(y|x) = \frac{p(x|y)p_y(y)}{p_x(x)}. \quad (2.16)$$

This formula can be interpreted in the following way. Suppose that the random variable  $Y$  represents the possible states of nature, sometimes called hypotheses, and the random variable  $X$  represents the observations. Suppose we know the probabilities  $p_y(y)$  for the different states of nature. These probabilities are called *prior probabilities* or *priors*. The conditional probability  $p(x|y)$  can be understood as the distribution of the observations given a certain state of the nature defined by the value  $y$  which are also known. Using the Bayes theorem, we can calculate the probabilities of the different states of the nature  $p(y|x)$  given the observation  $x$ . These probabilities are called *posterior probabilities* or simply *posteriors*. They can be used for the decision about the most plausible state of the nature. We will use this idea in Chap. 5 for prediction.

### Covariance and Correlation

For the measuring the degree of dependence between two variables, the concepts of covariance and correlation are used. If  $X$  and  $Y$  are two random variables with the means  $\mu_X$  and  $\mu_Y$ , the covariance is defined as the expected value of the product of the centralized variables  $(X - \mu_X) \cdot (Y - \mu_Y)$ . The correlation is defined by standardizing covariance according to the standard deviations of the variables  $X$  and  $Y$ . Correlation measures in how far a linear equation  $Y = \alpha + \beta X$  describes the relationship between the two variables. It is denoted by the symbol  $\rho$  and takes the values in the interval  $[-1, 1]$ . Corresponding to the sign of the slope of the line  $Y = \alpha + \beta X$ , we talk about positive relationships ( $\beta > 0$ ) and negative relationships ( $\beta < 0$ ). The values 1 and  $-1$  indicate that the linear relationship is correct, and a value of 0 means no relationship. The values in between are frequently classified with terms ranging from almost no relationship ( $|\rho| < 0.2$ ) up to strong relationship ( $|\rho| > 0.8$ ).

### Model Elements and Generic Questions

The basic model elements of probability structures are families of probability distributions. A family of probability distributions is defined by an indexed set of distributions  $\{P(x, \theta); \theta \in \Theta\}$ . The values of  $\theta$  are called parameters of the family.

The generic questions for probabilistic structures are concerned with properties of the distribution. Well-known questions are computation of quantiles, values of the parameters of a distribution like mean and variance, or questions about the distribution of transformations of random variables. Probability theory offers well-developed techniques for studying such questions. There is an abundance of distribution models, and the interested reader is referred to the monograph series of [32] that describes the models from a theoretical as well as an applied point of view. We will mention here only the binomial distribution as basic model for counts and the normal distribution as the most important model for measurements.

### *Binomial Distribution*

The *binomial distribution* is the standard model for counting processes. If we are interested in the occurrence of a specific event, for example, credit defaults, we want to know the probability that the event occurs  $k$  times in  $N$  trials. Provided that the probability of the event is the same for all trials and the event occurs in each trial independently from all other trials, the probability can be computed using the formula

$$p(k) = \binom{N}{k} \pi^k (1 - \pi)^{N-k} \quad k = 0, 1, \dots, N. \quad (2.17)$$

The probability  $\pi$  of occurrence of the event in one trial is the parameter of the distribution which defines the family of binomial distributions.

### *Normal Distribution*

The *normal distribution* is the most frequently used distribution if the random variable represents the results of a measurement. The application is justified in cases where we have the idea that there exists one central value for the random variable, all other values spread symmetrically around this value, and values far away from the center are rather unlikely. Such a model is justified either by the idea that the measurements are subject to error, or that there is some kind of natural variability in the observed phenomenon. Besides its practical importance as modeling tool, the normal distribution plays a central role in probability theory as an approximation of distribution of sums of random variables.

The normal distribution is usually defined by the probability density

$$\phi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}. \quad (2.18)$$

The parameter  $\mu$  is the mean characterizing the center of the distribution, and the parameter  $\sigma$  is the standard deviation modeling the variability of the values around the mean. These parameters define the family of normal distributions and allow the adaptation of the model to practical problems.

In case of more than one variable, the most important model is the *multivariate normal distribution*. Given normal distributed random variables  $X_1, X_2, \dots, X_p$

with known means and variances, the multivariate normal distribution can be characterized by using the covariances of the variables as additional parameters.

### 2.4.3 Statistical Structures

In BI, the connection between probabilistic models and empirical data is of utmost importance. This requires additional language elements defined in a statistical language, which is applied for modeling data in the cross-sectional view and the state view. As in the previous section, we will first introduce the basic concepts followed by model elements and generic questions.

#### Language

Empirical data of business processes usually contain data for well-defined entities generating the business process instances. In statistical terminology, we call the entities *statistical units* or *observation units*. The collection of all statistical units in the data is called *sample*. Although we nowadays collect numerous entities, it is not realistic to assume that available data cover all possible entities. For example, collecting information about the opinion of customers from entries in a forum will hardly reflect the opinion of all customers. Consequently, we have to introduce a concept which defines a reference for the entities in the sample. In statistics, this reference is defined by the term *population* which is specified by spatial and temporal reference, together with a number of other attributes. In the case of the customer perspective, such attributes may be age or education; in the case of the production perspective, the size of an enterprise or the economic activity could be attributes defining the population. Temporal and spatial reference is important, because a population always has a temporal dynamic, and changing the spatial reference is many times of interest. The considerations about the population are of utmost importance, because BI aims at learning from data about the properties of the population for further applications, and the evaluation of the results of a BI activity has to be done with respect to this population. For example, in the customer perspective, we want to make predictions about new customers.

The statistical units in the sample data inform about a number of attributes. The observed attributes are interpreted as values of random variables and are called *observable variables*. The interpretation of the attributes as random variables is important for data in the cross-sectional view and in the state view. In the cross-sectional view, the interpretation as a random variable reflects the fact that there is always some kind of uncertainty about the values of a variable for the statistical unit. For example, it seems unrealistic to know everything about customer motivations or to keep track of all parameters influencing the health status of a patient. In the state view, looking at the attributes as random variables allows the modeling of temporal variability, for example, changes in the number of sold products in time or changes in the health status of a person.

The concepts sample, population, and observable variables allow empirical modeling using probabilistic structures by transferring concepts of probability distributions to the sample. Instead of the distribution of a random variable, we use the concept of an *empirical distribution* of the sample data, also called *frequency distribution* or *sample distribution*. Accordingly, all the other concepts are transferred using the term *empirical* as prefix. There are some standard rules for transferring the probabilistic notation into a statistical notation. For distributions, a default rule adds a subscript  $N$  to the symbols which indicates the number of observations, and for parameters, the usual distinction is to use Latin letters for the parameters in the sample and Greek letters for parameters of a distribution.

## Model Elements and Generic Questions

With respect to statistical model elements, one usually distinguishes between descriptive elements and inferential elements. In the case of data in the cross-sectional and the state view, descriptive model elements play an essential role in BI for descriptive analysis goals. These models allow summary descriptions and visualizations of the data. We will treat them in detail in Chap. 4. The most important inferential model elements are techniques used for answering three generic questions about the properties of statistical models. The first one is an estimation of a probability distribution from the empirical distribution, the second is about the assessment of the reliability of an estimate of a distribution, and the third is about testing hypothesis about a statistical model. In the following, we briefly describe these techniques.

### *Statistical Estimation*

Usually, a probabilistic model is defined as an element of a certain distribution family with a number of unknown parameters. Estimation allows the determination of the parameter in such a way that the model is in agreement with observations. Classical examples are the determination of the probability of an event (i.e., the parameter  $\pi$  of a binomial distribution) or the mean and variance of a normal distribution.

Statistical estimation theory offers a number of procedures for solving this task, and mathematical statistics provides the theoretical background and properties of the procedures under some regularity assumptions. The standard assumptions are that the observation units generate the data independently from each other, and that these data represent a sample of random variables with a distribution defined by a family of distributions. A frequently used method for estimation is *maximum likelihood estimation*. This method first specifies the likelihood of the observations, and afterwards, it determines the unknown parameters of the model in such a way that the likelihood of the observations is maximized. For example, in the case of estimating the mean of a normal distribution, the empirical mean can be justified as the maximum likelihood estimator. The most important method for measuring the accuracy of an estimate is the *standard error* defined as the standard deviation of the estimate.

### Confidence Regions

Due to random variability of data, the results of an estimator will vary from sample to sample, and we cannot expect that estimated value gives the exact “true” value of the parameter. Therefore, one is interested in measuring the reliability of the parameter. This can be done by using a *confidence interval*. A confidence interval defines a region that covers the true value of the parameter for different samples in a proportion of samples defined by the confidence level. Usually, one takes as confidence level the value 0.95, which means that we can be confident that if we repeat the calculation with different samples from the same population, 95 % of all calculations will lead to an interval that contains the exact value. Obviously, such procedures assume that the specification of the model is correct. We will discuss the problem of model specification in detail in Chap. 5.

In the case of more complex problems, the idea of the confidence interval has to be transferred to *confidence regions* for vectors of parameters or to *confidence band* for functions. This is technically often demanding, but the interpretation that the region or the band covers the exact values with chosen confidence is the same as in the case of one real-valued parameter.

### Statistical Tests

Statistical tests offer procedures for deciding between two different hypothesis about reality called *null hypothesis* and *alternative hypothesis*. For example, if we want to know whether two different treatment procedures for patients lead to the same results, we formulate the null hypothesis that the two procedures lead to the same result and the alternative hypothesis that the two procedures lead to different results. For answering such questions, statistical significance tests give a decision according to the steps shown in the overview box.

Note that this procedure is asymmetric with respect to the role of the null hypothesis and the alternative hypothesis: for the null hypothesis, the error of wrong decision against the null hypothesis is controlled by the significance level, but no statement is made about the alternative hypothesis. In BI, such an asymmetry is many a time not sufficient, and we will discuss in Chap. 5 how we have to modify the ideas in BI applications.

#### Principle of Significance Tests

1. Formulate null hypothesis and alternative hypothesis.
2. Compute a test statistic from the data.
3. Compute the *p*-value as the probability of how likely the result of the test statistic is under the null hypothesis.
4. If the *p*-value is smaller than a predefined significance level, usually <0.05 or 0.01, the null hypothesis is unlikely, and we decide against the null hypothesis. Otherwise, we decide to keep the null hypothesis.

Statistical tests resemble the decision of a judge about a defendant using only empirical evidence. The null hypothesis is the assumption that the defendant is

not guilty. A small  $p$ -value indicates overwhelming evidence that the defendant is guilty. Only in a small proportion of decisions (5 % or 1 %) empirical evidence would lead to a wrong condemnation of a person who is not guilty.

#### 2.4.4 Modeling Methods Using Analytical Structures

In this section, we will introduce Markov chains as an important example of probabilistic modeling and linear models as a frequently used statistical modeling technique.

##### Markov Chains

Markov chains are useful models if one takes the state view on the business process, i.e., we are interested in events interpreted as states. The transitions between states are not defined by logical rules but follow random decisions. A well-known example is the Web surfing behavior of customers. There are no deterministic rules which determine a “correct” sequence of the visited sites. More realistic is a model which assumes that the visited sites are defined by random decisions of customers.

For the representation of such a process, we interpret the instances of the process as realizations of a discrete time stochastic process observed at times  $t = 0, 1, \dots, T$ . At each time  $t$ , we observe a certain state of the process which is modeled as the outcome of a random variable  $S_t = s_j$ . Furthermore, we assume that the number of possible states is finite, and we denote the states by an index set  $\mathcal{S} = \{s_1, s_2, \dots, s_M\}$ . In the example of Web surfing, the states would be the possible sites a customer can visit, and we observe the visits of  $T$  sites of a customer starting at the time  $t = 0$  at some site  $s_{j0}$ . Note that multiple visits are possible. A stochastic process in discrete time with a finite number of states is also called a *chain*, and an instance of the process  $s = (s_{j0}, s_{j1}, \dots, s_{jT})$  is called a *(process) path*. For describing the path behavior, we use the conditional probabilities that the system is in state  $s_j$  at some time  $t$ , given the past states  $(s_{j0}, s_{j1}, \dots, s_{jt-1})$ . In order to make calculations easier, we assume that the *Markov property* holds and define a *Markov chain* as follows:

##### Definition 2.2 (Markov Chain)

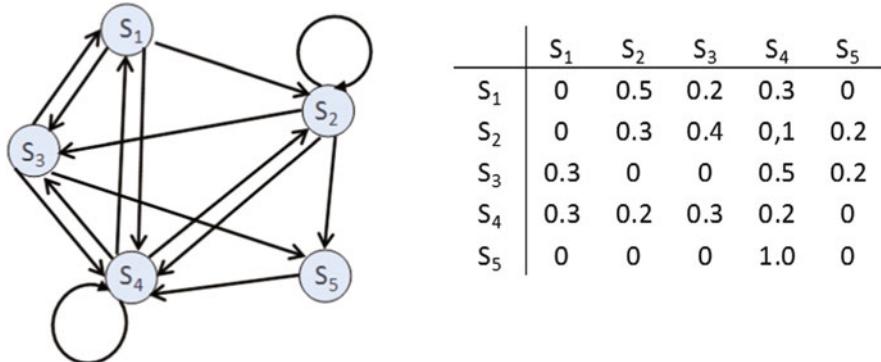
A discrete time stochastic process with finite states is a Markov process if for the conditional probabilities of state transition the following equation holds:

$$P(S_{t+1} = s_j | S_0 = s_{i0}, S_1 = s_{i1}, \dots, S_t = s_i) = P(S_{t+1} = s_j | S_t = s_i).$$

The conditional probabilities  $p_{ij}(t) = P(S_{t+1} = s_j | S_t = s_i)$  are called *transition probabilities*.

A Markov chain is a *stationary Markov chain* if the transition probabilities are independent of time:

$$P(S_{t+1} = s_j | S_t = s_i) = p_{ij}.$$



**Fig. 2.6** A Markov chain and matrix of transition probabilities

The matrix  $P = (p_{ij})_{i=1 \dots M}^{j=1 \dots M}$  has only nonnegative entries with  $\sum_{j=1}^M p_{ij} = 1$  and is called *stochastic matrix*.

The Markov property assumes that the probability for switching from state  $s_t$  in state  $s_{t+1}$  depends only on the actual state. This means that the behavior of the process is completely determined by the matrices of conditional probabilities  $P(t) = (p_{ij}(t))$  at different times and the probabilities for different states at time  $t = 0$ . In the case of stationary Markov chains, the representation is even more simplified, because we need only one transition matrix. Stationary Markov chains allow a representation of the process as a directed graph where the labels on the edges are interpreted as transition probabilities. Figure 2.6 shows an example of a Markov chain with five states and the corresponding Matrix of transition probabilities. A path in the chain could be  $(s_1, s_2, s_2, s_4, s_3, s_5, s_4)$ . If we assume that the initial probability for being in state  $s_1$  is  $P(s_1) = 0.7$ , then the probability of this path would be  $0.7 \cdot 0.5 \cdot 0.3 \cdot 0.1 \cdot 0.3 \cdot 0.2 \cdot 1.0 = 0.00063$ .

Let us mention that an alternative representation of Markov chains is using stochastic Petri nets, but the representation given here is often preferred in practice and has some advantages. In particular, readability is much higher. In Chap. 6, we will present analysis techniques for Markov chains and more complex models derived from this basic model.

### Statistical Modeling

A standard method in statistical modeling is combining function structures with probabilistic structures according to the three modeling topics outlined in Sect. 2.1.3 and defining a model by the equation

$$Y = f(\mathbf{X}) + \varepsilon. \quad (2.19)$$

The variable  $Y$  is often called output or response variable and represents the variable of interest, for example, a KPI, for which we seek an explanation in the analysis.

Usually, it is some characteristic of the business process in the cross-sectional view or a state variable depending on time in the state view. For the explanation of this variable, we use a model configuration which describes the mean behavior of  $Y$  in dependence of a vector of explanatory variables or input variables  $\mathbf{X} = (X_1, X_2, \dots, X_p)'$ . Also for these variables, we use data in the cross-sectional view or in the state view. In the case of the state view, one of the explanatory variables is time. The variable  $\varepsilon$  represents the noise or variability of the data, which cannot be explained by the explanatory variables.

The function  $f$  describing the relationship between the output variable  $Y$  and the explanatory variables  $\mathbf{X}$  is a model element from the function structures introduced in Sect. 2.4.1. Different specifications are possible and often characterized by the terms *parametric models* and *nonparametric models*:

### Types of Regression Models

- *Parametric linear models*: These models define a vector space of functions which allow the representation of the model as a linear combination of known basic functions. The best known example are models, where the dependence of the output is modeled as a linear function in the explanatory variables.
- *Parametric nonlinear models*: The model is defined by a nonlinear function depending on some parameters. Typical examples are exponential functions or trigonometric functions.
- *Nonparametric models*: Nonparametric models use large classes of functions for the modeling of the relationship between input and output. These functions cannot be described by a finite number of basic functions. Typical examples are differentiable functions with smoothness properties defined by conditions on the derivatives of the function  $f$ .

Sometimes, transformations of the variables are a necessary prerequisite for the simplification of the relation between input and output. A well-known example are *generalized linear models* which, instead of a linear relation for the mean of the response variable, define a linear relation between a function of the mean of the response and the explanatory variables. Such models are of interest in BI for classification tasks (cf, Sect. 1.2.4) where the response variable takes only the values 0 and 1 indicating a group or the occurrence of an event  $E$ . In this case, we are interested in modeling the probability of the occurrence of the event and define a linear model for the logarithm of the odds for the occurrence of the event:  $\log(\text{Odds}(E)) = f(\mathbf{X}) + \varepsilon$ . This model known as *logistic regression* will be treated in Chap. 5.

Other applications of transformation are the definition of new explanatory variables from the existing ones. A typical example is the idea of how to model the dependency of the response on interactions between the two explanatory variables  $X_j$  and  $X_k$ . A standard method is the definition of a new explanatory variable  $X_{jk} = X_j \cdot X_k$  as a new basic element and add this variable to the explanatory variables.

Another issue is the treatment of nonnumeric variables in functions which are defined only for real-valued variables. The standard method realized in analysis software is the introduction of dummy variables for the different values of a non-numeric variable. A *dummy variable* is defined as a variable taking only the values 0 and 1:

$$X = \begin{cases} 1 & \text{if condition } E \text{ holds} \\ 0 & \text{otherwise.} \end{cases} \quad (2.20)$$

Given a vector of dummy variables  $\mathbf{X} = (X_1, X_2, \dots, X_p)'$  defined by conditions  $E_1, E_2, \dots, E_p$ , a function  $f(x_1, x_2, \dots, x_p)$  assigns the output values according to the pattern of the holding conditions. If the conditions define a partition of all possible states, one needs  $p - 1$  dummy variables for describing the outcome. For example, assume that in connection with grading students, there are three conditions defined by `good`, `passed`, and `failed`. We define two dummy variables  $(X_1, X_2)$  for the conditions `good` and `passed` and the domain by  $X = \{(0, 0), (1, 0), (0, 1)\}$ . A function defined on  $(X_1, X_2)$  can represent all three conditions:  $f(1, 0)$  gives the result of the function under the condition `good`,  $f(0, 1)$  gives the result of the function under the condition `passed`, and  $f(0, 0)$  under the condition `failed`.

### 2.4.5 Summary: Analytical Structures

Analytical structures play a central role in models for the customer perspective. They combine elements from calculus, linear algebra, probability, and statistics and are the most important modeling technique for data in the cross-sectional view or the state view. Calculus allows the formulation of models in terms of equations which describe the behavior of an output variable in dependence of a number of input variables. Moreover, calculus is of central importance for generic questions about functions like finding the minimum or the maximum of a function. These analyses allow the formal treatment of questions about the optimal behavior of the business process.

Vector calculus is of special interest because it describes formally the concept of similarity between observations. An important class of functions are linear functions which define many a time intuitive models for the dependence of the output from a number of input variables.

Probability is the standard technique for the description of the variability of the data. The most important model elements in probability are distributions which characterize the uncertainty in terms of a distribution function. The two most important distribution functions are the binomial distribution for counts and the normal distribution for quantitative variables. For modeling the variability of a number of random variables, the concepts of joint distribution and conditional

distribution were introduced. In decision problems, conditional distributions are frequently used for the formulation of the dependence of a model on the state of nature. If the prior distribution of the different states of nature is known, Bayes' theorem allows the calculation of the posterior distribution of the possible states of nature given the observations.

Statistical models build the bridge between the observed reality and probabilistic models. The central elements of a statistical model are the observation unit, the population, and the observed values of the random variable. Statistical inference analyzes the connection between a probabilistic model and observations using the three techniques estimation, confidence intervals, and statistical tests.

Finally, Markov chains and regression models were introduced as modeling methods using probabilistic and statistical structures.

## 2.5 Models and Data

Besides data provisioning, data generation and data quality play an essential role in BI. In this section, we discuss the important aspects of data generation, the temporal aspect of data, and data quality.

### 2.5.1 Data Generation

A good deal of data understanding in BI requires knowledge on how the data used in the analysis are generated. An interesting and comprehensive presentation about current ways of data generation can be found in [21]. A more statistical oriented approach can be found in [48]. Here, we will concentrate on two issues in data generation. The first is data collection and the second concerns the measurement of variables.

#### Types of Data Collection

Most data used in BI are so-called *secondary data* from internal and external sources collected for other purposes than usage in BI applications. Drawing conclusions from BI applications based on such data requires their interpretation as a sample of a population to which the conclusions apply.

Data from internal sources are in most cases the result of administrative routine collections of the business process. Let us consider some examples.

#### *Example 2.2 (Examples of Internal Secondary Data)*

- a) In medical applications like the EBMC<sup>2</sup> use case (Sect. 1.4.1), such routine data inform about the staff in the hospital, the patients' basic demographic information, the number and time of visits of patients, or the notification of applied treatments.

- b) In the case of business applications like the CRM use case introduced in Sect. 1.4.4, routine data like the sales to customers or the number and time of visits in a shop are collected mainly for accounting purposes. Basic information about customers are collected in connection with issuing a customer card.
- c) In the HEP use case (cf. Sect. 1.4.2), all data about students, registration, grading, and activities in the forum are examples of routine data collection for administrative purposes.

Besides internal secondary data, external data are used for understanding the business under consideration in a larger context. Such data may be information about the demographic structure of a potential customer population, data about the economic situation, or data about competitors in the business. In the EBMC<sup>2</sup> use case, an example of such data derives from the Austrian cancer registry about the occurrence of skin cancer in the population. These data are structured according to the patients' sex, age at the time of diagnosis, and survival time. External data are collected by various public agencies like statistical offices, commercial data suppliers, or trade associations. Another important source for external data is the Internet, in particular social networks.

In case of insufficient information for the analysis purpose in secondary data, one has to collect data for the analysis by using surveys or observational studies.

Surveys collect information about variables of interest from observation units sampled from a population according to selection rules for the units. In business applications, a typical example of surveys occurs if one is interested in the opinion and attitudes of customers. The survey mentioned in the CRM use case in Sect. 1.4.4 is an example of such kind of data generation. There exists a well-developed methodology for collecting such information starting from the selection of the units from the population over methods of data collection up to the analysis and presentation of the data. The interested reader is referred to the literature, for example, [16].

If we have a specific business question in mind, a frequently used type of data collection are *observational studies*. In the case of observational studies, we have also the idea that the data are collected for units which represent a sample from a population, but additionally, data collection is more oriented towards a specific research question and a specific model structure for analyzing relationships between variables. Examples are data from monitoring process behavior according to a well-defined rule or recording a set of characteristics whenever an event occurs, for example, the registry at a counter in a supermarket. The data described in the pre-eclampsia use case in Sect. 1.4.1 or in the logistics use case in Sect. 1.4.3 are examples for observational studies. The distinction between observational data and survey data is not always strict, and one can use survey data in modeling in a similar way as observational data. In business applications, the term survey is more popular, whereas in medical or technical applications, observational studies are preferred.

All these methods of data collection allow the investigation of the relationships between different variables, but one should not mistake association for causality. For example, if we want to investigate the causal relations between a new treatment

and the effect of the treatment, it is not sufficient to randomly select the statistical units in the sample. Additionally, we have to randomly split the sample units into two groups. One group receives the new treatment and the other group is treated with a conventional method. The theory of *statistical experiments* describes designs for data collection allowing causal interpretation, but in BI applications, such experiments are rarely applied. The problem of causality in drawing conclusions from data is elaborated in detail in [37] and often requires additional understanding of empirical information based on domain knowledge.

### Methods of Measurement

Independent from the type of data collection, we have to think about the operational procedure for obtaining the values of the variables of interest. Due to the fact that all kinds of business activity are based on the interplay between humans and systems, two different types of measurement occur [20]. The first and most prominent one is *representational measurement* corresponding to the idea of scientific or technical measurement. For modeling purposes, we have to distinguish in this case between different scale types: nominal, ordinal, interval, and ratio scales. This classification is of utmost importance in business analysis for selecting appropriate analysis methods. The second way of measurement is the *operational measurement* characterized by the fact that the measurement device is not completely standardized and influences the measurement. Typically, operational measurement occurs in connection with surveying customer behavior in a business process using questionnaires or by evaluating the performance of a process based on indices or score. Behavioral sciences and social sciences have developed methods for the standardization and comparison of operational measurement, for example, a Likert scale for opinion using the five levels: strongly disagree, disagree, neither agree or disagree, agree, and strongly agree. Obviously, in modeling, such values can be interpreted as an ordinal scale in the sense of operational measurement, but in case of using such measurements from secondary data, one has to think about the used operationalization for correct interpretation of the results.

#### 2.5.2 The Role of Time

From a general point of view, all events and activities of a business process have to be seen as an occurrence in time. Hence, from a data-provisioning perspective, we should, in the ideal case, rely on temporal data-bases. This implies that from a formal modeling point of view, data models should be based on extensions of predicate logic for temporal modeling (cf. [14]). A well-known extension is *linear temporal logic* (LTL), which uses the following additional operators for propositions: next ( $X\phi$ ), meaning that proposition  $\phi$  holds at the next time point; always ( $G\phi$ ), meaning that the proposition ( $\phi$ ) holds at all following time points; eventually ( $F\phi$ ), meaning that proposition  $\phi$  holds at least at one following time point; until ( $\psi F\phi$ ), meaning that proposition  $\psi$  holds at least until  $\phi$

(which may hold now or in the future); and `release` ( $\psi R\phi$ ), meaning that proposition  $\phi$  holds until  $\psi$  becomes true (if  $\psi$  becomes never true,  $\phi$  is always true).

Using temporal logic in practice is often tricky and not always possible, because the measurement of time for real-life situations is often not easy. For example, in medical applications, we can precisely record the time points of prescription of a certain drug, but usually we have no control about the usage of the drug by the patient, and the duration of its effect is also not easy to measure. Such problems are known by the headings *granularities* (different measurement scales for time), *indeterminacy* (at what time the event of interest happened), *absolute* and *relative temporal occurrence* (sometimes time is defined relative to another event), *point* and *interval occurrence* (the event of interest has a duration shorter than the considered time unit), or *delays* (the effect caused by the event is instantaneous or has some delay). A detailed description of such problems in the medical context can be found in [10], but the problems seem similar in other application domains.

The handling of such problems in artificial intelligence has led to different proposals. A popular proposal is Allen's interval-based logic, which uses the time interval as primitives together with eight basic relations. Another example is the event calculus of Kowalski and Sergot using the event as primitives [10].

All these approaches have some deficiencies in BI applications; in particular, considering point events and interval events simultaneously is not easy and also handling the context of temporal information is not satisfactory, covered. An interesting approach which tries to overcome such deficiencies is the *Knowledge-Based Temporal Abstraction Method and Ontology* (KBTA) which has been implemented as the RÉSUMÉ system [42]. We will describe the approach following [10].

### Elements of the Knowledge-Based Temporal Abstraction Method

- *Time stamps*  $T_i$  are the basic primitives with a predefined granularity and a well-defined zero.
- Time intervals  $T = [T_{\text{start}}, T_{\text{end}}]$  are defined as pairs of time stamps for start and end. Time points are zero length intervals.
- An *interpretation context*  $\xi$  is a proposition that can change the interpretation of parameters within the scope of a time interval. Interpretation contexts can be nested.
- A *context interval*  $< \xi, I >$  defines time intervals for which the interpretation context holds.
- An *event proposition*  $e$  represents the occurrence of an external volitional action or process and has to be distinguished from a measurable datum.
- An *event interval*  $< e, I >$  represents the temporal duration of an event  $e$ .
- A *parameter schema*  $\pi$  is a measurable aspect of the state of the world (states of a process) with values in some domain  $v \in V_\pi$ . Parameter schemas may be of different types: primitive parameters (measurable data), abstract parameters (concepts), and constant parameters (instant specific or instant independent).

- A *parameter proposition*  $\langle \pi, v, \xi \rangle$  defines the values of parameters in a context.
- An *abstraction function*  $\theta \in \Theta$  maps parameters into abstract parameters.
- A *parameter interval*  $\langle \pi, v, \xi, I \rangle$  denotes the value  $v$  of the parameter  $\pi$  in the context  $\xi$  during time interval  $I$ .
- An *abstraction* is a parameter or a parameter interval.
- An *abstraction goal*  $\psi \in \Psi$  represents a specific intention or goal.
- An *abstraction goal interval*  $\langle \psi, I \rangle$  represents the idea that abstraction goal  $\psi$  holds in interval  $I$ .
- *Induction of context intervals* allows the induction of events, parameters, or abstraction goal propositions for some context interval.

In this description, we have used the original notation from [10], but note the similarity to the ideas on how we described the business process. Events correspond with the notion of event in the event view of the business process, event propositions represent the events caused by the actors in the process, and parameters correspond to the variables describing states in the state view. An abstraction can be understood as a transformation of the states, and the abstraction goal resembles the idea of a KPI. The important new element of KBTA is that it explicitly allows the definition of a temporal context and that this context may change the interpretation of the parameters. In BI, context has to be analyzed in the business and data understanding task of the *iMine* method (cf. Sect. 1.3.6).

In many cases, the main analysis goal consists of modifications of the interpretation of parameters. For example, if we are interested in understanding the interaction of different drugs, we can apply ideas of the KBTA approach. The event proposition is, in this case, the prescription of the drug. The time interval of interest is the duration of taking the drug of the patient, and the context interval is defined by the time interval in which the drug affects the health status of a person. This health status may be described by various parameters, and an abstraction of the parameters may be defined by the well-being of the patient. The parameters together with the context define a parameter interval. If we are interested in finding adverse drug events, the analysis question is whether the usage of a second drug changes the context in the parameter interval. Answering this question can be based on the induction of context intervals.

### 2.5.3 Data Quality

Data and information quality is a core topic in BI, and there are many approaches towards this topic ranging from theoretical considerations up to practical guidelines in different domains. The International Association for Information and Data Quality (IAIDQ) aims for advancing the quality of information in all kinds of businesses and offers a lot of material and examples of best practices [28]. We will take an application-oriented approach and briefly discuss the definition of data

quality, reasons for the lack of data quality, methods for improving data quality, and some organizational aspects for managing data quality.

### Definition of Data Quality

Using well-known ideas from quality management of other products, we start with definition of the term quality as given in the ISO norm:

“The totality of features of a product or service that fulfill stated or implied needs.”

The definition takes a user-centric view on quality. In our case, the products of interest are data and information which have to meet or exceed the customers’ expectations and requirements. The tricky thing with data quality is that the user-centric approach lacks a clear definition of the requirements of different users, and it seems practically impossible in data production to foresee all future users of the data. In fact, as we have argued in Sect. 2.5.1, BI uses, in most cases, secondary data and has only limited control about the production of data.

A popular way to handle the problem is the definition of so called *quality dimensions*, and numerous dimensions have been defined. If one looks at the literature about quality in connection with BI applications, the most important dimensions seem to be the following ones, which are also used by Eurostat for defining the quality of statistical products. The definitions used here are as far as possible derived from the IAIDQ glossary [28] or from Eurostat:

#### Quality Dimensions for Data

- **Relevance:** Relevance measures in how far the data are useful in the intended context.
- **Accuracy:** Accuracy is the degree of conformity of a measure to a standard or a true value.
- **Completeness:** Completeness is a characteristic measuring the degree to which all required data is known, with respect to depth, breath, and scope.
- **Timeliness:** Data coming early or at the right time, appropriate or adapted to the times or the occasion.
- **Consistency:** Consistency is expressed as the degree to which a set of data is equivalent in redundant or distributed databases.
- **Coherence:** Coherence refers to the adequacy of the data to be reliable combined in different ways and for various uses.
- **Reliability:** Reliability is a characteristic of an information infrastructure to store and retrieve information in an accessible, secure, maintainable, and fast manner.

### Reasons for the Lack of Data Quality

We discuss here reasons for the lack of data quality occurring in the first six quality dimensions. The quality dimension reliability is more an issue of the infrastructure for data provisioning and will be treated in Chap. 3.

Relevance of the data is an important factor in the case of using secondary data. The analyst has to decide whether the available data are really useful for the analysis question. In some sense, the concept of models of data described in Sect. 2.1.1 can be seen as an analytical task for evaluating the relevance of the data. Sometimes, a wrong specification of the values of an attribute may cause a problem of relevance. A frequently occurring problem is the relevance of temporal information. Time stamps in a database may refer to the *valid time*, i.e., the time point when the observed value of the attribute is the true value, or to the *transaction time* defined as the time point when the value is entered in the database.

Accuracy is probably the best understood quality dimension and can be measured in different ways. Well-known reasons for lack of accuracy are typing errors, missing values, or misspecification of attributes if they are used outside the limitations of its intended use. In case of survey data, the accuracy depends on the sample size and on the measurement instruments. For measuring accuracy in connection with the sample size, one can use standard deviations or standard errors of estimates. A typical example for lack of accuracy occurs frequently in connection with the specification of timescales. For example, if we use for the time stamps a resolution based on day, month, and year, we cannot define an order relation between events occurring at the same day.

The quality dimension completeness has to be discussed in connection with the definition of the population to which the data refer. In the case of surveys, high quality in completeness needs careful planning of the survey, starting from the specification of observational units, continuing with the definition of the attributes under consideration, and ending with the decision of appropriate measurement instruments and the measurement process. In BI applications, the design of a survey resembles many times the design of surveys for marketing purposes and requires many a time collaboration with domain experts. Wrong specification of observation units may cause errors in the dimension completeness, sometimes also called coverage of the data. Measurement problems for qualitative attributes are frequently caused by an unclear formulation of questions in a questionnaire. In the case of secondary data, definition of the underlying population is many a time rather difficult. For example, in the case of collecting data from the Internet, it is not easy to define the population to which the data refer.

For the quality dimension timeliness, it is important that we use data representing the actual state of the business. Timeliness may be a problem in the case of secondary data from external data sources. In such cases, control of the actuality of the information is frequently impossible. Issues of timeliness have to be seen many a time in connection with the relevance of the data and has to be discussed in connection with the role of time discussed in connection with the above-described knowledge-based temporal abstraction logic.

Consistency and coherence are quality dimensions which are of utmost importance in the case of combining multiple data sources. The basic problem is in this case, the identification of attributes which allow the matching of the information in the different sources. Besides problems of record matching, a frequently occurring reason for the lack of consistency are different values for the same attribute in

different databases. For example, a customer can have different addresses in two databases. Two obvious reasons are possible for such a difference: the entries in the databases were done at different times and the customer has changed the address or the customer has two different residences. Technical problems in connection with consistency and coherence will be discussed in Sect. 3.5.2.

### Methods for Improving Data Quality

Improving data quality is often called *data cleaning*. It aims at “detecting and removing errors and inconsistencies from data” [38]. Basically, there are two broad classes of methods. The first one is database oriented. A detailed treatment of techniques for ensuring data quality from the architectural perspective can be found in [4]. A special feature in combining data is record linkage in case of different representations or incompleteness of the key attributes. A detailed presentation of this problem showing different techniques can be found in [24].

The second approach is the statistically oriented approach. The monograph [12] considers the application of descriptive methods and data mining tools for assessing and improving data quality. An extensive discussion of the treatment of missing values and implausible data (outliers) can be found in [44]. A detailed typology of missing values for process data is given in [8].

### Managing Data Quality

Keeping this list of possible errors in mind, it is obvious that data quality management is a process, which has to accompany the entire life cycle of the data. From the organizational point of view it has to be treated as a main goal in the enterprise in an appropriate organizational framework [2]. In connection with data warehouses, this is called master data management (MDM) [39]. Consequently, we can define KPIs for data quality and implement methods for checking and ensuring data quality.

An important issue for all data quality considerations is knowledge about the data with respect to the different quality dimensions. A well-known way for handling such knowledge is via metadata if available. Besides the common understanding of the term metadata in database modeling, such metadata must encompass a conceptual description of the data, information on how the data were obtained, how they are managed, and in how far the data are checked with respect to criteria of consistency. Keeping in mind that data quality is the product of the entire life cycle of the data, it is necessary to accompany all steps of data production with the corresponding metadata production. A project which aims at such a documentation concerned with building a national census population from different data sources can be found in [5].

### 2.5.4 Summary: Models and Data

This section dealt with data from a modeling perspective. For modeling, it is important to know how the data were generated and which methods for measurement were used in data collection. With respect to data generation, one must be aware that BI frequently uses secondary data from internal and external sources which were collected for other purposes. This fact has to be taken into account in the interpretation of the analysis results.

Another important aspect in business and data understanding is proper understanding of the temporal dimension of the data. The knowledge-based temporal abstraction method was introduced as a reference framework for understanding the different aspects of time.

Finally, we stressed the importance of data quality for BI applications. For assessing data quality, a number of quality dimensions were defined and reasons for the lack of data quality were discussed. To ensure data quality, an appropriate framework for data quality inside an organization has to be established.

## 2.6 Conclusion and Lessons Learned

Modeling is a rather intricate activity in BI. One can use different approaches towards model representation and model presentation. Independent of the approach, three key topics in modeling can be identified. The first one is the definition of a model configuration. This mode model configuration depends on the business perspective of interest and can use different formal model languages. For representation of the domain knowledge, logical structures are useful. These structures can be implemented as ontologies, as frames, or as traditional data models. For capturing the production perspective of the business process, the most important model structures are based on graph theory. The Business Process Model Notation (BPMN) and Petri nets are two frequent models. In the customer perspective, analytical structures, in particular probability and statistics, are of utmost importance. Two frequently used models in this area are Markov chains and regression models. The second topic in modeling is the connection of the model configuration with the observations from business process instances. This requires understanding of the process of data generation, interpretation of the temporal information, and knowledge about data quality. The third topic is the formulation of a model component for capturing the variability of the different process instances. This issue plays a central role in models for the customer perspective which mostly use data in the cross-sectional view.

## 2.7 Recommended Reading (Selection)

This chapter covered a wide range of topics, and there exist many excellent text books for the different areas. For understanding the term model in a broad sense, we recommend Frigg and Hartmann (2009). Modeling is covered in many textbooks in connection with the different modeling methods. A good introduction of modeling aspects in connection with logical structures is Spies (2004). For modeling of business processes, we recommend Mendling (2008). Issues of modeling using analytical structures can be found in Hand, Manilla, and Smyth (2001) and Weiers (2005). The last two references treat also the role of data in connection with modeling.

- Frigg R, Hartmann S (2009) Models in science. Stanford encyclopedia of philosophy, <http://stanford.library.usyd.edu.au/archives/spr2009/entries/models-science/>
- Hand D, Mannila H, Smyth P (2001) Principles of data mining. The MIT Press, Cambridge, MA and London
- Mendling J (2008) Metrics for process models: empirical foundations of verification, error prediction, and guidelines for correctness. Lecture notes in information system processing, vol 6, Springer, Heidelberg
- Spies M (2004) Einführung in die Logik. Spektrum Verlag, Elsevier, München (in German)
- Weiers RM (2005) Introduction to business statistics. Thomson Learning, Belmont CA

## References

1. Abazi F, Bergmayr A (2009) Knowledge-based process modelling for nuclear inspection. In: Karagiannis D, Jin Z (eds) KSEM'09: international conference on knowledge science, engineering and management. Lecture notes in computer science, vol 5914. Springer, Heidelberg, pp 406–417
2. Apel D, Behme W, Eberlein R, Merighi C (2010) Datenqualität erfolgreich steuern. TDWI Europe, Hanser, Munich (in German)
3. Baader F, Horrocks I, Sattler U (2008) Description logics. handbook of knowledge representation, Chapter 3. Springer, Berlin/Heidelberg, pp 135–180
4. Batini C, Scannapieco M (2006) Data quality—concepts, methodologies and techniques. Springer Data-centric systems and applications. Springer, New York
5. Berka C, Humer S, Moser M, Lenk M, Schwerer E, Rechta H (2011) A quality framework for statistics based on administrative data sources using the example of the Austrian Census 2011. Austrian J Stat 39(4):299–308
6. Blaha M (2010) Patterns of data modeling. CRC/Taylor and Francis Group, Boca Raton, FL
7. Bodlaender HL, de Fluitter B (1996) Parallel algorithms for series parallel graphs. In: Díaz J, Serna MJ (eds) ESA'96: European symposium on algorithms. Lecture notes in computer science, vol 1136. Springer, Heidelberg, pp 277–289
8. Bose RP, Jagadeesh C, Mans RS, van der Aalst WMP (2013) Wanna improve process mining results? It's high time we consider data quality issues seriously. BPMcenter.org

9. BPMN (2011) Business Process Model and Notation (BPMN), Version 2.0, OMG Document Number: formal/2011-01-03, Standard document. <http://www.omg.org/spec/BPMN/2.0>. Accessed 24 June 2014
10. Combi C, Keravnou-Papailiou E, Shahar Y (2010) Temporal information systems in medicine. Springer, New York
11. Crawley MJ (2005) Statistics: an introduction using R. Wiley, New York
12. Dasu T, Johnson T (2003) Exploratory data mining and data cleaning. Wiley, New York
13. Frigg R, Hartmann S (2009) Models in science. <http://stanford.library.usyd.edu.au/archives/spr2009/entries/models-science/>. Accessed 21 March 2014
14. Galton A (2008) Temporal logic. The Stanford encyclopedia of philosophy (Fall 2008 Edition). In: Zalta EN (ed) <<http://plato.stanford.edu/archives/fall2008/entries/logic-temporal/>>. Accessed 5 May 2014
15. Grenander U (1996) Elements of pattern theory. John Hopkins University Press, Baltimore/London
16. Groves RM, Fowler FJ, Couper MP, Lepkowski JM, Singer E, Tourangeau R (2004) Survey methodology. Wiley, New York
17. Gruber TR (1993) A translation approach to portable ontology specifications. *Knowl Acquis* 5(2):199–220
18. Gruber TR (2008) Ontologies. In: Encyclopedia of database systems. Springer, New York
19. Guarino N (1998) Formal ontology in information systems. In: Int'l conference Frontiers in artificial intelligence and applications. IOS Press, Amsterdam, pp 3–15
20. Hand DJ (1996) Statistics and the theory of measurement. *J R Stat Soc Ser A* 159:445–492
21. Hand DJ (2007) Information generation. Oneworld Publication, Oxford
22. Hand DJ, Mannila H, Smyth P (2001) Principles of data mining. MIT, Cambridge, MA/London
23. Hepp M (2008) Ontologies: state of the art, business potential, and grand challenges. In: Hepp M, De Leenheer P, de Moor A, Sure Y (eds) *Ontology management, semantic web and beyond computing for human experience*, vol 7. Springer, New York, pp 3–22
24. Herzog TN, Scheuren FJ, Winkler WE (2007) Data quality and record linkage techniques. Springer, New York
25. Hillier FS, Lieberman GJ (2010) Introduction to operations research, 9th edn. McGraw-Hill Higher Education, New York
26. Hodges JL Jr, Lehmann EL (2004) Basic concepts of probability and statistics. Classics in applied mathematics, vol 48. SIAM, Philadelphia
27. Horrocks I, Patel-Schneider PF, Van Harmelen F (2003) From SHIQ and RDF to OWL: The making of a web ontology language. *Web Semantics* 1(1):7–26
28. International Association for Information and Data Quality (IAIDQ). <[iaidq.org/main/about.shtml](http://iaidq.org/main/about.shtml)>. Accessed 5 May 2014
29. Jungnickel D (1994) Graphen, Netzwerke und Algorithmen, 3rd edn. BI-Wissenschaftsverlag (in German)
30. Karagiannis D, Kühn H (2002). Metamodelling platforms. In: Bauknecht K, Tjoa AM, Quirchmayr G (eds) ECWeb'02: international conference e-commerce and web technologies. Lecture notes in computer science, vol 2455. Springer, Heidelberg, p 182
31. Karagiannis D, Grossmann W, Hoefferer P (2008) Open models—a feasibility study. Open models initiative. [http://cms.dke.univie.ac.at/uploads/media/Open\\_Models\\_Feasibility\\_Study\\_SEPT\\_2008.pdf](http://cms.dke.univie.ac.at/uploads/media/Open_Models_Feasibility_Study_SEPT_2008.pdf)
32. Kotz S, Johnson NE, Balakrishnan N (1994) Continuous univariate distributions. Wiley series in probability and statistics. Wiley, New York
33. Mendenhall W, Beaver RJ, Beaver BM (2008) Introduction to probability and statistics, 13th edn. Duxbury, Pacific Grove
34. Mendling J (2008) Metrics for process models: empirical foundations of verification, error prediction, and guidelines for correctness. Lecture notes in information system processing, vol 6. Springer, Heidelberg

35. Nielsen M, Thiagarajan PS (1984) Degrees of non-determinism and concurrency: a Petri net view, Foundations of software technology and theoretical computer science. Springer, Berlin/Heidelberg, pp 89–117
36. Nielsen M, Plotkin G, Winskel G (1981) Petri nets, event structures and domains, part I. *Theor Comput Sci* 13(1):85–108
37. Pearl J (2001) Causality—models, reasoning, and inference. Cambridge University Press, Cambridge
38. Rahm E, Do HH (2000) Data cleaning: problems and current approaches. *IEEE Data Eng Bull* 23(4):3–13
39. Scheuch R, Gansor T, Ziller C (2012) Master data management. TDWI Europe, dpunkt Verlag, Heidelberg (in German)
40. Schuette R, Rotthowe T (1998) The guidelines of modeling—an approach to Enhance the quality of information models. In: Ling TW, Ram S, Lee M-L (eds) ER'98: international conference on conceptual modeling. Lecture notes in computer science, vol 1507. Springer, Heidelberg, pp 240–254
41. Sedgewick R, Wayne K (2011) Algorithms, 4th edn. Addison Wesley (Pearson Education), Boston
42. Shahar Y, Musen MA (1993) RESUME: a temporal-abstraction system for patient monitoring. *Comput Biomed Res* 26(3):255–273
43. Spies M (2004) Einführung in die Logik. Spektrum Akademischer Verlag, Heidelberg Berlin (in German)
44. Svolba G (2012) Data quality for analytics using SAS. SAS Institute, Carry, NC, USA
45. van der Aalst WMP (1998) The application of Petri nets to workflow management. *J Circuits Syst Comput* 8(01):21–66
46. Vert JP, Tsuda K, Schölkopf B (2004) A primer on kernel methods. In: Kernel methods in computational biology. MIT, New York, pp 35–70
47. Wang HH, Noy N, Musen M, Redmond T, Rubin D, Tu S, Tudorache T, Drummond N, Horridge M, Seidenberg J (2006) Frames and OWL side by side. In: Int'l Protégé Conference, pp 54–57
48. Weiers RM (2005) Introduction to business statistics. Thomson Learning, Belmont, CA
49. <http://en.wikipedia.org/wiki/Phenomenon>. Accessed 17 March 2014
50. [http://en.wikipedia.org/wiki/Ontology\\_%28information\\_science%29](http://en.wikipedia.org/wiki/Ontology_%28information_science%29). Accessed 30 April 2014

# Chapter 3

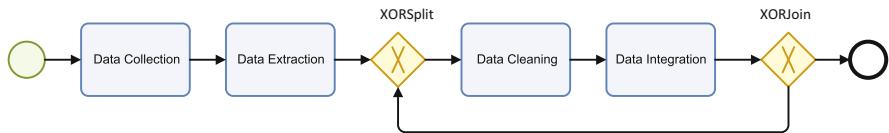
## Data Provisioning

**Abstract** This chapter elaborates on the data provisioning process ranging from data collection and extraction to a solid description of concepts and methods for transforming transactional data into analytical data formats. By the term transactional, data we also encompass data with a specific temporal structure, which will be later used in process analysis. Additional focus will be put on big data and data quality.

### 3.1 Introduction and Goals

Data provisioning constitutes the prerequisite for any Business Intelligence (BI) project. Clearly, without any data basis, there will be no analysis at all and without a database of good quality, the expected quality of the analysis can be expected to be low as well. However, data collection, extraction, and integration are often the most complex and expensive tasks in a BI project. As already stated by Inmon [38]: “What at first appears to be nothing more than the movement of data from one place to another quickly turns into a large and complex task far larger and more complex than the programmer thought.” According to [8], companies state that “information integration is thought to consume about 40 % of their budget.” Kimball states that the design and development of the underlying “systems consumes the lion’s share of effort during a DW/BI project” [40]. In addition, due to current developments, such as big data, more and more data is available holding potential for valuable analysis.

Possible sources for large data volumes are e-business and social network data [20]. On top of data volume, data variety and data velocity pose additional challenges [10]. Data velocity, for example, might demand for data extraction in very short time frames or even in a continuous way (online data extraction [4]). Data variety addresses the fact that data from different sources might be structured, semi-structured, or even unstructured while being available in different formats. Data volume, variety, and velocity are referred to as the three Vs in big data where additional Vs such as data veracity, i.e., the trustworthiness of the data, might also be an important issue (cf. [64]). Specifically, it has to be acknowledged that real-world data is dirty [37]. Therefore, data quality constitutes a crucial challenge as well.



**Fig. 3.1** Data provisioning process (in BPMN notation)

In summary, the goals of this chapter are to provide an understanding of how to collect and describe data (cf. Sect. 3.2), extract data from various sources (cf. Sect. 3.3), find the adequate target format for subsequent analysis (cf. Sect. 3.4), as well as to clean and integrate data (cf. Sect. 3.5) in such a way that the analysis goals can be achieved. The corresponding data provisioning process is depicted in Fig. 3.1. Note that data cleaning and integration might be done in an iterative way depending on whether the data quality has reached a sufficient level and if an adequate target format has been chosen for later analysis (cf. Sect. 2.5.3).

## 3.2 Data Collection and Description

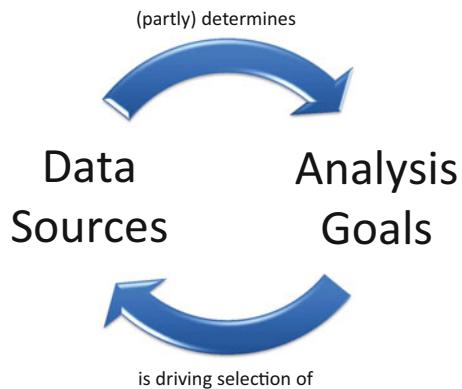
What is often underestimated is the effort of collecting the data for the BI project including the identification and selection of relevant data sources. As shown in Fig. 3.2, in some projects, the data sources might even become the driver for later analysis, i.e., the analysis goals might partly depend on which data sources are available. Clearly, it is more promising regarding the outcome of the BI project if the analysis goals drive data collection. In practice, an often-occurring realistic outcome is a compromise between the desired analysis goals and the available data [24].

In the following, we report on our experiences in collecting data stemming from the two use cases on patient treatment (cf. Sect. 1.4.1) and higher education (cf. Sect. 1.4.2).

### EBMC<sup>2</sup> Use Case: Data Considerations

The main goal of the EBMC<sup>2</sup> project (see Sect. 1.4.1) is the process-oriented analysis of patient treatment data [24]. Hence, we started by identifying data sources regarding their potential relevance for the analysis goals and their availability. An additional side demand was to find sources that offer a reasonable amount of data in order to make the results of the analysis more meaningful. Another question in the project was whether the connection between skin cancer treatment processes and other treatment processes can yield more insights than an isolated analysis. Hence, we were looking for different data sources that can be connected to skin cancer treatment processes. For all data sources we had to take into consideration data ownership and privacy issues. In the end, data collection resulted in an iterative process where we started with a subset of the initially considered data sources and successively added further data sources

**Fig. 3.2** Connection between data sources and analysis goals



due to content-related but also practical reasons such as costs for data access. Details on the EBMC<sup>2</sup> use case can be found on the homepage of the book: [www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

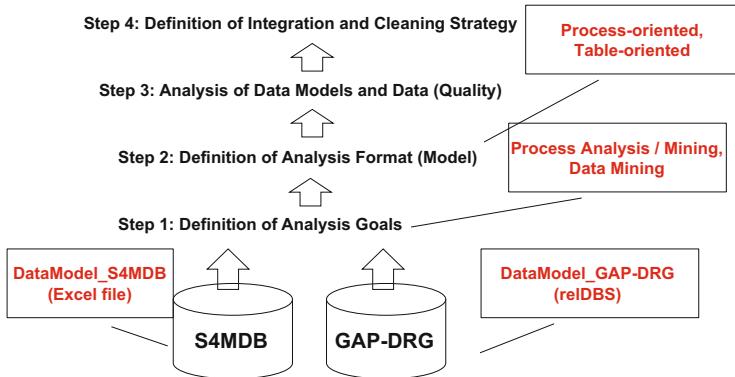
#### HEP Use Case: Data Anonymization

In the HEP project (see Sect. 1.4.1), a single data source was available that contained all data on the different education processes. Hence, no explicit selection of data sources was necessary. We anonymized the data in order to not interfere with data protection policy [15]. Details on the EBMC<sup>2</sup> project can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

An important task of the data collection phase is the description of the data sources (e.g., data format, documentation of anonymization, analysis goals). In general, procuring meta-information in any phase of the business intelligence process is very helpful for various reasons, such as maintenance, support of new users, and further adaptations [38].

Figure 3.3 illustrates the description of the EBMC<sup>2</sup> data sources in a schematic way. The data to be analyzed stems from two sources, i.e., the Stage IV Medical Database (S4MDB) that contains the treatment data of skin cancer patients of stage IV and the GAP-DRG database that stores medical billing data in Austria. The S4MDB database was available in Excel format, whereas GAP-DRG is a relational database. Since they were designed for different purposes, the sources are differing in their conceptual schema possibly leading to integration challenges. As the analysis goals primarily focus on patient treatment, the chosen target format should enable process-oriented analysis. Additional questions, such as survival analysis, can be tackled by data mining techniques demanding for associated target formats.



**Fig. 3.3** Data source description for the EBMC<sup>2</sup> project

### 3.3 Data Extraction

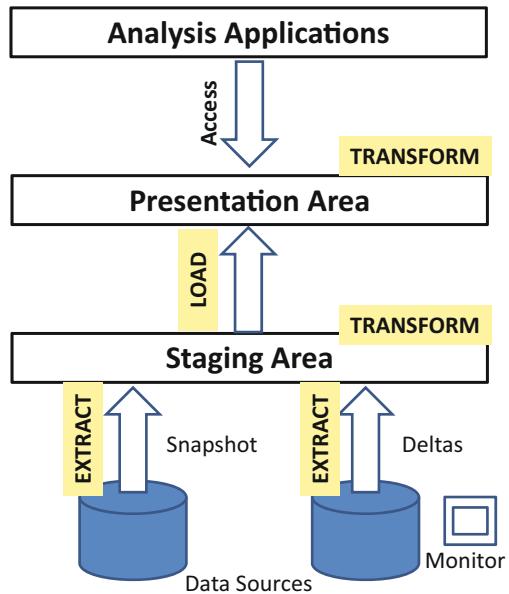
After selecting the relevant data sources and describing them including analysis questions and formats, the next step is to extract relevant data from their sources. Typically, data extraction is part of the so-called extraction-transformation-load (ETL) process, which will be described in Sect. 3.3.1. We will also take a look at challenges concerning extraction in the context of big data in Sect. 3.3.2.

#### 3.3.1 *Extraction-Transformation-Load (ETL) Process*

In this section, the necessary steps are introduced for transforming and integrating data from various possibly heterogeneous data sources for analysis purposes. The overall procedure is referred to as the ETL process [62]. Figure 3.4 depicts a schematic overview. At first, data stored within different source systems, such as databases, legacy systems, or XML documents, is extracted into the so-called *staging area* that provides different services for transforming and cleaning data [41]. The term *staging area* stems from the data warehouse (DWH) area, and constitutes one part of the DWH reference architecture. From the staging area, the cleaned and integrated data can be loaded into a presentation area where users can perform analyses. In the following, we describe the steps of extraction and loading in a more detailed manner. Transformation as a means to clean and integrate data will be implicitly addressed in Sect. 3.5.

Data extraction is concerned with different questions. In data extraction one should first of all think about the question *when* data is extracted. In some settings, it might be sufficient to extract data from the sources only when resulting in a partial or complete *snapshot* of the source data (cf. Fig. 3.4). Often, particularly if data is extracted from operational systems, this is not sufficient as the data might quickly

**Fig. 3.4** ETL process  
(adapted from [41])



become outdated. In order to stay informed about updates within the sources, typically, the sources are monitored for updates. Extraction of updates might be realized, for example, by pull or push strategies between data source and staging area.

Connected with the question *when* to extract is the question *what* to extract. Intuitively, it can be overly complex to extract all data within a snapshot each time the data source undergoes an update. Instead, it is often desired to only extract the *delta* compared to the last data snapshot within the staging area. A common example would be the list of participants from a conference registration system that was extracted at time  $T_1$  as complete snapshot:

$$T_1 = \langle \text{Smith, Brown, Mayer, Jaeger} \rangle$$

Assume that there is an update at time  $T_2$  and the new snapshot would be

$$T_2 = \langle \text{Smith, Brown, Jaeger, Jones} \rangle$$

Then it would be favorable to only transfer the new participants, i.e., Jones, and the participants who are not attending the conference any longer, i.e., Mayer instead of the complete participant list at  $T_2$ .

Connected with the question of *what*, i.e., snapshot versus delta, is the question of *how* to extract. If the data source is a database system, for example, it typically offers many ways for extracting both, snapshots and deltas. Snapshots can be extracted by SQL queries. Deltas can be extracted by utilizing database logs. However, not all sources offer such convenient support. As legacy systems are still present in enterprises, and often do not offer any support for data extraction, it is, in some cases, only possible to take snapshots of the data followed by calculating differential snapshots between the last and the current version of the data. The

efficient calculation of such snapshots has been tackled by different approaches, e.g., the window algorithm [29].

After extracting data updates from the sources, the data is to be transferred to the staging area of the DWH for data cleaning and integration purposes. For transferring massive data sets, *load* techniques such as bulk loader [7] are offered, for example, the Oracle SQL\*Loader. When loading large data sets in a one-batch fashion, COMMIT and related integrity checks are postponed after the entire loading procedure. On the contrary, when loading (small) amounts of data in a continuous manner, COMMIT and related checks are performed after each loaded portion [52]. In the first case, loading a large set of data might be much faster, but should be only done for cleaned and trusted data. Thus, loading is typically applied after cleaning and integrating the extracted data within the staging area.

In the following, data extraction from two different sources is illustrated by means of the EBMC<sup>2</sup> case study including tool support.

#### EBMC<sup>2</sup> Use Case: Data Extraction

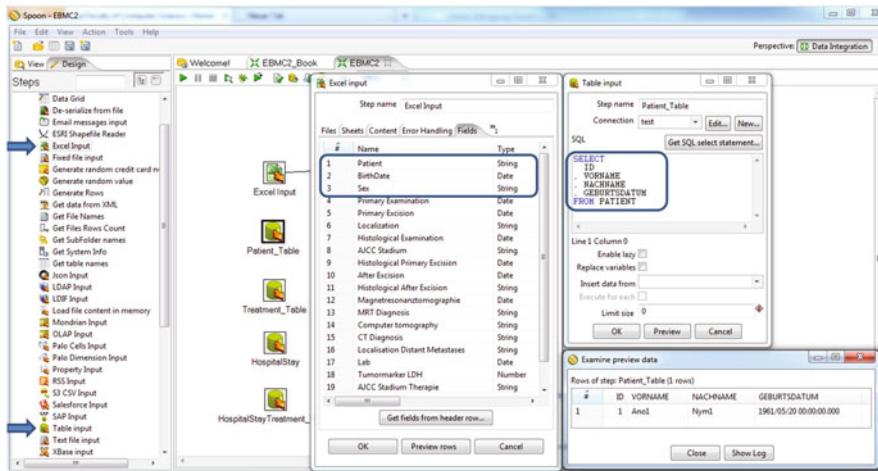
The data extraction process is illustrated by means of (simplified) data structures from skin cancer patient treatment as described in [24] and by using the Pentaho open-source data integration tool Kettle.

Figure 3.5 illustrates the connection and extraction of data from the two sources S4MDB and GAP DRG as Excel and table input, respectively. Pentaho Kettle offers the possibility to describe the entire ETL process by means of a graphical workflow. Figure 3.5 shows the input activities Excel Input, Patient\_Table, Treatment\_Table, HospitalStay, and HospitalStayTreatment. The pane to the left contains the available connectors and adapters for a variety of data source types. For all connected sources, a retrieval of the included data is possible. For table Patient\_Table, for example, the SQL statement as depicted in Fig. 3.5 yields the result tuple (1, Ano, Nym, 1961/05/20 00:00:00.000). Note that at this point, the entire data is extracted from the sources. Based on this, it can be specified which data should be transferred to subsequent transformation and cleaning steps. Details on the EBMC<sup>2</sup> project can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

A selection of data extraction tools:

- Pentaho Spoon (open source): <http://community.pentaho.com/>
- Talend Open Studio (open source): <http://www.talend.com/products/big-data>
- Oracle Data Integrator: <http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>
- SAS Data Integration Server: <http://www.sas.com/resources/factsheet/sas-data-integration-server-factsheet.pdf>
- SAP Business Objects Data Integrator: <http://www54.sap.com/pc/tech/enterprise-information-management/software/data-integrator/index.html>



**Fig. 3.5** Health-care data extraction (using Pentaho Spoon)

Overall, these tools offer a variety of data connectors in order to connect and access different data sources as for the example connectors above to Excel or relational data sources. Moreover, most of the tools enable the definition of processes/workflows that describe the data extraction and integration tasks. Further, it is possible to define data transformations and mappings within the tools. Both aspects contribute to the documentation of the integration process, and, consequently they facilitate maintenance and adaptations. However, selection, extraction, and integration of data remains a manual task no matter if an integration tool is used or not.

### 3.3.2 Big Data

The extensive use of social media (e.g., half a billion tweets a day on Twitter [59], more than one billion active users on Facebook [25]), sensor applications (e.g., for measuring health parameters or environmental conditions), as well as the immediate provision of possibly large result data by modern search engines has led to a massive increase in produced and potentially interesting-to-analyze data. Kearny amounts worldwide spending for *big data* to \$114 billion in [39].

According to [10], key challenges in the context of handling big data are *data volume*, *data velocity*, *data variety*, and *data veracity*. In short, data volume refers to processing huge amounts of data, data velocity to the frequency with which new data enters the integration and analysis process, data variety to the diversity of data, and data veracity to the trustworthiness of the data. In this section, we analyze which challenges are posed on data extraction and integration by these four “Vs.”

Particularly interesting is whether these “Vs” lead to new challenges that cannot anymore be solved by traditional approaches for data extraction and integration.

*Data volume:* Currently, NoSQL databases are on the rise for tackling the challenge of data volume. Such databases can be roughly categorized into *key value stores* [58], *graph databases*, as well as *XML databases* and are expected to dissolve the potential restrictions imposed by relational databases such as the demand for a schema, ACID<sup>1</sup> transactions, and consistency. In turn, NoSQL databases claim to be schema-free and eventually consistent (“data fetched are not guaranteed to be up to date, but updates are guaranteed to be propagated to all nodes eventually” [17]). Further, they conduct BASE (i.e., basically available, soft state, eventually consistent [17]) instead of ACID transactions are open source, distributable [47], and hence ready for big data. Stonebraker [58] puts it that what is usually expected from using NoSQL databases is flexibility (not being bound by a schema) and performance.

According to [2], *key-value storage* systems are adopted by various enterprises. The basic data model consists of key-value pairs and an additional payload [58] and can be implemented in various ways. The following statement

```
store[:Key1] = "Some string"
```

would create an object with key Key1 and value "Some string" (based on the ruby example presented in [53]). Instead of a simple string, more complex objects reflecting, for example, an object hierarchy can be assigned as value to a key [53]. The applied data analysis paradigm is MapReduce [23]. *MapReduce*, basically, follows a divide and conquer approach by first fragmenting the data set into smaller portions (*Map*). Then, the calculations are performed on the data portions and finally merged into the final result (*Reduce*). An open-source implementation is Hadoop [66] with a widespread adoption in industry and academia.

In *graph databases*, the data are represented as graph structure where queries navigate on. In the following, we will illustrate the characteristics of a graph database based on the example of sones GraphDB<sup>2</sup> which combines object-oriented aspects and graph database aspects. The basic structure is a graph  $G := (V, E)$  where  $V$  is a set of vertices and  $E$  is a set of edges. Queries are defined on the basis of graphical query language (GQL), for example:

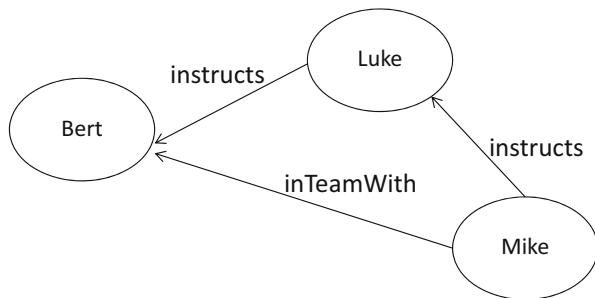
```
CREATE VERTEX TYPE Person
ATTRIBUTES (SET<Person> Trainees, SET<Person> Colleagues, String name)
INCOMINGEDGES (Person.Trainees instructs, Person.Colleagues inTeamWith)
INSERT INTO Person Values (name='Bert')
INSERT INTO Person VALUES (name = "Luke", Trainees = SETOF(name = "Bert"))
INSERT INTO Person VALUES (name = "Mike", Trainees = SETOF(name = "Luke"),
Colleagues = SETOF(name = "Bert"))
```

---

<sup>1</sup>ACID is an acronym that describes the basic properties of database transactions, i.e., atomicity, consistency, isolation, and durability [35].

<sup>2</sup><https://github.com/sones/sones>.

**Fig. 3.6** Example graph data structure



This GQL code creates the graph structure depicted below using tools such as sones database (cf. Fig. 3.6). The graph structure fosters the quick processing of certain queries that navigate over edge types, for example, finding all friends of Mike.

Another example for a graph database is OrientDB. Please find a discussion in Appendix A.

Data velocity refers to the frequency of updates from the data sources. Whereas for data warehouses some years ago, updates in the data sources could be treated in a periodic way; nowadays, continuous data updates have become a frequent scenario, e.g., in the case of sensor data (also referred to as *streaming data*). As argued in [4], for many enterprises it is crucial to analyze online data from different Web sources such as reports on the current market situation or price indexes. This necessitates the “just-in-time” extraction of data from Web sites. In [4], for example, the method of Web data extraction is conducted by a Web connector that “comprises means to extract data from Web applications” [4]. Furthermore, it enables data transformation, cleaning, and subsequent loading into a data warehouse.

As an open source tool for the extraction and provision of streaming data, H2O is discussed in Appendix A.

Gaber et al. [28] provide a survey on techniques for data management and analysis of streaming data. The survey distinguishes between data-based techniques from task-based techniques. They comprise:

#### Handling of Streaming Data (Based on [28])

- *Sampling*: An item of the data stream is randomly selected to be analyzed. Critical is that the size of the overall data set is unknown and items of interest might be not considered.
- *Load shedding*: Sequences of the data streams are randomly dropped. Same critical issues as for sampling.
- *Sketching*: The items are selected based on a random projection of features. The critical issue here is accuracy.
- *Sliding window*: Assigns a higher interest to the data of higher actuality (see also the approach of [36] discussed in the following).

For analyzing streaming data, *sequential online analytical processing (OLAP)* [36] and clustering approaches [48] have been proposed. OLAP usually operates on multidimensional data structures and will be discussed in Sect. 3.4.1. Sequential OLAP as presented in [36] covers continuous data update by a so-called *tilted time frame*. This means that due to time and space complexity, the data is processed into cubes depending on the distance to the last update, i.e., the most recent updates are kept at the lowest granularity in the cube whereas updates with a larger distance to the current point in time are stored at a higher granularity. In addition, the other dimensions are not fully materialized, but kept as so-called *critical layers*.

*Data variety* reflects the increasing number of different data formats that might have to be integrated or, put in a more specific way, structured, semistructured, and unstructured data. The integration becomes particularly difficult if schema information is missing. As stated in [9], “approximately half of the XML documents available on the Web do not refer to a schema.” In literature, different techniques were introduced that derive schema information from a set of XML documents in a tree-based [9] or text-based [30] manner. Tree-based approaches, for example, take the structure tree of each XML document and aggregate them according to certain rules. The underlying schema can be derived based on the aggregated tree. Several XML tools and editors offer to automatically derive the underlying XML schema from a set of XML documents (e.g., Liquid Studio<sup>3</sup>).

Another possible solution to address a mix of structured and unstructured data, at least in the context of XML data, is to use an XML database such as BaseX.<sup>4</sup> BaseX is able to store structured, semistructured, and unstructured data, and hence it actually addresses the two challenges of data volume and variety. The basic object to be stored is an XML document or a collection of XML documents. Figure 3.7 shows an example of an XML document collection for literature management as stored and managed in BaseX. In particular, it can be seen that the XML documents contain structured as well as semistructured data. The data can be queried using XQuery. Contributions to high data volume have been made by developing internal data structures that are optimized towards efficient processing of navigating queries (such as for XQuery) [33].

*Data veracity* connects big data with the question where the data comes from, e.g., data that is stored in a cloud. In such settings, we have to think about how we can ensure trust in the data we collect and want to analyze. In this context, techniques for auditing data by, for example, a third-party auditor have been proposed in literature (see, e.g., [65]).

A survey and comparison of existing tools for big data can be found in Appendix A.

## Discussion on New Challenges in the Context of Big Data

Data variety is an old and still existing problem, particularly, if integration necessitates knowledge on the semantics of data structures. Big data volume has led to the

---

<sup>3</sup><http://www.liquid-technologies.com/xml-studio.aspx>.

<sup>4</sup><http://basex.org/>.

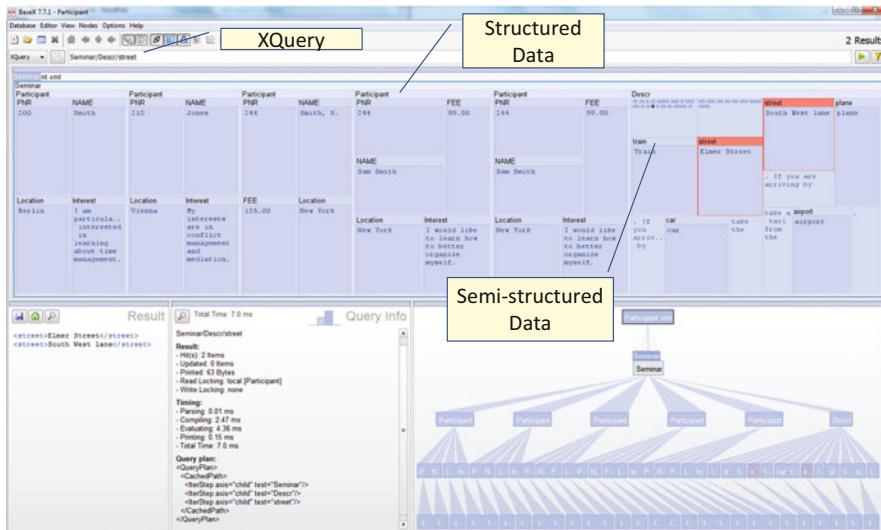


Fig. 3.7 Example literature XML database (using BaseX)

increased use of NoSQL databases, such as graph databases, key-value stores, and document databases. Another development were extensions of relational databases and BigTable solutions, e.g., Google BigTable. The basic data structure of BigTable consists of `rows`, `columns`, and `timestamps` that are mapped onto a `String` and stored as multidimensional sorted map [18]. How the BigTable solution is applied in practice is described in [18] by means of several Google applications such as Google Analytics (data volume approximately 200 TB) or Google Earth (data volume approximately 70 TB). Huge data volumes also require techniques to quickly move data from one place to the other (cf. bulk loading). Finally, if moving data is expensive, it might also become necessary to conduct analyses close to the data [32].

Facing big data, the new challenges for the ETL process are mainly caused by data velocity, i.e., just-in-time data extraction becomes necessary. However, novel approaches for data integration are particularly interesting in this context. Shifting from ETL to *ELT* (extraction-load-transformation), for example, has been suggested for big data projects. The idea is that data storage frameworks such as Hadoop work on the “raw data before making it available to other systems for further analysis” [26]. Essentially, the transformation and cleaning steps are shifted after first analyses have been applied. In this way, analyses can be conducted more rapidly, shifting expensive transformation and cleaning tasks to later phases of the project. Additionally, this might help to address the data velocity challenge as, potentially, data updates can be fed into the analyses more quickly.

Finally, there are strong indications that analysts with a particular expertise on processing and analyzing big data are missing at a large scale. The problem is not

that the mountains of data are missing, but people who know how to dig out the diamonds.

### 3.3.3 Summary on Data Extraction

Data extraction is more than just grabbing data from a source. Their major challenges and their main countermeasures are as follows:

- Data availability, ownership: In order to obtain data from a certain domain, it is often indispensable to develop some understanding for the domain. Soft skills help to communicate with the domain experts in order to overcome possible resistances. (Keep in mind that analyzing data might also lead to unexpected or even undesired results.) Further, legal knowledge can be of advantage when it comes to data privacy questions.
- Heterogeneous data sources: Many tools exist that offer a bunch of adapters and extractors to facilitate data extraction. However, the basic design of the data extraction process remains a manual task.
- Big data: The challenge of volume is not the difficult one, and hence, the term big data might be a bit misleading. More crucial are variety, update frequency, and trustworthiness of the data. However, it is most crucial to define what to analyze in a huge bulk of data, i.e., asking the right questions.

## 3.4 From Transactional Data Towards Analytical Data

Section 3.3 presented how to extract data from possibly heterogeneous sources for later analysis purposes. As it is not constructive to analyze each portion of extracted data separately, the extracted data is to be cleaned and integrated. However, before data integration can take place, we have to decide in which format the data should be integrated (*integration format*). In some cases, the integration format is already the format the later analyses will work with. In other scenarios, it becomes necessary to provide additional *analytical formats* that are based on the integration format. One example is that different analyses will be conducted that require different analysis formats. In other words, the choice of the analytical format depends on the analysis questions and the key performance indicators. The choice of the integration format depends on the results of the data extraction step including the considerations on the later analytical format. Finally, both choices should be made bearing in mind data quality issues (cf. Sect. 3.5).

Figure 3.8 depicts a selection of basic integration and analysis formats that can be transformed into different analytical models (cf. Sect. 2). The columns are labeled with formats including a top-level distinction between structured and unstructured formats. Structured formats can be further distinguished into flat



Structured Data Formats				Unstructured Data
	Flat (e.g., relational, CSV)	Hierarchical (e.g., XML)	Hybrid (e.g., XES)	Text
Table Formats	Flat	Contains / generates (mapping)	Generates (mapping)	Contains
	Multidimensional	Generates (mapping & aggregation)		Generates (mapping & aggregation)
Log		Generates (mapping & transformation)	Generates (mapping & transformation)	Contains or generates (transformation)

Mining and generation

**Fig. 3.8** Selection of integration and analysis formats

formats, such as relational tables hierarchical formats, such as XML and hybrid formats, such as XES. The rows are labeled with formats featuring a distinction into table and log formats. Table formats can be further distinguished into flat and multidimensional formats.

For unstructured data, we mainly refer to text data but also to unstructured XML documents. For the analysis of unstructured data, *text mining* will be introduced in Chap. 8. Text mining provides the means to analyze text data. It could be also used to transform text data into structured data (see, e.g., [1]).

Structured data formats can be divided into *flat*, *hierarchical*, and *hybrid* structures. Typical flat formats comprise relational tables, comma-separated values (CSV), or Excel files (as it is the case in our health-care example). A prominent hierarchically structured format is XML since the structure for XML documents can be mapped onto a tree structure. XES (eXtensible Event Stream) [63] is also an XML-based structure but forms an extra column since it additionally aims at comprising flat as well as log-structured data. The different analysis (and possibly integration) formats denoted by the row labels range from flat *table* structures over *multidimensional* structures to process-oriented *log* structures. All these formats are explained in the following sections.

The cells describe how integration and analysis formats can be transferred to other integration and analysis formats (read from top to bottom as indicated by the arrow).

Given two data sources A and B, the following transformations are conceivable:

- A *contains* B: The format of A equals the format of B, and B is a subset of A
- A *generates* B:

- By *aggregation*: A flat or multidimensional, B multidimensional, aggregation function, e.g., SUM, AVG; aggregation refers to defining different abstraction levels within the schema and aggregating the data along these levels (typically applied for multidimensional data formats, cf. Sect. 3.4.1).
- By *mapping*: dimensionality might be changed; describes a set of attribute correspondences between two schemata based on which one schema can be mapped onto the other [50]. We will further elaborate on this issue in Sect. 3.5.
- By *transformation*: A and B of any format; transformation changes a schema (format) in order to obtain a desired target schema (format).

### 3.4.1 Table Formats and Online Analytical Processing (OLAP)

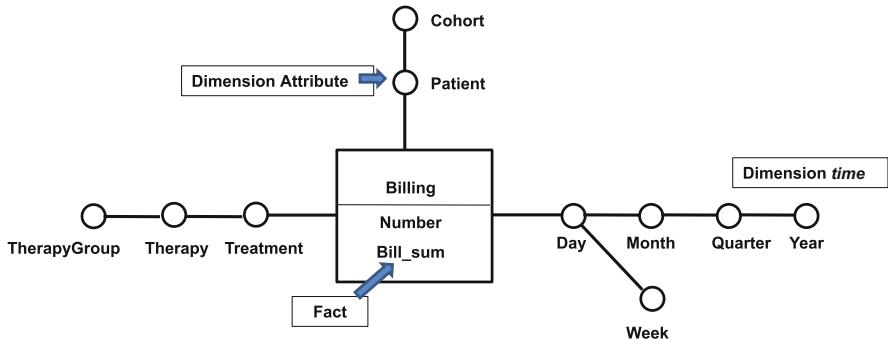
Table formats refer to multidimensional table structures as typically employed in DWH systems. As a common metaphor, the data cube is used, but the data structures do not have to be necessarily structured in a three-dimensional manner. Following the dimensional modeling approach presented by Kimball [40], multidimensional structuring basically is “dividing the world into measurements and context.” Typical measurements in the business context are turnover or revenue; in a meteorologic context, measurements could be temperature or precipitation. In order to process and analyze the measurements, they mostly turn out to be numeric. Such numeric measurements are called *facts* [40].

Instead of only looking at the sheer facts, it is often more interesting to analyze them in a different context. This context is realized by different properties or *dimensions* that describe the facts. One prominent example is *time* describing, for example, the turnover or the temperature measured at a certain day. Referring to this example, it might not only be interesting to look at facts on a daily basis. We could also be interested into analyzing them at a more fine-grained level, e.g., on a minute basis, or at a more coarse-grained basis, such as based on quarters.

These different *granularity levels* within the dimensions are reflected within the multidimensional model. This enables operations, such as *roll up* or *drill down*, that navigate through the granularity levels of the dimensions by applying aggregation functions. To roll up, for example, the turnover per day to the turnover per month, applying the sum function on the daily facts for all days of the month is a conceivable aggregation function. However, this would not make much sense for certain facts such as temperature. Here, aggregation functions, such as min, max, or average, are conceivable.

#### *Example 3.1 (Health Care)*

Figure 3.9 depicts an excerpt of medical data structured in a multidimensional way. Interesting measurements corresponding to key performance indicators, such as cost-effectiveness of treatment or survival time (cf. Chap. 1), are reflected by the facts number of patients or billing sum. As for many applications, a time-related view on the facts is interesting for this example as well. This results in



**Fig. 3.9** Multidimensional modeling of medical data (adopting conceptual modeling notion proposed by Golfarelli et al. [31])

Billing		MRT	XRay	SkinCheck	SUM
2001	Cohort1	100	20	120	240
	Cohort2	50	30	40	120
	SUM	150	50	160	360
2002	Cohort1	110	30	100	240
	Cohort2	40	40	40	120
	SUM	150	70	140	360
2003	Cohort1	100	100	30	230
	Cohort2	10	10	40	60
	SUM	110	110	70	290
SUM		410	230	370	1010

**Fig. 3.10** Report with aggregated facts from medical domain (fictive data)

the dimension `time`, which enables the aggregation along the dimension attributes `day`, `month`, `quarter`, and `year`.

Figure 3.10 shows an example report based on the application of aggregation functions to the multidimensional structure presented in Fig. 3.9. In this report, the billing sums per patient cohorts, therapy, and year have been aggregated. Aggregation is one ingredient of online analytical processing (OLAP) operations that are typically used to analyze multidimensional data (more examples for OLAP operations are provided in the sequel).

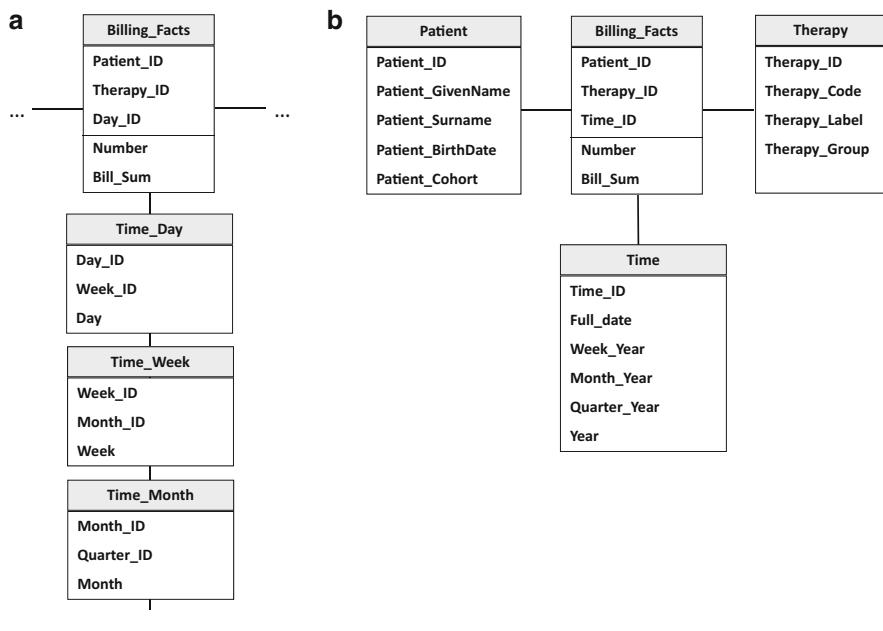
How shall multidimensional structures be implemented? Basically, three options exist, i.e., multidimensional OLAP (MOLAP), relational OLAP (ROLAP), and hybrid OLAP (HOLAP) [19]. MOLAP applies native, multidimensional structures. ROLAP maps multidimensional structures onto relational tables. HOLAP refers to the combined application of relational and multidimensional structures.

All approaches show advantages and limitations. By using ROLAP, on the one hand, the full database functionality can be exploited. On the other hand, mapping

multidimensional and partly hierarchical structures onto flat table structures can become cumbersome. Hence, certain schemata have been developed to support the mapping of multidimensional to relational structures. Prominent representatives for ROLAP data structures are the *snowflake* and the *star schema* [44]. Both schemata have in common that they organize facts and dimensional data within different tables, i.e., there is a separate fact table and tables that represent the dimensions. The difference lies in the way how the snowflake and star schema organize the dimensional tables.

In the snowflake schema, the fact table references all dimension tables of lowest granularity, and for each further classification level, another dimension table is kept that is referenced by the table of the next lowest granularity. For the medical example as illustrated in Fig. 3.11a, the fact table `Billing_Facts` references table `Time_Day`, table `Time_Day` references table `Time_Week`, and so forth. Although this structure is normalized, long join chains may become necessary when applying OLAP operations, e.g., aggregating facts up to the year level as for the example depicted in Fig. 3.10.

The star schema also stores a fact table, but only one table per dimension that holds all information on this dimension. Figure 3.11b depicts the example of Figs. 3.9 and 3.10 realized as a star schema. The fact table `Billing_Facts` references all three dimension tables `Patient`, `Time`, and `Therapy`. The dimension



**Fig. 3.11** Medical data modeled as snowflake schema (fragment, a) and as star schema (b)

tables contain the information of all dimensions. The decreased complexity of the schema comes at the price of de-normalization.

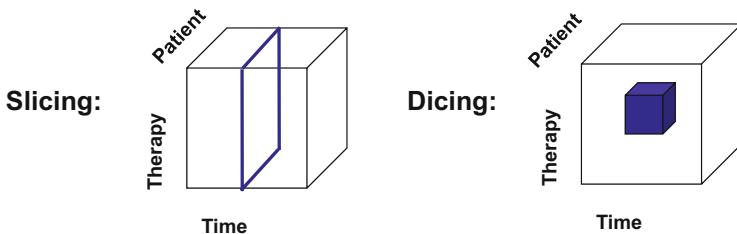
The basis for the fact table is always the data at the lowest granularity level. An example entry in the `Billing_Facts` table could be (`p177, th244, t855, 1, 20.5`), an example entry in the `Time` table (`t855, 2012-03-02, 9_2013, 3_2012, 2012`), referring to the star schema, Fig. 3.11. This means that for patient `p177`, a billing sum of `20.5` occurred on March 2, 2012. Along the time dimension, the associated entries are the 9th week and the third month of year 2012. Imagine another fact entry (`p178, th200, t856, 1, 30`) with associated time entry (`t856, 2012-03-04, 9_2013, 3_2012, 2012`). Further on, assume that these two entries are the only ones for March 2012. Then, aggregation along the time dimension from day to month by using the summation of the billing sum would result in `50.5` for March 2012.

As discussed in [44], the star schema provides, obviously, a more compact representation of the multidimensional data. This comes at the cost of de-normalization. The snowflake schema constitutes a normalization of the star schema. However, the modeling and maintenance of several dimensional tables might negatively influence query performance for, for example, highly dimensional data with multiple granularities per dimension.

How can multidimensional data structures be analyzed? At first, often, the data is analyzed in an explorative way for a better understanding. For this cause, different OLAP operations can be applied. Thus, we can look at the multidimensional data from different perspectives. Metaphorically, this is achieved by rotating the corresponding cubes.

Rotation constitutes a mostly visual inspection without further manipulation of the data. Hence, it is especially suited for understanding and discussing the data and possible hypotheses formulated on the basis of the data. In addition to visual inspection, a set of OLAP techniques exists that change the granularity and/or dimensionality of the data cube in order to gain additional insights. Increasing or decreasing the granularity of the data is achieved by moving along the dimensional classification, e.g., aggregating the data from granularity level day → month → quarter → year on dimension `Time` in Fig. 3.13. For doing so, we can use the operation *Roll up* that generates new information by the aggregation of data along the dimension where the number of dimensions is not changed. The complementary effect, i.e., the navigation from aggregated data to detailed data along the classification hierarchy, is achieved by the operation *Drill down*. By applying the *Drill across* operation, we can calculate facts in another classification hierarchy or dimension (i.e., from one cube to another). One example would be to roll up the dimension `Time` from `Quarters` to `Months` or vice versa.

Operations that do not change the granularity of the data, but the dimensionality or the size of the cube, respectively, are *Slice (OLAP)* and *Dice*. *Slice* generates individual views by cutting “slices” from the cube based on point and list restrictions on classification attributes. In general, this operation reduces the number of dimensions. As an example, one could be interested in the average duration of all



**Fig. 3.12** Two OLAP operations: slice and dice

process instances for a certain patient X in the current year. To answer this question, a slice of the data would be generated by the following query:

```
SELECT ... WHERE year = '2006' AND patient = 'X'.
```

Dicing means that the dimensionality of the cube is not reduced, but the cube itself is reduced by cutting out a partial cube. An example question answered by dicing could regard the number of patient treatments within a certain time frame, and the solution can be accomplished by range queries. Figure 3.12 depicts a slice operation (left-hand side) as well as dicing the data cube (right-hand side).

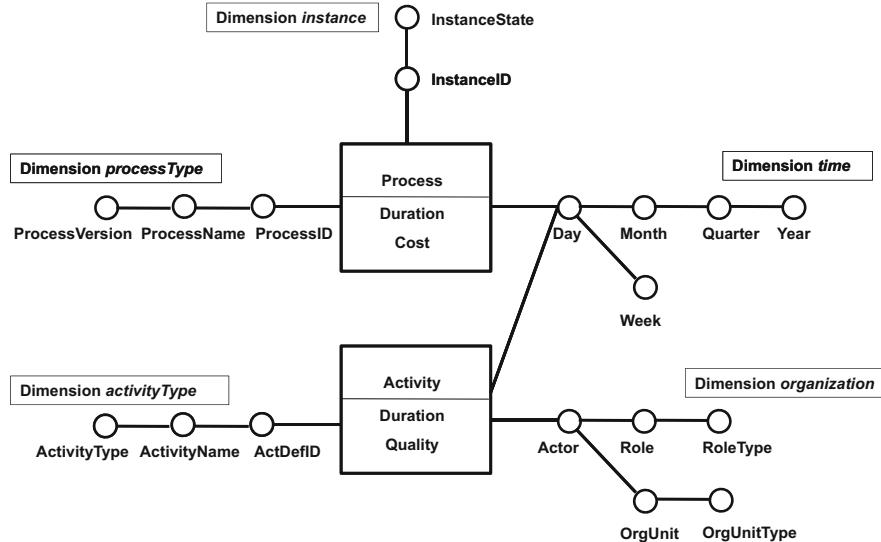
The representation and analysis of process execution data as multidimensional structures have been proposed by process warehouse approaches [5, 13]. We will illustrate this by means of an example from patient treatment:

#### *Example 3.2 (Patient Treatment)*

Figure 3.13 depicts the process warehouse data structure combining fact tables on the processes with the process activities which can both be analyzed along the dimension time. Activity-related facts are further specified by the dimension organization and activityType. Key performance indicators that could be determined on the basis of this structure could be the average duration of all executions of process activity approve loan (ActivityName) in branch X (OrgUnit) in 2012 (Year).

### 3.4.2 Log Formats

Process warehouse approaches offer techniques to analyze process-oriented data as multidimensional data structures, answering questions such as *what was the throughput time of all patient treatment processes in 2013?* The specific characteristics of process executions as temporal sequences of activity executions cannot be fully exploited in this way. An example question relating to the temporal ordering of activities would be *was the diagnosis always followed by an excision?* We will discuss the specifics of temporal data in Chap. 6. As a consequence, data formats are required to store temporal information on process executions in an explicit way. This is achieved through *log formats*.

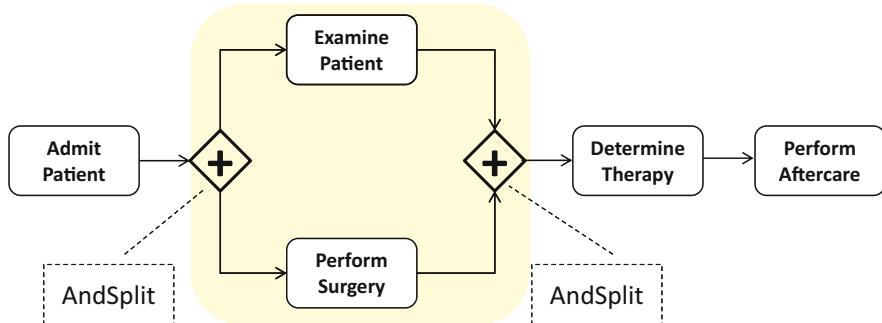


**Fig. 3.13** Multidimensional process warehouse data structure as suggested in [5, 13] (notation based on [31])

In general, a log can be defined as a collection of events recorded during runtime of an information system. Logging has a long tradition, for example, in the area of database recovery [35]. Lately, event logs have become prominent as a basis for process analysis and mining [60]. They store process data in an event-based way, i.e., for each observed execution of a process activity, at least one event is written to the corresponding process log. Typically, process logs are recorded by *Process-Aware Information Systems* (PAIS), such as workflow or ERP (enterprise resource planning) systems, possibly distributed via several PAIS.

For process-oriented analysis, it is crucial that the log contains information on the event order. Either it is based on time stamps connected with the events or the assumption holds that the order of the events within the log reflects the order in which they occurred during process execution. Further, the events of different process executions must be distinguishable, e.g., an event for executing activity `PerformSurgery` for patient X (executed within process instance X) must be distinguishable from the event for executing activity `PerformSurgery` for patient Y (executed within process instance Y). This requires some sort of instance ID within each event.

Currently, there are two process-oriented log formats that are predominantly used for process analysis, i.e., Mining XML (MXML) [34] and eXtensible Event Stream (XES) [63]. For illustration, see the example process depicted in Fig. 3.14, particularly the process fragment highlighted in yellow. It expresses a parallel execution of process activities `PerformSurgery` and `ExaminePatient` at



**Fig. 3.14** Process example (BPMN notation)—parallel branching highlighted in *light yellow*

runtime. Consequently, two possible execution logs can be produced by this process schema:

```

σ₁ = <= AdmitPatient, PerformSurgery, ExaminePatient,
      DetermineTherapy, PerformAftercare >
σ₂ = <= AdmitPatient, ExaminePatient, PerformSurgery,
      DetermineTherapy, PerformAftercare>

```

Figure 3.15a displays the corresponding log entries defined in the MXML format. The basic tag is `<AuditTrailEntry>` reflecting any event recorded within the log. The first audit trail entry, for example, refers to the start event produced by the process activity `PerformSurgery` (tag `WorkflowModelElement`). It took place at the given time, and there is no originator, i.e., the actor who has performed the corresponding task has been logged. Different tags can be analyzed in different ways. Workflow model elements and time stamps are mainly exploited for process mining techniques, for key performance indicators, and business intelligence questions such as process discovery or process performance. Information kept within the `Originator` tag can be exploited for organizational and social network analysis, for example, finding out about the organizational structures behind the processes. We will discuss these techniques in detail in Chap. 8.

Although the process mining manifesto [61] claims that “Event Data Should Be Treated as First-Class Citizens,” i.e., companies should provide the means to (semi)automatically record process-oriented log data within their information systems, reality still looks different in many settings. In the beginning of the EBMC<sup>2</sup> project, for example, there was no log data available, but instead relational tables and excel files. As process-oriented analysis techniques become increasingly important for business intelligence projects, the question is how to deal with such non-log data (for a classification of data along with its suitability for process-oriented analysis, see the L\* model in [61]).

<b>a</b> MXML	<b>b</b> XES
<pre>&lt;AuditTrailEntry&gt;   &lt;WorkflowModelElement&gt;5 PerformSurgery&lt;/WorkflowModelElement&gt;   &lt;EventType&gt;start&lt;/EventType&gt;   &lt;Timestamp&gt;2012-10- 02T13:56:36.075+01:00&lt;/Timestamp&gt;   &lt;Originator&gt;unknown&lt;/Originator&gt; &lt;/AuditTrailEntry&gt;</pre>	<pre>&lt;event&gt; &lt;string key="org:resource" value="unknown"/&gt; &lt;date key="time:timestamp" value="2012-10- 02T14:56:36.075+02:00"/&gt; &lt;string key="concept:name" value="5 PerformSurgery"/&gt; &lt;string key="lifecycle:transition" value="start"/&gt; &lt;/event&gt;</pre>
<pre>&lt;AuditTrailEntry&gt;   &lt;WorkflowModelElement&gt;5 PerformSurgery&lt;/WorkflowModelElement&gt;   &lt;EventType&gt;complete&lt;/EventType&gt;   &lt;Timestamp&gt;2012-10- 02T13:56:36.078+01:00&lt;/Timestamp&gt;   &lt;Originator&gt;unknown&lt;/Originator&gt; &lt;/AuditTrailEntry&gt;</pre>	<pre>&lt;event&gt; &lt;string key="org:resource" value="unknown"/&gt; &lt;date key="time:timestamp" value="2012-10- 02T14:56:36.078+02:00"/&gt; &lt;string key="concept:name" value="5 PerformSurgery"/&gt; &lt;string key="lifecycle:transition" value="complete"/&gt; &lt;/event&gt;</pre>
<pre>&lt;AuditTrailEntry&gt;   &lt;WorkflowModelElement&gt;4 ExaminePatient&lt;/WorkflowModelElement&gt;   &lt;EventType&gt;start&lt;/EventType&gt;   &lt;Timestamp&gt;2012-10- 02T13:56:36.080+01:00&lt;/Timestamp&gt;   &lt;Originator&gt;unknown&lt;/Originator&gt; &lt;/AuditTrailEntry&gt;</pre>	<pre>&lt;event&gt; &lt;string key="org:resource" value="unknown"/&gt; &lt;date key="time:timestamp" value="2012-10- 02T14:56:36.080+02:00"/&gt; &lt;string key="concept:name" value="4 ExaminePatient"/&gt; &lt;string key="lifecycle:transition" value="start"/&gt; &lt;/event&gt;</pre>

**Fig. 3.15** Event log fragments (MXML, XES) reflecting the process fragment depicted in Fig. 3.14

One possible line of action is to enhance log-oriented formats by table-oriented data. An example for such a hybrid format is the eXtensible Event Stream (XES). (For the XML schema of XES, we refer to [63].) Tools based on XES tools such as XESame<sup>5</sup> facilitate the conversion between table formats and log formats. Figure 3.15b shows the process execution data in the XES format that corresponds to the MXML data on the left side.

### Import of Process-Oriented Data into Log Formats

Process-oriented data might be logged by different information systems, e.g., ERP systems or workflow systems. Several tools have been developed that support the import of this source data into target formats such as MXML and XES. For an overview, see the ProMImport framework.<sup>6</sup> There are also tools that enable the import of CSV data, i.e., Nitro<sup>7</sup> and Disco.<sup>8</sup> Nitro and Disco provide support to define mappings between the columns of the source Excel or CSV file and the target log format. The precondition for this mapping is that the source data is already structured in a process-oriented way.

<sup>5</sup><http://www.processmining.org/xesame/start>.

<sup>6</sup><http://www.promtools.org/promimport/>.

<sup>7</sup><http://www.fluxicon.com/nitro/>.

<sup>8</sup><http://fluxicon.com/disco/>.

### 3.4.3 Summary: From Transactional Towards Analytical Data

An important decision within the data provisioning process is to choose adequate integration and analysis formats. The choice of the integration format is merely driven by the integration process and related considerations (cf. Sect. 3.5.1). By contrast, the analysis format should reflect the analysis questions to be answered. In some settings, integration and analysis format might be identical.

Overall, we distinguish between flat, multidimensional, and process-oriented formats as well as—orthogonally—between flat, hierarchical, and hybrid formats. It might become necessary to generate one format based on another one by applying mapping, aggregations, and transformations. These methods will be discussed in more detail within the next section on schema and data integration.

In addition to more traditional integration and analysis formats, such as flat or multidimensional tables, process-oriented formats, i.e., log formats, are becoming increasingly important. However, quality issues and the modeling gap between transactional data and target formats pose crucial challenges that will partly decide on the success story of process-oriented analysis.

## 3.5 Schema and Data Integration

Once the appropriate integration format is chosen, the possibly heterogeneous data sources must be integrated within this target format. Hence, in addition to the mappings, aggregations, and transformations between different formats as summarized in Fig. 3.9, we have to address the challenges of *schema integration* and *data integration*.

### 3.5.1 Schema Integration

*Schema integration* means to unite participating schemata  $S_1, \dots, S_n$  into one integrated schema  $S_{\text{int}}$  where  $S_{\text{int}}$  should meet the following criteria (based on [3]):

- Completeness: no information loss with respect to the entities contained within schemata  $S_i$ ,  $i = 1, \dots, n$ .
- Validity:  $S_{\text{int}}$  should reflect a real-world scenario that can be seen as a union of the real-world scenarios reflected by  $S_i$ ,  $i = 1, \dots, n$ .
- No contradictions within  $S_{\text{int}}$
- Minimality: no redundancies, every entity contained in  $S_i$ ,  $i = 1, \dots, n$  should occur just once in  $S_{\text{int}}$ .
- Understandability: the transformation and integration steps should be documented in order to enable the traceability and reproducibility of the result.

Why is building  $S_{\text{int}}$  often difficult? Remember that schemata  $S_1, \dots, S_n$  might stem from possibly heterogeneous data sources. This often results in a bunch of *conflicts* between the participating schemata. In general, the occurrence of *semantic and descriptive, heterogeneity*, as well as *structural* conflicts can be observed. In the subsequent paragraphs, we describe the different conflicts in more detail following [56].

Semantic and descriptive conflicts refer to the way people perceive the set of real-world objects to model. More precisely, a semantic conflict arises if modelers choose different entities to describe the same real-world scenario. Assume two schemata A and B describing patient administration. In schema A, the entity `Patient` is used, whereas in schema B, the entity `StatPatient` describes patients in a hospital scenario.

Descriptive conflicts happen if modelers use the same or similar entities but different attribute sets to describe these entities. For example, in schema A, a patient is described by `Name` and `Age`, whereas in schema B, the patient is described by `Social Insurance Number` and `BirthDate`. Heterogeneity conflicts occur if the schemata are defined using different formats, e.g., relational versus XML. Finally, structural conflicts arise if different constructs are used despite choosing a common format. An example would be if in XML schema A patient age is modeled as an attribute and in XML schema B patient age is modeled as an element. In particular, semantic and descriptive conflicts are hard to solve without any further knowledge. For semantic and descriptive conflicts, *ontologies* can be used to resolve conflicts such as homonyms and synonyms (e.g., by using `vehicle` instead of `car`).

The *schema integration process* as described in [3] comprises the following phases: (a) pre-integration, (b) schema comparison, (c) schema conforming, and (d) schema merging and restructuring. Schema merging and restructuring might be conducted iteratively if the result does not meet the criteria for the integration schema  $S_{\text{int}}$  as described above. Results of phase (d) might be even played back to schema comparison in order to, for example, gain more details on certain participating schemata.

Within the pre-integration phase, the participating schemata  $S_1, \dots, S_n$  are analyzed for their format and structure as well as their metadata. Another part of the pre-integration phase is to determine the *integration strategy*, particularly if more than two schemata are to be integrated. In [3], different strategies are introduced. One example is the “one-shot” strategy that aims at integrating  $S_1, \dots, S_n$  at once. By contrast, binary strategies would integrate  $S_1, \dots, S_n$  in a pairwise manner. Here, it can be distinguished in which order  $S_1, \dots, S_n$  are integrated. The order, in turn, might increase or decrease the influence of single schemata on the result  $S_{\text{int}}$ .

*Schema mapping* and *matching* are techniques that are applied during schema comparison and schema conforming. According to [50], a matching takes “two schemas as input and produces a mapping between elements of the two schemas that correspond semantically to each other.” Formally, a schema mapping can be defined as follows (adapted from [6, Chapter 4, pp. 82 ff]):

**Definition 3.1 (Schema Mapping)** Let  $\mathcal{S}$  be the set of all schemata of interest, and let  $S, T$  be two schemata in  $\mathcal{S}$ ;  $S, T \in \mathcal{S}$ . Let further  $A_S$  and  $A_T$  be the attribute sets of  $S$  and  $T$ , respectively. Assume that  $\mathcal{A}$  denotes the superset of all attributes or schemata in  $\mathcal{S}$ . Then function  $m$  maps schema  $S$  onto schema  $T$  as follows:

$$m : \mathcal{S} \times \mathcal{S} \mapsto 2^{\mathcal{A} \times \mathcal{A}}$$

$$m(S, T) := \{(a_S, a_T) \mid a_S \text{ corresponds to } a_T, a_S \in A_S, a_T \in A_T\}$$

Informally, a mapping  $m$  defines a set of attribute correspondences between schemata  $S$  and  $T$ . An attribute correspondence can be based on different criteria, but most naturally it means to find those attributes in  $S$  and  $T$  that describe the same real-world entity [45].

How can we determine a mapping  $m$  between schema  $S$  with attribute set  $A_S$  and schema  $T$  with attribute set  $A_T$ ? The idea is to build the cross product  $A_S \times A_T$  between all attributes in  $A_S$  and  $A_T$ . For each pair, calculate its similarity, e.g., regarding attribute name or stored data. Then, choose a mapping based on the most similar pairs until a certain threshold. Additionally, certain constraints can be considered (e.g., attribute A.a must not correspond to attribute B.c). Doing this manually can become quite tedious [50]. Thus, tool support for defining schema mappings can be engaged.

Schema matching approaches try to find schema mappings automatically [43]. Finding correspondences is based on similarity measures. These measures can be based on (attribute) labels, schema structure (e.g., the neighborhood of attributes), and instances, for example, attribute type [43, 50]. Automatic finding (supported by tools) is mostly confined to find similar labels based on similarity metrics such as string edit distance [21]. Newer approaches base structural schema matching on the graph structures reflecting the schemata [43].

Tool support for schema matching is provided, for example, by Altova MapForce,<sup>9</sup> Microsoft BizTalk, and SAP NetWeaver, i.e., process integration from commercial source [6]. Open-source or research tools for schema mappings are COMA 3.0<sup>10</sup> and Protégé<sup>11</sup> together with plug-in Prompt.<sup>12</sup> Specifically, Protégé provides support for ontology matching [6].

In the next paragraphs, we discuss resolution strategies for heterogeneity conflicts that arise if the participating schemata  $S_1, \dots, S_n$  are of different formats.

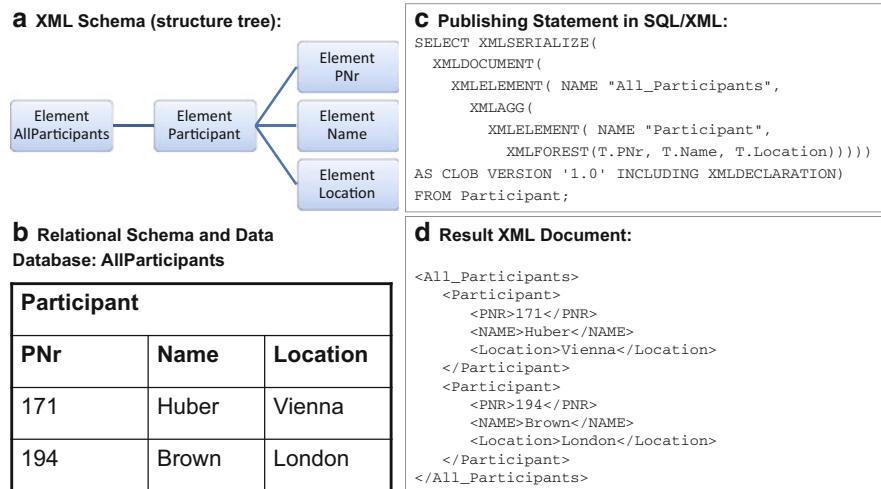
In order to integrate two heterogeneous schemata, a common data model has to be chosen, mostly one of the two existing ones. In the following, we elaborate on two commonly used data models and their integration, i.e., bringing together relational and XML-based schemata. Besides the practical importance of both data models, their case shows a particular challenge, i.e., integrating a hierarchical data model

<sup>9</sup><http://www.altova.com/mapforce.html>.

<sup>10</sup><http://dbs.uni-leipzig.de/de/Research/coma.html>.

<sup>11</sup><http://protege.stanford.edu/>.

<sup>12</sup><http://protege.stanford.edu/plugins/prompt/prompt.html>.



**Fig. 3.16** Mapping relational schema to XML schema, example based on the SQL/XML standard

(XML) with a flat one (relational). For illustrating the challenges of integrating heterogeneous schemata, we showcase the integration of a relational schema within an XML-based schema.

In Fig. 3.16, the database table (b) is to be integrated with the XML document (a) reflected by its underlying structure tree. Note that the structures of both the XML document and the relational table are already aligned. The challenge is to construct the hierarchical structure of the XML document based on the flat relational data. This task is referred to as *structuring* [54], i.e., it has to be decided which of the database attributes has to be converted in elements (or attributes) and in which hierarchical order. A “cooking recipe” for this is provided in [42]: the database (here `AllParticipants`) is mapped onto the root element; then the table constitutes the child element of the root, and the attributes are mapped to leaf elements.

The other task is *tagging*, i.e., defining and inserting meaningful tags within the resulting XML documents. How structuring and tagging can be done “by hand” using database ingredients and tools is described in [54].

By contrast, the SQL/XML standard offers publishing functions that enable the extraction of relational data as XML documents. The example displayed in Fig. 3.16c applies the publishing functions as offered by DB2 [46]. The result is shown in Fig. 3.16d. Other database vendors offer similar functions.

Of course, Fig. 3.16 displays a toy example with already corresponding schemata. Hence, no schema mapping becomes necessary, and the integration can be done at data level. If we think of mapping different database tables with integrity constraints, e.g., primary or foreign keys, their mapping to XML becomes more complicated. The key reference, for example, can be expressed at the XML schema level using XPath expressions. The following statement would reflect attribute `PNr`

as the primary key. Foreign key relations can be expressed using `keyref`. Other definitions at XML schema level comprise NOT NULL, type declarations, and UNIQUE.

```
<xss:key name="PNr_key">
    <xss:selector xpath="/AllParticipants/Participant"/>
    <xss:field xpath="PNr"/>
</xss:key>
```

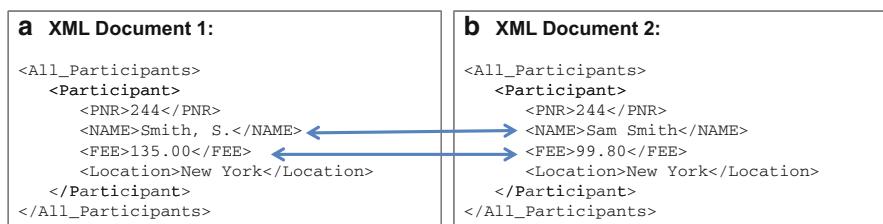
The challenge when storing XML data within relational databases is to flatten the XML data without losing the structuring information. Different strategies are conceivable here, ranging from generic tables that can store any XML document to specific mapping that can be defined manually or automatically [14]. Generic solutions store the content as well as the structuring information by bookkeeping parent/sibling information on the underlying XML tree (see, e.g., [27]). Despite the genericity of this approach, querying the stored data can become quite complex.

### 3.5.2 Data Integration and Data Quality

In Sect. 2.5.3, an introduction and general reflections on data quality in BI projects are provided. This section contains a more specialized discussion of data quality and its role in the data provisioning process.

After schema integration, there might still be inconsistencies at the data level, necessitating data integration actions. Take, for example, the two XML documents depicted in Fig. 3.17. Apparently, they adhere to the same XML schema but show conflicts at the data level, for example, encoding names in a different manner or displaying fees in a different currency. In order to integrate both XML files into one, the data conflicts have to be detected and resolved accordingly. This process is also referred to as *data fusion*, i.e., data from different sources being instance to the same real-world object is integrated to represent the real-world object in a consistent and clean way [12].

According to [43], the following problems at data level might occur: data errors (e.g., typos), different formats, inconsistencies (e.g., the zip code does not match the



**Fig. 3.17** Data integration problems (exemplified)

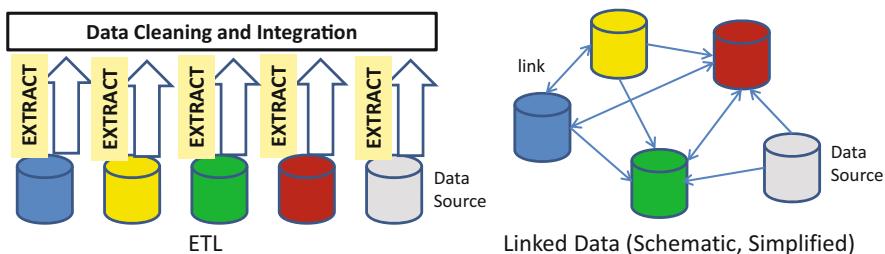
city), and duplicates. These problems are closely connected to the discussion on *data quality* in Sect. 2.5.3, i.e., is the data credible, relevant, and complete (no missing values, all real-world objects considered)? One measure to improve data quality is *data cleaning*, which aims at “detecting and removing errors and inconsistencies from data” [51]. The reasons for data errors are manifold and can be classified into the categories *single/multiple data sources* and *schema-instance level* [43, 51].

An example for schema-based data errors are missing integrity constraints; instance-based data errors might occur due to typos. Tackling data quality problems can be conducted in the following steps [43]: first of all, it is important to obtain an overview on the data and the possible problems. This is done in the *profiling* phase based on statistical or pattern-based analysis of the data using methods described in Chap. 4. In the *assessment*, certain conditions can be stated on data values (e.g., patient age <120). This can be also referred to as *plausibility* checks that are mostly based on rules.

Afterwards, concrete *measures* to fix data errors and remove error sources are determined. Finally, *monitoring* controls the success of the applied measures. As a preparatory step, data values might be *normalized* first [43] following principles of text mining (cf. Chap. 8). Examples for such actions are stemming and de-capitalization for string values (e.g., reducing “farmer,” “farming,” and “farmhouse” to the root word “farm” in the English language [57]) and unification of formats, for example, date formats or scales. These actions are well supported by tools.

### 3.5.3 Linked Data and Data Mashups

So far, we have discussed a process that extracts data from sources, cleans them, and integrates them outside the sources. A totally different idea is to use data sources that are integrated already, making extraction, cleaning, and integration tasks obsolete. This is the idea of *linked data*. The principle here is to link data sources that are available on the Web and by doing so, provide an integrated data source for later queries or analysis (see, e.g., [11]). The difference between the classical ETL process and the idea of linked data is illustrated in Fig. 3.18.



**Fig. 3.18** ETL and linked data

Linked data is realized through Semantic Web technologies such as Resource Description Framework (RDF)<sup>[13](#)</sup> and SPARQL (the query language for RDF).<sup>[14](#)</sup> RDF enables the definition of content with an additional definition of its meaning and allows for the possibility to link data to other data sources. Being able to define the meaning or semantics of data is an essential prerequisite for later data integration.

*Data mashups* enable the creation of new content/data based on existing and pre-processed data sources and are more suitable for nontraditional applications that are used by a rather low number of power users [16, 22]. Typical sources for data mashups are Web data and Web services that act as the components of the mashup. The data can be exchanged between the component either directly or in a mediated manner. One way to integrate, for example, Web services in a mediated way are Web service orchestrations or Web service choreographies [49]. An orchestration necessitates a central coordinator that is aware of the integration logic, whereas a choreography refers to the distributed execution of the integration logic without any central knowledge. Overall, data mashups are well suited for the service-oriented world and constitute a novel way to enable the participation of Web data sources in building up integrated databases. This is a similarity to linked data approaches as described above. However, as the usage of the term “integration logic” before already suggests, the basic integration questions such as how to parse and understand data sources and how to find appropriate mappings between them remain an important task in the context of data mashups as well.

### 3.5.4 Summary: Schema and Data Integration

In Sect. 3.5, we discussed several challenges and techniques in the context of schema and data integration. The presented techniques address merely syntactical conflicts such as mapping between different formats or models. Another challenging task is to achieve schema and data integration at a semantic level. It is necessary to determine the semantics of attributes in order to identify similar attributes as input for schema matching. Often, domain models or ontologies are suggested as support for this task, e.g., knowing that the attribute bank is used in a financial context instead of describing a scenario in flooding protection.

In addition to schema and data integration for later analysis tasks, integration and related questions as discussed in this section play an important role in other applications as well. An example is Web service integration [55], i.e., the joint provision of the functionalities of two web services.

---

<sup>13</sup><http://www.rdfabout.com/> and <http://www.w3.org/RDF/>.

<sup>14</sup><http://www.w3.org/TR/rdf-sparql-query/>.

### 3.6 Conclusion and Lessons Learned

Data provision constitutes an important prerequisite for any BI project. In most cases, it results in a complex and expensive task due to various reasons discussed in this section. Hence, it is recommended to plan for sufficient time and manpower. Additionally, not only BI experts but also domain experts should be included in the project team. It is of high importance to document every step in the integration process, possibly supported by tools. The analysis goals must be taken into consideration during the entire data provisioning process. The former two recommendations are part of the important issue of *data provenance* which includes not only the documentation of data provisioning but also of all subsequent preprocessing and analysis steps (see, e.g., [15]). Finally, it is very likely that maintenance of the integrated data might become necessary as, for example, data sources or even analysis goals change.

### 3.7 Recommended Reading

The following books and articles provide a good introduction into the topics of data integration. Bleiholder and Naumann (2011) provide an overview on different conflicts that might arise during an integration process as well as their solution at the data level. Inmon (2002) and Kimball and Ross (2010) address all aspects relevant to data warehousing. An introduction to log formats and building databases for process mining projects is provided by van der Aalst (2011). Finally, Bellahsene, Bonifati, and Rahm (2011) collect recent approaches on schema integration tasks.

- Bleiholder, J, Naumann F (2009) Data fusion. ACM Computing Survey 41(1): 1–41
- Inmon, WH (2002) Building the data warehouse. J. Wiley
- Kimball R, Ross M (2010) The Kimball Group reader: relentlessly practical tools for data warehousing and business intelligence (1). J. Wiley, New York
- van der Aalst WMP (2011) Process mining—discovery, conformance and enhancement of business processes. Springer, Heidelberg
- Bellahsene, Z, Bonifati, A., Rahm, E (eds) (2011) Schema Matching and Mapping. Springer, Heidelberg

## References

1. Adelberg B (1998) NoDoSE—a tool for semi-automatically extracting structured and semistructured data from text documents. SIGMOD Rec 27(2):283–294
2. Agrawal D, Das S, Abbadi Amr El(2011) Big data and cloud computing: current state and future opportunities. In: Ailamaki A, Amer-Yahia A, Patel JM, Risch T, Senellart P, Stoyanovich J (eds) EDBT’11: international conference on extending database technology. ACM, New York, pp 530–533

3. Batini C, Lenzerini M, Navathe SB (1986) A comparative analysis of methodologies for database schema integration. *ACM Comput Surv* 18(4):323–364
4. Baumgartner R, Gottlob G, Herzog M (2009) Scalable web data extraction for online market intelligence. *VLDB Endowment* 2(2):1512–1523
5. Becker M, Chamon P (2006) Process performance management—verzahnte Prozesse stets im Blick. *Fachbeitrag BI-Spektrum* 01:24–26 (in German)
6. Bellahsene Z, Bonifati A, Rahm E (2011) Schema matching and mapping. Springer, New York
7. Berchtold S, Böhm C, Kriegel H-P (1998) Improving the query performance of high-dimensional index structures by bulk load operations. In: Schek H-J, Saltor F, Ramos I, Alonso G (eds) *EDEB'98: international conference on extending database technology*. Lecture notes in computer science, vol 1377. Springer, Heidelberg, pp 216–230
8. Bernstein PA, Haas LM (2008) Information integration in the enterprise. *Commun ACM* 51(9):72–79
9. Bex GJ, Neven F, Vansumeren S (2007) Inferring XML schema definitions from XML data. In: Koch C, Gehrke J, Garofalakis MN, Srivastava D, Aberer K, Deshpande A, Florescu D, Chan CY, Ganti V, Kanne CC, Klas W, Neuhold EJ (eds) *VLDB'07: international conference on very large data bases*. ACM, New York, pp 998–1009
10. Beyer M (2011) Gartner says solving ‘Big Data’ challenge involves more than just managing volumes of data. Gartner. <http://www.gartner.com/it/page.jsp?id=1731916>. Accessed 19 May 2014
11. Bizer C, Heath T, Berners-Lee T (2009) Linked data—the story so far. *Int J Seman Web Inf Syst* 5(3):1–22
12. Bleiholder J, Naumann F (2009) Data fusion. *ACM Comput Surv* 41(1):1–41
13. Bonifati A, Casati F, Dayal U, Shan M (2001) Warehousing workflow data: challenges and opportunities. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT (eds) *VLDB'01: international conference on very large data bases*. Morgan Kaufmann, San Francisco, pp 649–652
14. Bourret R, Bornhovd C, Buchmann A (2000) Generic load/extract utility for data transfer between XML document and relational databases. In: *WECWIS'00: international workshop on advance issues of e-commerce and web-based information systems*. IEEE, New York, pp 134–143
15. Buneman P, Khanna S, Tan W-C (2000) Data provenance: some basic issues. In: Kapoor S, Prasad S (eds) *Foundations of software technology and theoretical computer science*. Lecture notes in computer science, vol 1974. Springer, Heidelberg, pp 87–93
16. Cappiello C, Daniel F, Matera M (2014) Mashups a journey from concepts and models to the quality of applications. *ICWE 2014 tutorial*
17. Cattell R (2011) Scalable SQL and NoSQL data stores. *SIGMOD Rec* 39(4):12–27
18. Chang F, Dean J, Ghemawat S, Hsieh WC, Wallach DA, Burrows M, Chandra T, Fikes A, Gruber RE (2008) Bigtable: a distributed storage system for structured data. *ACM Trans Comput Syst* 26(2):4
19. Chaudhuri S, Dayal U, Ganti V (2001) Database technology for decision support systems. *Computer* 34(12):48–55
20. Chaudhuri S, Dayal U, Narasayya V (2011) An overview of business intelligence technology. *Commun ACM* 54:88
21. Cohen W, Ravikumar P, Fienberg S (2003) A comparison of string metrics for matching names and records. In: Kambhampati A, Knoblock CA (eds) *IIWeb-03: proceedings of IJCAI-03 workshop on information integration on the web*, pp 73–78
22. Daniel F, Matera M (2014) Mashups: concepts, models and architectures. *Data-centric systems and applications*. Springer, New York
23. Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. *Commun ACM* 51(1):107–113
24. Dunkl R, Binder M, Dorda W, Fröschl KA, Gall W, Grossmann W, Harmankaya K, Hronsky M, Rinderle-Ma S, Rinner C, Weber S (2012) On analyzing process compliance in skin cancer treatment: an experience report from the evidence-based medical compliance cluster (EBMC2).

- In: Ralyte J, Franch X, Brinkkemper S, Wrycza S (eds) CaISE'12: international conference on advanced information systems engineering. Lecture notes in computer science, vol 7328. Springer, Heidelberg, pp 398–413
- 25. Facebook Key Facts (2012) <http://newsroom.fb.com/Key-Facts>. Accessed 5 Jan 2013
  - 26. Ferguson M (2014) Improving access to data for successful business intelligence. White Paper. Progress
  - 27. Florescu D, Kossmann D (1999) Storing and querying XML data using an RDMBS. *IEEE Data Eng Bull* 22:27–34
  - 28. Gaber MM, Zaslavsky A, Krishnaswamy S (2005) Mining data streams: a review. *ACM Sigmod Rec* 34(2):18–26
  - 29. Garcia-Molina H, Labio WJ (2006) Efficient snapshot differential algorithms for data warehousing. Technical Report. Stanford University
  - 30. Garofalakis M, Gionis A, Rastogi R, Seshadri S, Shim K (2003) XTRACT: Learning document type descriptors from XML document collections. *Data Min Knowl Disc* 7:23–56
  - 31. Golfarelli M, Maio D, Rizzi S (1998) The dimensional fact model: a conceptual model for data warehouses. *Int J Coop Inf Syst* 7(02n03):215–247
  - 32. Gretschmann M (2013) Everything new with big data. In: Keynote at the predictive analytics conference, Vienna, 25 September 2013 (in German)
  - 33. Grün C, Holupirek A, Kramis M, Scholl MH, Waldvogel M (2006) pushing XPath accelerator to its limits. In: Bonnet P, Manolescu I (eds) ExpDB'06: International workshop on performance and evaluation of data management systems. ACM, New York
  - 34. Günther C, van der Aalst WMP (2006) Generic import framework for process event logs. In: Eder J, Dustdar S (eds) Business process management workshops. Lecture notes in computer science, vol 4103. Springer, Heidelberg, pp 81–92
  - 35. Haerder T, Reuter A (1983) Principles of transaction-oriented database recovery. *ACM Comput Surv* 15(4):287–317
  - 36. Han J, Chen Y, Dong G, Pei J, Wah BW, Wang J, Cai YD (2005) Stream cube: an architecture for multi-dimensional analysis of data streams. *Distrib Parallel Databases* 18(2):173–197
  - 37. Hernandez MA, Stolfo SJ (1998) Real-world data is dirty: data cleansing and the merge/purge problem. *Data Min Knowl Discov* 2(1):9–37
  - 38. Inmon WH (2002) Building the data warehouse. Wiley, New York
  - 39. Kearny AT (2014) Beyond big: the analytically powered organization. Online Report, [http://www.atkearney.com/analytics/featured-article/-/asset\\_publisher/FNSUwH9BGQyt/content/beyond-big-the-analytically-powered-organization/10192](http://www.atkearney.com/analytics/featured-article/-/asset_publisher/FNSUwH9BGQyt/content/beyond-big-the-analytically-powered-organization/10192). Accessed 21 Nov 2014
  - 40. Kimball R, Ross M (2010) The Kimball Group Reader. Relentlessly practical tools for data warehousing and business intelligence. Wiley, New York
  - 41. Kimball R, Ross M, Thornthwaite W, Mundy J, Becker B (2011) The data warehouse lifecycle toolkit. Wiley, New York
  - 42. Klettke M, Meyer H (2003) XML and databases. dpunkt (in German)
  - 43. Leser U, Naumann F (2007) Information Integration. dpunkt (in German)
  - 44. Levene M, Loizou G (2003) Why is the snowflake schema a good data warehouse design? *Inf Syst* 28(3):225–240
  - 45. Li W-S, Clifton C (2000) SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data Knowl Eng* 33(1):49–84
  - 46. Moos A (2008) XQuery und SQL/XML in DB2-Datenbanken. Vieweg+Teubner
  - 47. NoSQL (2013) <http://nosql-database.org/>. Accessed 20 Jan 2013
  - 48. O'Callaghan L, Mishra N, Meyerson A, Guha S, Motwani R (2002) Streaming-data algorithms for high-quality clustering. In: Agrawal R, Dittrich KR (eds) ICDE'02: 18th international conference on data engineering. IEEE, New York, pp 685–694
  - 49. Peltz C (2003) Web services orchestration and choreography. *Computer* 36(10):46–52
  - 50. Rahm E, Bernstein PA (2001) A survey of approaches to automatic schema matching. *VLDB J* 10(4):334–350
  - 51. Rahm E, Do HH (2000) Data cleaning: problems and current approaches. *IEEE Data Eng Bull* 23(4):3–13

52. Santos RJ, Bernardino J (2009) Optimizing data warehouse loading procedures for enabling useful-time data warehousing. In: Desai BC, Saccà D, Greco S (eds) IDEAS'09: international database engineering and applications symposium. ACM, New York, pp 292–299
53. Seeger M (2009) Key value stores: a practical overview. medien informatik. slideshare.net, <http://de.slideshare.net/marc.seeger/keyvalue-stores-a-practical-overview>. Accessed 20 Jan 2013
54. Shanmugasundaram J, Shekita E, Barr R, Carey M, Lindsay B, Pirahesh H, Reinwald B (2001) Efficiently publishing relational data as XML documents. VLDB J 10(2–3):133–154
55. Shvaiko P, Euzenat J (2005) A survey of schema-based matching approaches. J Data Seman IV:146–171
56. Spaccapietra S, Parent C, Dupont Y (1992) Model independent assertions for integration of heterogeneous schemas. VLDB J 1:81–123
57. “Stemming”. Wikipedia, the Free Encyclopedia, <http://en.wikipedia.org/w/index.php?title=Stemming&oldid=535260860>. Accessed 28 Jan 2013
58. Stonebraker M (2010) SQL databases v. NoSQL databases. Commun ACM 53(4):10–11
59. Terdiman D (2012) Report: twitter hits half a billion tweets a day, CNET [http://news.cnet.com/8301-1023\\_3-57541566-93/report-twitter-hits-half-a-billion-tweets-a-day/](http://news.cnet.com/8301-1023_3-57541566-93/report-twitter-hits-half-a-billion-tweets-a-day/). Accessed 5 Jan 2013
60. van der Aalst WMP (2011) Process mining—discovery, conformance and enhancement of business processes. Springer, New York
61. van der Aalst WMP et al. (2012) Process mining manifesto. In: Daniel F, Barkaoui K, Dustdar S (eds) Business process management workshops. Lecture notes in business information processing, vol 99. Springer, Heidelberg, pp 169–194
62. Vassiliadis P, Simitsis A, Skiadopoulos S (2002) Conceptual modeling for ETL processes. In: Song I-Y, Theodoratos D (eds) DOLAP'02: ACM fifth international workshop on data warehousing and OLAP, pp 14–21
63. Verbeek E, Buijs J, Dongen B, van der Aalst WMP (2011) XES, XESame, and ProM 6. Inf Syst Evol 72:60–75
64. Walker M (2012) Data Veracity. [www.datasciencecentral.com/profiles/blogs/data-veracity](http://www.datasciencecentral.com/profiles/blogs/data-veracity). Accessed 12 Sept 2013
65. Wang C, Wang Q, Ren K, Lou W (2010) Privacy-preserving public auditing for data storage security in cloud computing, INFOCOM'10: 29th IEEE international conference on computer communications, pp 1–9
66. White T (2012) Hadoop: the definitive guide. O'Reilly Media, Sebastopol

# Chapter 4

## Data Description and Visualization

**Abstract** This chapter presents the basic principles for data description and visualization. After a brief introduction, we consider the description and visualization of structural properties of the business process in Sect. 4.2. The description and visualization for collections of process instances is treated in Sect. 4.3, which later outlines the essentials of interactive and dynamic graphics. Section 4.4 introduces frequently used visualization techniques together with applications to the use cases. Finally, Sect. 4.5 discusses certain aspects of infographics and reporting.

### 4.1 Introduction

Data description and visualization play a central role in BI and are used in all BI activities. At the beginning of any BI project, they are an essential part of *business and data understanding techniques* and support the understanding of the business process and the assessment of data. In the modeling phase, they provide input as *data preparation techniques*; in the analysis, they are not only used for the analysis as such, but also for the presentation of the results. Further, evaluation and reporting fundamentally rely on data description and visualization. In this spirit, one can understand data description and visualization orthogonally to the tasks of the iMine method outlined in Fig. 1.4. In contrast, one can understand data description and visualization as a specific analysis technique for achieving descriptive goals. Such an approach has the advantage that one is not forced to define a detailed analytical business model in advance and can combine the *business and data understanding task* with the *analysis task*. Such an approach is well known under the heading *Visual Analytics*, which is nowadays understood as a separate discipline within BI.

In agreement with this special position of data description and visualization, we structure this chapter in a format corresponding to the goals for data description and visualization as defined by information needs. With respect to information needs, one can distinguish between information about the structure of the business process, information about process instances, and information used for reporting. The following box outlines how this approach fits into the framework developed in Sect. 1.2.

### Data description and visualization in context of information needs

- *Data Description and Visualization for Business Processes:* The main goal is a description of the structure and usage of the business process from the production and organization perspective. The information used are structural data about the events and data describing the organizational infrastructure of the business. The events defining the business process are visualized together with the roles of the actors and the relation between the actors. Besides a static view describing the business process, we are interested in a dynamic view on the process which allows monitoring the execution of a process instance.
- *Data description and visualization for collections of business processes instances:* The main goal is the description and visualization of instances observed over some period of time. Due to the fact that customers frequently serve as process actors and generate the process instances, we take, in this case, the customer perspective. For information about the instances, we use primarily data in the cross-sectional view or in the state view on the business process. In case of data in the event view, the data have to be summarized in an appropriate way.
- *Data description and visualization for reporting:* The main goal is the presentation of the BI activities in high-level reports for deployment and application of the results. Usually, such reports are based on data in the cross-sectional view on the business. Depending on the BI scenario (cf. Sect. 1.2.1), this reporting has to put the findings in context of the overall business goals. Moreover, the assessment of data quality is of importance for proper understanding of the results.

The following section treats the description and visualization for analysis goals in the production and organizational perspective. Section 4.3 presents some general considerations for the visualization of collections of process instances in case of analysis goals in the customer perspective. Furthermore, we outline ideas for interactive and dynamic visualization. Section 4.4 shows how these ideas are realized in the context of basic visualization techniques and applies these techniques for the use cases introduced in Sect. 1.4. Reporting is sketched in Sect. 4.5.

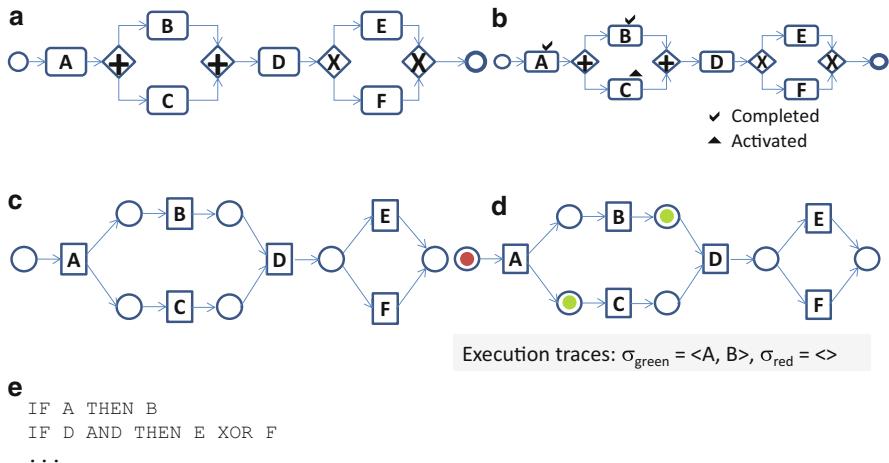
## 4.2 Description and Visualization of Business Processes

This section provides an introduction to the layout and visualization of business processes, structured along the phases of the process life cycle, i.e., design time and run-time. Specific visualization details in the context of process analysis and mining are discussed in Chaps. 7 and 8.

### 4.2.1 Process Modeling and Layout

At *process design time*, business processes are described in terms of process models. In most cases, these models are represented in a graphical way based on meta-models such as BPMN, Petri nets, or event-driven process chains (EPCs) as introduced in Sect. 2.3.2. Figure 4.1a shows graphical representations of a simple business process as a BPMN model and Fig. 4.1c as a Petri net.

The visualization of process models is, to some degree, determined by the different node and link types the process meta-model provides, for example, differentiating states and transitions for Petri nets and using distinct logical connectors such as AND and XOR splits/joins for BPMN. How to visualize these meta-model elements, i.e., what symbols are to be used, is also partly predefined by the meta-model or tools, respectively. If not defined in an imperative but rather declarative manner, process models might also be described by a set of constraints. An example project for visualizing declaratively modeled business processes is Declare.<sup>1</sup> Figure 4.1e provides a glimpse on how the Declare visualization looks like. The process model is defined on the basis of two rules. The first one states that after executing activity A, activity B must be executed, and the second that after executing D, either E or F must be executed.



**Fig. 4.1** Examples of different process models and their visualization. (a) BPMN model. (b) Process instance based on BPMN model. (c) Petri net model. (d) Process instance based on Petri net model. (e) Declarative model

<sup>1</sup><http://www.win.tue.nl/declare>.

### 4.2.2 *The BPM Tools' Perspective*

The *control flow* of a business process describes what has to be done (by activities) and in which order (in regard of dependencies between activities). Beyond the symbols provided by the different meta-models, some of the business process modeling tools offer layout options. Table 4.1 informs about such options concerning four well-known tools. Signavio,<sup>2</sup> for example, enables the automatic alignment of processes in a vertical and horizontal way. The same holds for ARIS Express.<sup>3</sup> In addition, ARIS Express supports horizontal and vertical arrangement of so-called satellite objects, i.e., objects that augment the control flow graphs.

Yet Another Workflow Language (YAWL)<sup>4</sup> provides an editor for designing processes that can be executed by the YAWL engine in the sequel. The YAWL editor offers several layout options with respect to the colors of process elements and background as well as the predefined augmentation of tasks, for example, if a task is to be executed manually or automatically.

Finally, the IBM Business Modeler Advanced Version 7.0<sup>5</sup> provides support for the aligning process automatically. In addition, user-defined icons for the process model elements, e.g., process activities, can be imported and used for the process model layout.

**Table 4.1** Layout of control flow in business process management tools (selection)

Signavio®	ARIS Express	YAWL4Study	IBM Business Modeler Advanced Version 7.0
Vertical and horizontal alignment	Vertical and horizontal alignment	Element alignment	Horizontal alignment
	Alignment of satellite objects		
	Background color	Background color	
		Background picture	
		Element color	Element color
Space between elements	Space between elements	Space between elements	Space between elements
			Import and assignment of user-defined icons

<sup>2</sup><http://www.signavio.com/>.

<sup>3</sup><http://www.ariscommunity.com/arис-express>.

<sup>4</sup><http://www.yawlfoundation.org/>.

<sup>5</sup><http://www-03.ibm.com/software/products/us/en/modeler-advanced/>.

### 4.2.3 Process Runtime Visualization

At *process runtime*, business processes reflected by process models can be instantiated and executed.<sup>6</sup> For each business case to be handled, e.g., a patient or a customer, a process instance is created, started, and executed based on the process model. The question is how to represent the execution data.

Process execution data can be reflected in terms of process logs (cf. Sect. 3.4.2) as well as by annotating the different process instance states to the underlying process model. Both Fig. 4.1b, d depict two process instances. The first is reflected by instance state annotation to the BPMN process model and the second by colored tokens on the underlying Petri net model.

In the case of the BPMN model, the annotations precisely state which process activities have been already completed, which are currently under execution, and which have not been started yet. Note that for each process instance, logically, a copy of the model plus the execution information has to be shown. In the case of the Petri net, the token colors represent the different instances on one process model, e.g., green tokens represent the instance state for patient Smith. However, with an increasing number of process instances, holding all instance information on one model by using different colors quickly becomes confusing.

Alternatively to annotating the process models with execution information, we can present execution logs in addition to the process model. Consider again Fig. 4.1d where the state represented by colored tokens is also captured within the two execution logs  $\sigma_{\text{green}}$  and  $\sigma_{\text{ref}}$ . Log information enables the reconstruction of the instance state whenever necessary.

### 4.2.4 Visualization of Further Aspects

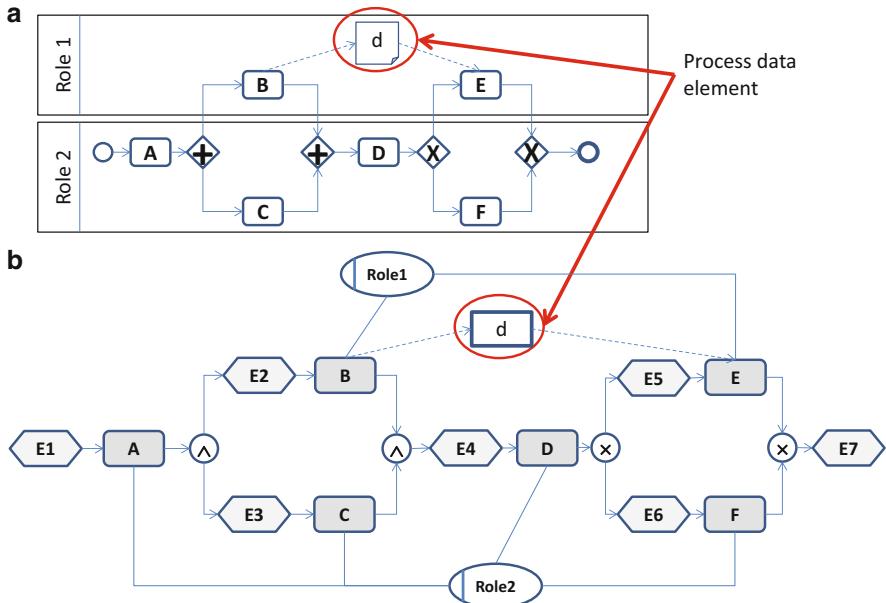
In addition to the control flow, business processes typically comprise data and organizational information.<sup>7</sup> Here, we can again see the connection to the BI perspective production (corresponding to the control flow), customer (corresponding to the data flow and decisions in the process), and organization (corresponding to the organizational perspective of the process). Different process meta-models offer different ways to describe and visualize process data and organizational information.

Data is usually depicted as objects using some related icon. Figure 4.2 shows how a data element  $d$  can be visualized in BPMN as well as in event driven process chains (EPC) models. Further, both models depict write-and-read access for data  $d$  in different ways (dashed versus full-line edges).

---

<sup>6</sup>In order to be executable, the meta-model used for process modeling has to have a defined operational semantics.

<sup>7</sup>Usually, more aspects are relevant for business processes at the modeling level and workflows at the implementation level, for example, regarding media or invoked services.



**Fig. 4.2** Visualization of process data and organizational structures. (a) BPMN model. (b) EPC model

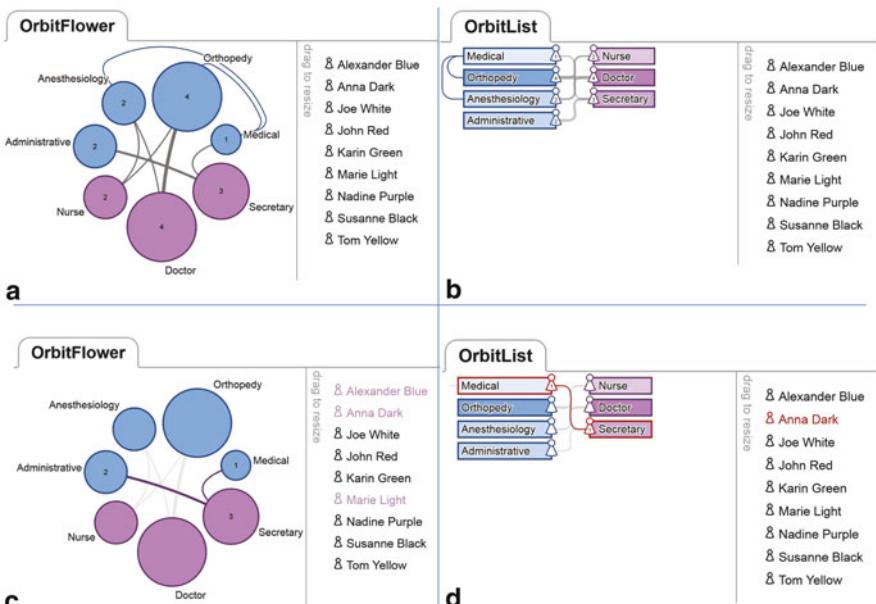
Organizational elements and structures can be modeled and visualized in different ways. One basic option is to maintain a separated *organizational model* and use its elements within the process model. Figure 4.2b displays an organizational model and a process model (both created using ARIS Express®) that use organizational elements comprised in the organizational model, e.g., Role 2. The depicted visualization as yellow rectangle is predefined by the meta-model (EPCs). Contrary to this kind of modeling (and subsequent visualization) is the use of *swimlanes* to describe organizational aspects within process model. Swimlanes partition the process models horizontally along different roles. In Fig. 4.2a, for example, swimlane Role 1 contains activities B and E. This means that Role 1 defines authorized users for performing activities B and E. Note that this is the same information as conveyed by the EPC model in Fig. 4.2b, i.e., by assigning organizational element Role 1 to activities B and E.

The node-link representation for organizational models as depicted in Fig. 4.2 and utilized by BPM tools can nowadays quickly become complex and hard to understand [14]. Hence, alternative visualizations might help to grasp the content of the model more easily and to foster the formulation of queries on the organizational model, for example, “how many performers assume the role doctor?” or “how many doctors do belong to the organizational unit Orthopedics?” Such questions can be relevant for security reasons as well. Think, for example, of a work assignment referencing to an empty role, i.e., a role for which nobody qualifies (e.g.,

after a change of staff). This immediately leads to the blocking of process execution during runtime and to possible security issues in the sequel, for example, if the affected task is then offered by the system to unauthorized performers [17].

Figure 4.3 shows two alternative ways of visualizing organizational information. On the left side, the so-called *OrbitFlower* visualization of a small medical organization is depicted, whereas the same scenario is visualized as *OrbitList* on the right side [14]. Within the *OrbitFlower*, roles and organizational units are depicted as circles of different colors (roles: purple, organizational units: blue). The size of the circle reflects the number of persons having the role or belonging to the organizational unit. The edges connect roles with organizational units meaning that there exists at least one person that has the role and belongs to the organizational unit. The thickness of the edge reflects the number of persons. Also, we can see all the persons contained in the organizational model. More over, selecting the role *Secretary* highlights the associated persons (cf. Fig. 4.3c) as well as the roles, organizational units, and edges associated with this role in the model.

The *OrbitList* arranges the roles and organizational units as rectangles, resembling a list structure. Again, the coloring is purple for roles and blue for organizational units. The edges connect roles with organizational units, reflecting the number of associations by their thickness. The number of persons having a certain role or belonging to a certain organizational unit is reflected by color intensity and



**Fig. 4.3** OrbitFlower and OrbitList for visualizing organizational models (a Orbit Flower, b Orbit List, c OrbitFlower with selection ‘Secretary’, d OrbitList with selection ‘Anna Dark’)

as a number in the associated user icon. When clicking on user Anna Dark, for example, the associated roles and organizational units are highlighted (cf. Fig. 4.3d).

Overall, the modeling and visualization of organizational information in PAIS are still done in a rather simple way. We still see a potential to better convey this important information to users and to offer more sophisticated ways of analyzing organization information. *OrbitFlower* and *OrbitList* offer means to visually inspect organizational models, also in an interactive way. In Chap. 8, further analysis methods on organizational information will be discussed that is also connected to the aspect of social networks.

#### 4.2.5 Challenges in Visualizing Process-Related Information

On top of the basic layout options for business process models and workflows presented in Sects. 4.2.1–4.2.4, the following challenges for process visualization have been discussed in literature [1, 2, 8, 13, 18]:

1. “Wallpaper processes” versus limited screen size: particularly in real-world settings, such as the automotive domain, process models might become quite large containing several hundreds of process activities [16]. In this case, visualizing the entire process models at once on screens of limited size is impossible. This problem becomes even aggravated if process models should be displayed on mobile devices. One solution is to use view [2] or abstraction [20] techniques to aggregate or abstract the process model information in a way that it fits the screen. One example is the hiding of a certain type of information such as data or organizational structures. Another technique is the hierarchical structuring of processes by subsuming parts of the process under complex activities.
2. A large number of process instances to be visualized at a time: besides the challenge of large process models, it might become necessary to convey information on multiple process instance execution within a short time frame, e.g., hundreds of executions events per second have to be monitored. Currently, it is investigated whether other techniques, such as sonification [8], can complement visualization approaches.
3. Visualizing process change information: change and evolution constitute major concerns in almost any process-aware application [25]. Since users are often supposed to define and apply changes to process models or running process instances, it is vital to support them by appropriate change visualizations. First approaches propose techniques such as change tracking graphs [9] or time lines [10].

#### 4.2.6 *Summary: Description and Visualization of Business Processes*

For conveying process-related information, it is important to distinguish between modeling and visualization aspects. Basically, the business process modeling language determines the elements used for the (mostly) graphical representation of the business processes as well as, in most cases, some kind of layout for these elements. BPMN, for example, offers a range of modeling elements with a set of icons representing these elements. Some business process modeling tools offer layout options, i.e., they enable to change the predefined layout in different ways. This also includes, for example, different colorings of process elements which refer to their visualization. As business processes comprise different aspects such as control flow, data flow, and organizational information, appropriate layout and visualization options for the different perspectives must be offered. This also holds for visualization of design and runtime aspects of business processes, e.g., visualizing a possibly large number of running process instances. For BI, conveying process-related information is crucial, particularly in the context of process change over time. A recent work [15], for example, showed how visualization techniques can be used as analysis tool for determining differences between process models. This can be helpful for BI questions related to business process compliance, i.e., “where did my business processes deviate from the prescribed models?”

### 4.3 Description and Visualization of Data in the Customer Perspective

The following paragraphs outline the principles for the description and visualization of data from collections of process instances in the customer perspective. The used data are based either on the state view or on the cross-sectional view. This means that we focus on data created by the execution of a number of business process instances. Section 4.3.1 discusses conventional graphical representation, and Sect. 4.3.2 considers interactive and dynamic graphics.

#### 4.3.1 *Principles for Description and Visualization of Collections of Process Instances*

The main challenge in the description and visualization of a collection of process instances is the definition of appropriate summaries and displaying these multivariate summaries in such a way that the complexity of the business process is captured in an understandable way for the observer. For readers, a graphic is usually more instructive than a table of numbers, and it is of utmost importance to give a correct

impression about reality. There are many pitfalls in the visualization of information. Sometimes, such errors are easy to notice; in other cases, it is rather difficult to find out how a graphic manipulates the perception of the viewer. Defining criteria for good graphics is not easy, because one has to take into account not only correct structuring of the information in the display but also aesthetics and perceptual phenomena. One can find a discussion about this issue in [22], and the Gallery of Data Visualization [5]<sup>8</sup> shows examples of good and bad graphics. Interestingly, Minard's map of Napoleon's Russian campaign published in 1869 is still considered as a masterpiece of visualization, although there is some imprecision in the map (cf. [25]). In our terminology, it would be a visualization of a number of state variables of a process.

Instead of aiming for a definition of a “good” graphical display, we want to present some structural considerations beyond the frequently used approaches of interfaces for designing graphics. The presentation is based on ideas of a grammar of graphs as described in detail in [25]. These concepts form the background for graphics in a number of statistical packages, for example, in SPSS/Clementine.<sup>9</sup> Here, we will follow the approach used in the R<sup>10</sup> package ggplot2 described in [24]. Rather independent from the software used for producing a graphic, the main topics of this approach seem useful.

### Visualization of a Collection of Process Instances

- *Definition of data:* The data are defined by a number of variables and have, in most cases, one of the following structures:
  - Multidimensional tables
  - Simple data structures
  - Complex data structures
- *Mapping of data:* Mapped to aesthetic attributes that are used for display.
- *Definition of layers:* Each graphic is constructed by a number of layers defined by:
  - The *Statistical transformation* which has to be displayed
  - The *Geometric object* used for displaying the statistics
  - The *Aesthetic mapping* for the geometric object
  - The *Position* for the geometric object
- *Coordinate system:* The coordinate system used for the graphic.
- *Facet specification:* Definition of small multiples for displaying subsets of the entire data set.

<sup>8</sup><http://www.datavis.ca/gallery/>.

<sup>9</sup><http://www-01.ibm.com/software/analytics/spss/products/modeler/index.html>.

<sup>10</sup><http://www.r-project.org/index.html>.

The starting point for all kinds of description and visualization is the selection of an appropriate data frame containing the variables of interest. This means that we have to decide what process instances we want to analyze and what attributes should be used. The selection of the instances is usually accomplished by defining a certain time interval for which data in the cross-sectional view or in the state view are analyzed. One can also use data in the event view provided that summaries are defined which describe properties of the events occurring during the execution of the process instances. The visualization of information about the instances in the event view will be treated in detail in Chap. 7. Besides the definition of a time interval for the collection of the instances, other criteria, such as the area for observation of the business process, certain organizational properties, or customer characteristics, can be used for data selection. The definition of the attributes of interest depends on the question one wants to answer. Moreover, we have to think about possible data transformations for existing attributes. Classical examples take the logarithm or the square root of a numeric variable.

In case of data in the state view, different transformations are possible. One can define summary characteristics of the time series from each instance like the mean, or the variance, or the duration of states above a certain threshold. In the case of regular time structures, one can also use transformations, e.g., first-order differences or summary measures like autocorrelation.

The data structures mentioned in the overview are those occurring most frequently in applications. Multidimensional tables, often called pivot tables, are defined by the values of qualitative variables called dimensions and by a summary attribute for the cells of the pivot table. This summary attribute is usually defined by the counts or means for a quantitative variable describing process instances. Simple data structures are defined by a matrix with rows representing the process instances and columns representing the values of the variables for the instances. Although the data have a simple matrix structure, there may be some internal structure, for example, groups of instances according to attributes like sex or age groups. Sometimes, such an internal structure may be hierarchically nested, for example, if we consider regions and outlets inside a region. In the case of complex data structures, we have a combination of the cross-sectional view and the state view on the process. As described in Chap. 3, for each instance, the data comprise attributes without temporal reference and state variables. For each state variable, the data contain a sequence of values of the state variable together with temporal information.

The mapping defines for each variable how it is represented in the graphics. Basic aesthetic attributes, which are used in a display, are the definition of an axis for the variable, color, size, and shape. Usually, a quantitative variable is mapped to an axis and qualitative variables to shapes. The mapping of a variable to the aesthetic attributes is called a *scale*. In order to explain this mapping to the viewer, one has to use guides which are either axis names or legends.

The definition of a layer starts with the specification of the *statistical transformations* one wants to display. Basically, these transformations correspond to the statistics of interest. The identity transformation corresponds to a display of

the values of a variable. The calculation of univariate characteristics of a variable, such as mean, variance, median, or quantiles, are summary transformations. The transformation necessary for displaying a histogram requires defining a partition of the range of a continuous variable by bins and counting the number of observations within each bin. A more complex statistical transformation is the calculation of a regression line. We will show examples of the most important and frequently used transformations in Sect. 4.4.

The statistical transformations are represented in a graphic using geometric objects. The basic geometric objects are points, text, lines, intervals, or polygons. These objects can be represented in different ways again using aesthetic mappings. For example, a line may be drawn with a certain line style, color, and thickness. Intervals can be mapped to rectangles with a certain color and boundary or to a line with error bars at the end. Often, aesthetic attributes help to represent the internal structure of the data. For example, we can use different colors or different point specifications for the representation of different groups. Another example is the representation of the importance of a point by the specification of size corresponding to the frequency of the value in the data.

There is a close relation between the statistical transformation and the geometric objects using specific aesthetic attributes. In Sect. 4.4, we will see how different geometric objects can be used for certain transformations. Basically, user interfaces for defining graphics offer selected statistical transformations together with a certain geometric representation as a guide for the definition of a graphic. Further, aesthetic attributes, like line styles, colors, legends, or axis labels, are options for the final layout of the graphic.

Position specification is sometimes necessary in order to avoid overplotting. A well-known example occurs in the case of bar plots where the bars of different groups can be stacked or placed side by side (clustered bars). Another example is *jittering* which means adding random noise to data points with the same coordinates in order to avoid overplotting.

Coordinates define the location of the points in space. The most frequently used coordinate system is the traditional Cartesian coordinate system. Using the well-known transformations of Cartesian coordinates in polar coordinates allows the representation of a graphic in polar coordinates. Examples of how to transform bar charts into pie charts are shown in [24].

Another useful technique for displaying complex structures are facets. Facets bind together different graphical displays into one graphic and are useful for displaying aspects of the data under different conditions. It is often more instructive to have different displays for groups side by side than putting everything into one plot using different colors or shapes. Facets are in close connection to the idea of conditioning plots or trellis plots.

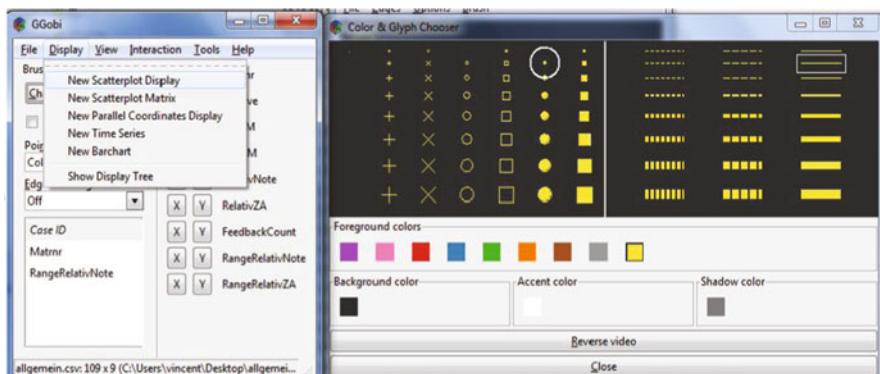
### 4.3.2 Interactive and Dynamic Visualization

For exploratory analysis, it is often useful to change graphics parameters, the geometry of plots, or the aesthetic components in order to get better insight into the data. Interactive and dynamic graphics support such activities, and we will briefly discuss the main elements of such activities. According to [6], “visual analysis typically progresses in an iterative process of view creation, exploration, and refinement,” i.e., data is not visualized in a single view, but using different views that serve different purposes and questions. In [6], a taxonomy of elements for interactive and dynamic analysis is defined, which consists of view specification, view manipulation, and the process and provenance of the conducted analysis.

View specification allows the interaction between the user and the visualization tool in the phase of definition of a visualization. Figure 4.4 shows an interface for view specification in GGobi.<sup>11</sup> It supports the specification of different variables and the mapping to the graphics, a selection of standard geometric objects for the envisaged statistical transformations, and the choice of details for aesthetic attributes like color, line style, and point style.

View manipulation often helps to get a better understanding of the data by selecting objects with specific features. Basic elements for view manipulation are queries, selection, linking, and zooming. Queries help to obtain exact information about data points, for example, the coordinates of a data point, or the values of the object represented by the data point. The standard method for the indication of such information in a graphic is a mouseover which is activated when the user moves the pointer over the object.

Selection helps to specify the subsets of interest in an interactive way. In particular, in the case of large data sets, it is often helpful to filter the data,



**Fig. 4.4** Interface for view specification in GGobi

<sup>11</sup><http://www.ggobi.org/>.

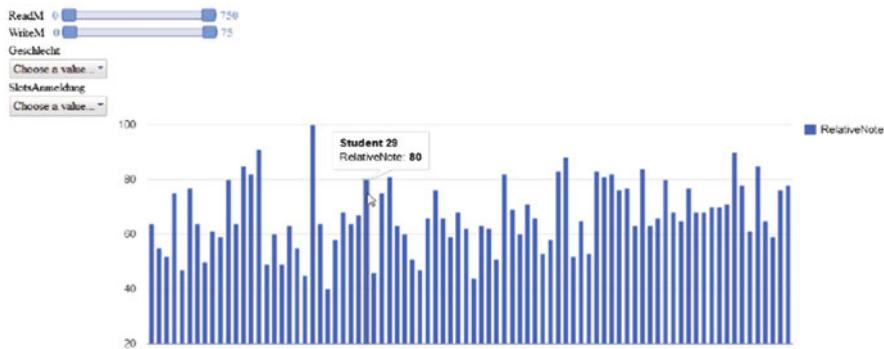
select variables, and reorder the variables in the display so that specific features become more visible. According to [26], we can distinguish between the individual selection of objects by specifying values of a variable and the selection of ranges for continuous variables, for example, an interval or a rectangle. A standard method for the visualization of a selection is highlighting. Special kinds of selection are brushing and painting. Brushing corresponds to a dynamic selection and painting to a permanent selection. Different tools can be used for this: A pointer selects single points a check box or a slicer can be used for selecting specific values of a variable and a drag box for selecting rectangular regions.

#### HEP Use Case: Students' Performance

In order to illustrate different selection techniques, Fig. 4.5 shows an interactive bar chart produced with HighCharts. The bars represent the percentage of completed exercises for each student, denoted by `RelativeNote`. With mouseover, one can show the percentage for each student. Two check boxes allow the selection of students depending on sex (`Geschlecht`) and the time of registration (`SlotsAnmeldung`). Two sliders `ReadM` and `WriteM` are used to dynamically select students according to activities in the forum.

In the case of viewing multiple facets of the data simultaneously, it is helpful to propagate the selections in one facet to the other facets. This technique is called linking. We distinguish between hot linking, which propagates a change in one facet automatically to all the other facets; warm linking, where the user decides which graphics will be linked; and cold linking, which links the different graphics only temporarily [23]. In the case of the simultaneous presentation of data and the results of calculations, for example, a regression line, the linking of the calculations is of interest. Such a linking allows the recalculations of models for a modified view specification after a view manipulation.

Another technique for large data sets, in particular for geographical data, is zooming. Besides basic zooming for changing the size, one can define logical and



**Fig. 4.5** Realization of different selection techniques with HighCharts

semantic zooming for better understanding of details and censored zooming for avoiding too much details (cf. [23]).

The process and provenance of an interactive analysis are important issues, particularly regarding the collaboration between different people. It helps in following the track of the different steps taken during the analysis process (cf. data provenance, Sect. 3.6).

In dynamic graphics, the basic elements of interactivity are augmented with additional elements like a rotating axis or a dynamic transformation of the axes. For the presentation of time series, *motion charts* have become popular in the last years as a special type of dynamic graphics. Motion charts combine the different techniques mentioned above and allow the dynamic visualization of the temporal behavior. Examples of motion charts applied to economic and sociodemographic time series can be found, for example, on <http://www.gapminder.org/>.

### 4.3.3 Summary: Visualization of Process Instances

In this section, we discussed the principles for designing graphics of instances of business processes either in the state view or the cross-sectional view. First, a visualization requires the definition of an underlying data structure which is mapped to aesthetic attributes used for the display. Usually, a graphic is composed of different layers and uses a coordinate system. Further, for displaying subsets of the entire data set the use of facets is helpful.

For interactive and dynamic visualization, a number of techniques for the iterative analysis were introduced. The most important are view specification, view manipulation, and the process and provenance of the analysis.

## 4.4 Basic Visualization Techniques

In the following, we will present some basic visualization techniques illustrated by the use cases introduced in Sect. 1.4. We will also comment on the structure and description of the data. For more details on the visualization of data, we recommend [23]. All plots in this section are produced using the R<sup>12</sup> statistical software.

---

<sup>12</sup><http://www.r-project.org/>.

#### 4.4.1 Description and Visualization of Qualitative Information

For qualitative information, the data structure usually is a pivot table with counts for the different attributes. The pivot table gives the frequencies of the different combination of values, either as absolute values or as percentages.

##### Bar Charts and Pie Charts

One variable can be visualized using a *bar chart* for absolute or relative frequencies or a *pie chart* for relative frequencies. For the visualization of more than one variable, a stacked or clustered bar chart can be used. In a stacked bar chart, the bars for the first variable are divided according to the frequencies of the second variable. In a clustered bar chart, the frequency bars for the values of the first variable are displayed side by side with the values of the second variable.

##### Mosaic Plots

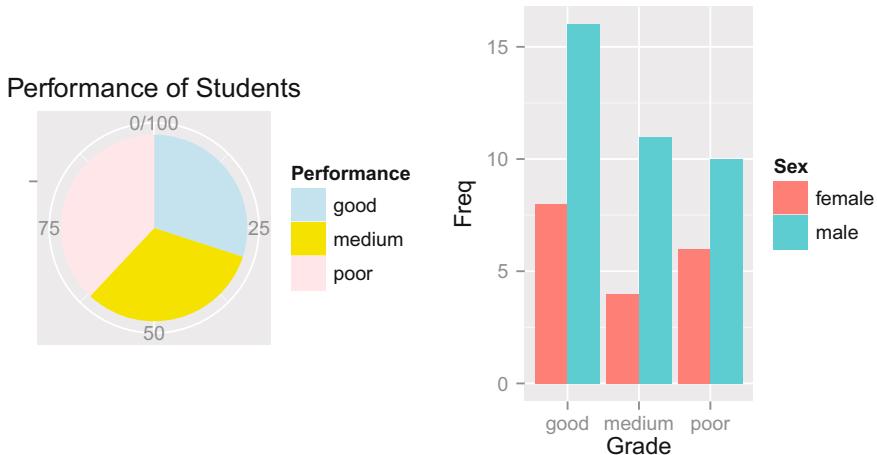
Many a time, a *mosaic plot* is more instructive than a bar chart for displaying two or more variables. In a mosaic plot, all data are represented as a square. The horizontal edge of the square is split according to the proportions of the first variable, and we obtain a number of rectangles with areas corresponding to the relative frequencies. Next, each rectangle is divided at the vertical axis according to the conditional probability of the second variable given the value of the first variable in the rectangle. A third variable is entered in the diagram by splitting again the horizontal axis of the rectangles according to the conditional frequencies given the first and the second variable. In this way, we obtain a representation of the table as rectangles, where the area of each rectangle corresponds to the frequencies of occurrence of the combination of values.

We will demonstrate the different visualizations with variables from the higher education use case.

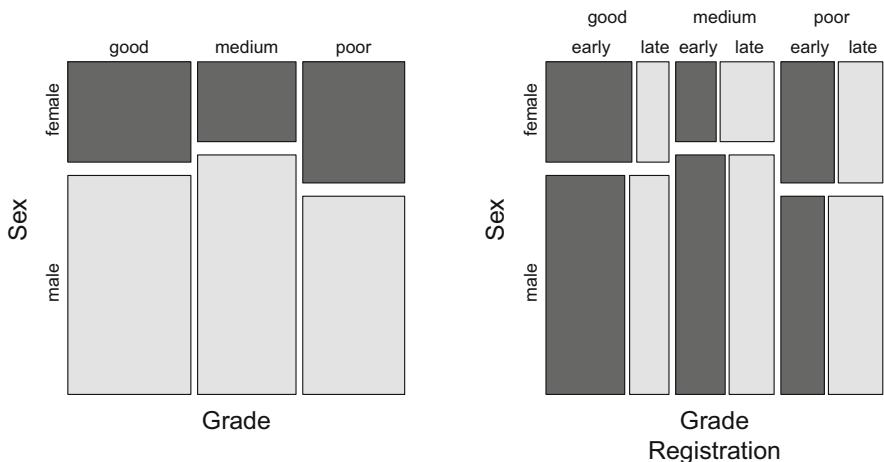
##### HEP Use Case: Students' Characteristics

In the case of the HEP data set (cf. Sect. 1.4.2), we start with a pivot table for the students grouped according to Sex and Grades (good, average, and poor) and with respect to their behavior regarding Registration (early and late). The graphic on the left in Fig. 4.6 shows the grades as a pie chart of the relative frequencies. On the right, there is a bar chart for absolute figures of the grades grouped by sex. The groups defined by sex are represented side by side. As we can see, the number of female students is less than 50 % of the male students.

A mosaic plot corresponding to the clustered bar chart is shown on the left side in Fig. 4.7. One can learn, for example, that female students scored more frequently good or poor than male students, but less frequently medium than male students. A mosaic plot also incorporating registration is shown on the right side in Fig. 4.7. The larger dark block for students with good grades indicates that for this group early registration is more frequent than late registration and that the behavior is almost the same for females and males.



**Fig. 4.6** Visualization of student characteristics: pie chart for grades and clustered bar chart for grades grouped by sex (R package `ggplot2`)



**Fig. 4.7** Mosaic plot for student characteristics (R graphics)

Many a time, pivot tables are used for the representation of summary measures of quantitative variables, for example, the sum or the mean of a variable. Such pivot tables can be visualized in a similar way using the summary measure instead of the counts for definition of the height of the bars.

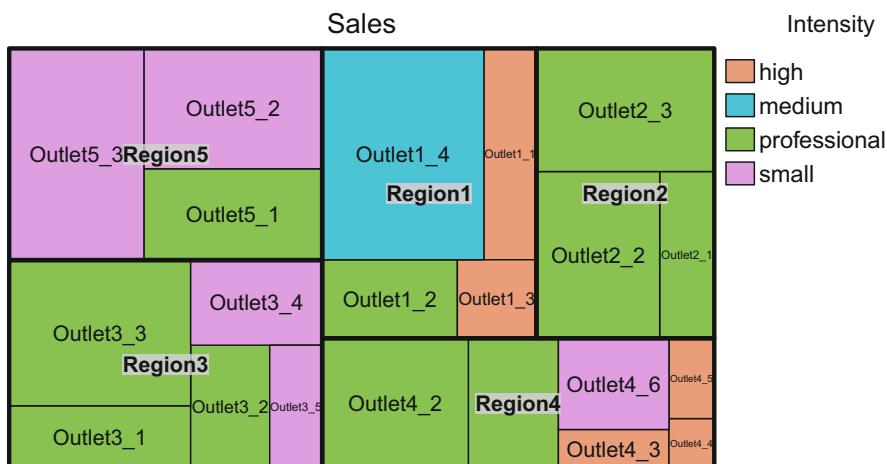
### Tree Maps

In the case of a qualitative variable with values defining a nested hierarchy of groups, a *tree map* is frequently used. The hierarchy is shown by nested rectangles, and the size of the rectangles represents the value of interest for the quantitative

variable. For displaying additional information, the colors of the rectangles can be used.

#### CRM Use Case: Tree Maps

Figure 4.8 depicts information about 21 outlets of the company located in five regions: region 1 has four outlets, region 2 has three outlets, region 3 has five outlets, region 4 has six outlets, and region 5 has three outlets. Obviously, the outlets are nested within the variable region and define a hierarchy. Each outlet defines a rectangle, and the area of the rectangles are defined by the values of the variable Sales. For example, in Region 1, there is one dominant outlet, Outlet1\_4, and in region 4, there is one outlet with very low sales. The colors of the rectangles inform about the dominant service intensity, Intensity, of the users in the outlet. Service intensity describes how intensive customers use the services offered by the shops and are classified into four intensity classes denoted by high, professional, medium, and small. For example, the outlet with the highest number of sales in region 1 mainly has customers with medium service intensity (blue rectangle), whereas the outlets with high service intensity do not contribute so much to sales. From the plot, one could derive that Sales seems rather independent from Intensity.



**Fig. 4.8** Tree map for sales and customers (R package `treemap`)

### 4.4.2 Description and Visualization of Quantitative Variables

A standard description of quantitative variables can be obtained by calculating the summary statistics such as mean, variance, standard deviation, minimum, maximum, and quartiles.

The visualization of one univariate variable can be based on different statistical transformations. A representation for the identity transformation displays values of the points along the  $x$ -axis of the plot. In order to avoid an overlapping of points, one can disturb the values slightly, a technique known as *jittering*. Frequently, such representation is not very informative except in cases with few observations. More useful is the display of the frequency distribution as a *histogram* or *boxplot*.

#### Histogram

The definition of a histogram starts with the assignment of nonoverlapping classes for the observations, so-called *bins*, which cover the entire range of the values. For each bin, the number of observations falling within the bin is counted. The height of the bars may be defined in different ways. The first one is defining the height of the bars by the absolute frequency (i.e., the counts), the second one is defining the height by relative frequency (i.e., percentage of observations within the class), and the third one is defining the heights in such a way that the area of the bars corresponds to the relative frequency of the bin. From a theoretical point of view, the best representation is the third one, which labels the height with the term density. The definition of the classes needs decisions about the center of the bars and the width of the bars. The perception of the histogram essentially depends on this specification and sometimes may be rather tricky.

#### Density Estimate

In case of sufficient number of observations, say at least a few hundred, an alternative to the histogram is a *density estimate*. The density estimate can be interpreted as a smoothed version of a histogram. In case of histograms using the third option for defining the height of the bars, the density and the histogram can be visualized as two layers in one plot. Such plots are useful for deciding about transformations of a variable, for example taking logarithms, in order to obtain data with some desired properties.

#### Boxplot

A boxplot is a schematic representation using the quartiles. The 0.25 quantile and the 0.75 quantile define a box of the central 50 % of the observations, and the median is marked within the box. Furthermore, whiskers are defined on both ends of the boxes. They mark the area in which all the data should fall provided the data follow a normal distribution. Data outside the hinges are marked and considered as candidates for outliers, which deserve special consideration. Boxplots are a useful visualization if one wants to schematically compare the distribution of one variable in different groups.

## QQ Plot

Frequently, one is interested in checking whether the assumption of a normal distribution for the data is justified. The assessment of this assumption can be done by using a *QQ plot*. This plot compares the quantiles of the observed variables with the quantiles of a normal distribution.

In the following, selected techniques are illustrated on the basis of the CRM use case (cf. Sect. 1.4.4).

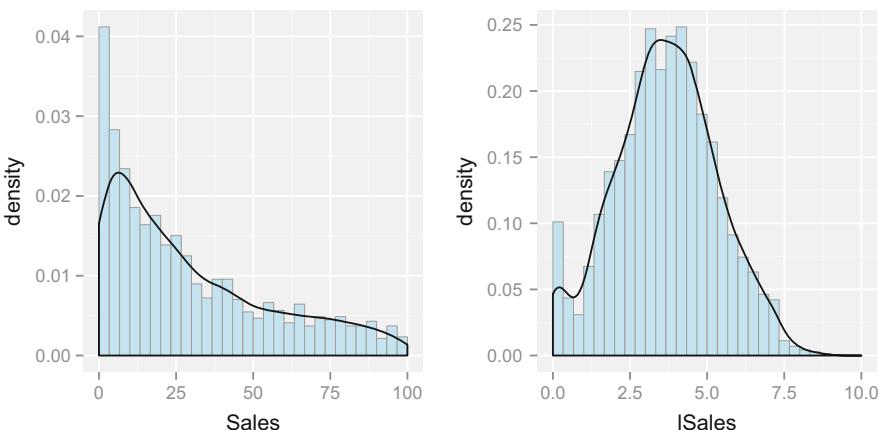
### CRM Use Case: Description of Variables

Table 4.2 contains a basic description for the variable `Sales` in the CRM example. The differences between mean and median, together with the asymmetric position of the quartiles around the median, indicate that there is some skewness in the data. Hence, we decide to consider also logarithms of sales (`Log-Sales`) to obtain a more symmetric distribution. A histogram of the variables `Sales` and `Log-Sales` is shown in Fig. 4.9. The figure clearly shows the effect of the transformation. However, in both cases, the assumption of normally distributed observations is not justified as one can learn from the QQ plot in Fig. 4.10. The main reason for deviation from the normal distribution are cases with low sales.

Figure 4.11 shows boxplots for `Sales` for a specific service, Service 3, for female and male customers. The graphic shows that extreme values occur for female customers and that there are a number of customers with missing values for sex.

## Contour Plots

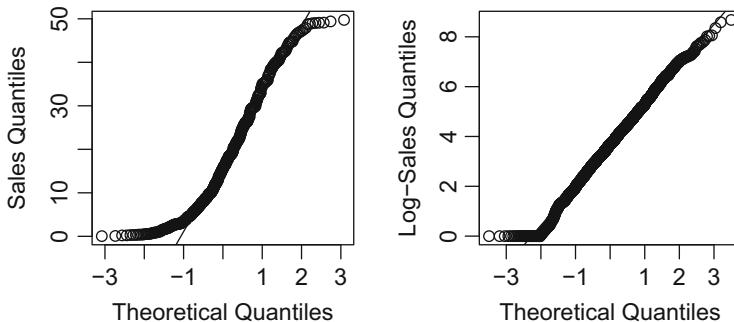
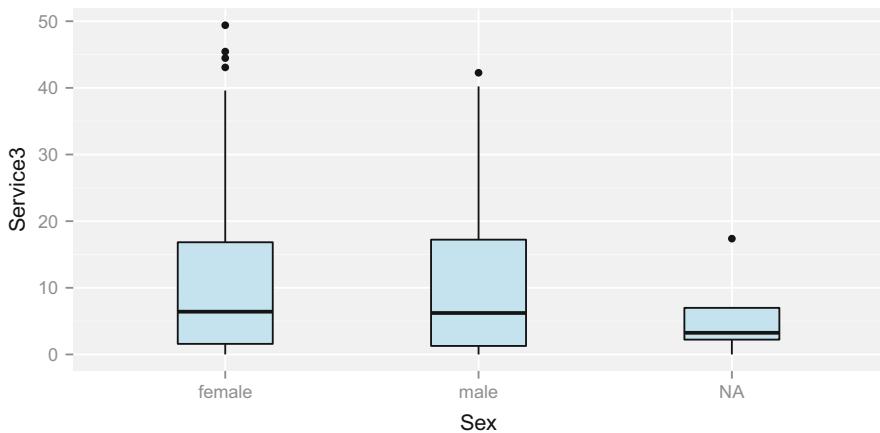
For the visualization of the joint distribution of two quantitative variables, the representation of the two-dimensional density by a *contour plot* can be instructive. The contour plot shows the level sets of the density, i.e., sets defined by the equation  $f(x, y) = c$ , as isolines for selected values of  $c$ .



**Fig. 4.9** Histogram and density for Sales and Log-Sales (R package `ggplot2`)

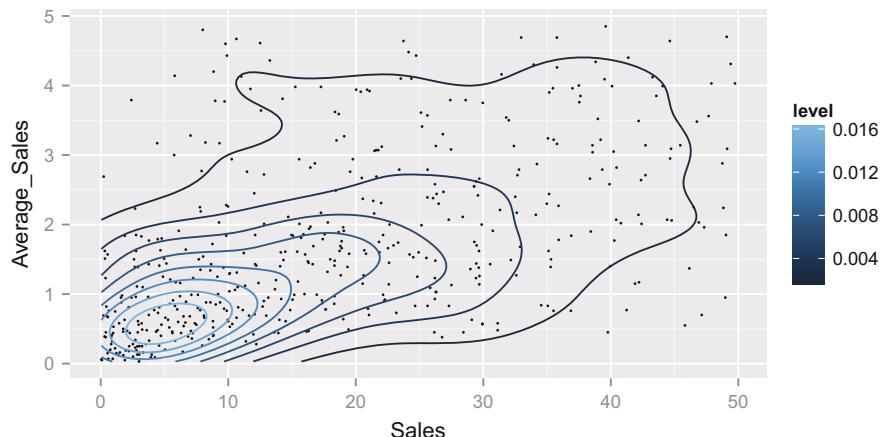
**Table 4.2** Basic description of sales from CRM use case

Mean	Variance	STDEV	Count	Missing
18.01	184.58	13.59	484	None
Min	Quartile 1	Median	Quartile 3	Max
0.05	6.23	15.75	27.48	49.74

**Fig. 4.10** Quantile plots for sales and log-sales (R graphics)**Fig. 4.11** Boxplots for Sales of Service 3 for female and male customers (R graphics)

#### CRM Use Case: Contour Plot for Sales and Average Sales

The contour plot shown in Fig. 4.12 shows the joint distribution of Sales and the sales indicator Average\_Sales for long-term customers. The plot shows that the peak of the density is around the values Sales = 5 and a Average\_Sales = 0.5. This can be interpreted as some kind of stability of customer behavior over time.



**Fig. 4.12** Contour plot of distribution of Sales and Average\_Sales (R package `ggplot2`)

#### 4.4.3 Description and Visualization of Relationships

##### Correlation

The basic description for relationships between quantitative variables is the correlation coefficient, which measures the strength of a (linear) relationship between two variables. Correlation can be classified as follows: *practically no correlation* (absolute values less than 0.2), *weak correlation* (absolute values between 0.2 and 0.5), *medium correlation* (absolute values between 0.5 and 0.8), and *strong correlation* (absolute values above 0.8). In the case of many variables, a visualization of the correlations by colors quickly provides a first impression about the relations between the variables. Such a representation is a special kind of a *heat map*, which represents the values of a matrix by colors. The choice of colors in a heat map is critical for the observer's perception, and different color schemes are used by different communities.

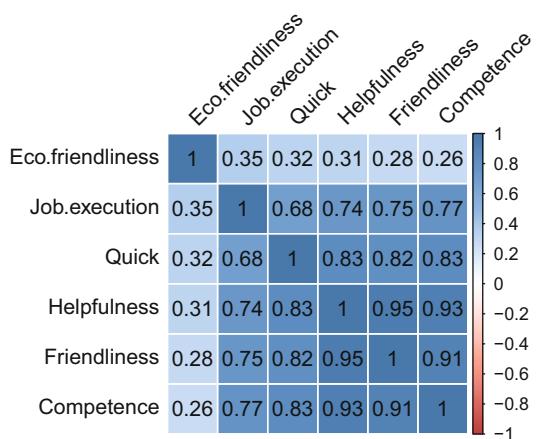
##### CRM Use Case: Correlation of Survey Questions

Figure 4.13 shows the correlation of six variables in the customer survey of the enterprise: competence (Comp), kindness (Kind), helpfulness (Help), service speed (Quick), ecological sustainability (Eco), and performance (Perf). It can be seen that there is, for example, a strong correlation between variables Comp and Kind reflected by the red color. All variables are at least weakly correlated.

##### Scatter Plots

For visualization of the relation between quantitative variables, the *scatter plot* is probably one of the best-known visualization techniques. If one wants to understand

**Fig. 4.13** Visualization of the correlations between survey variables (R package `corrplot`)

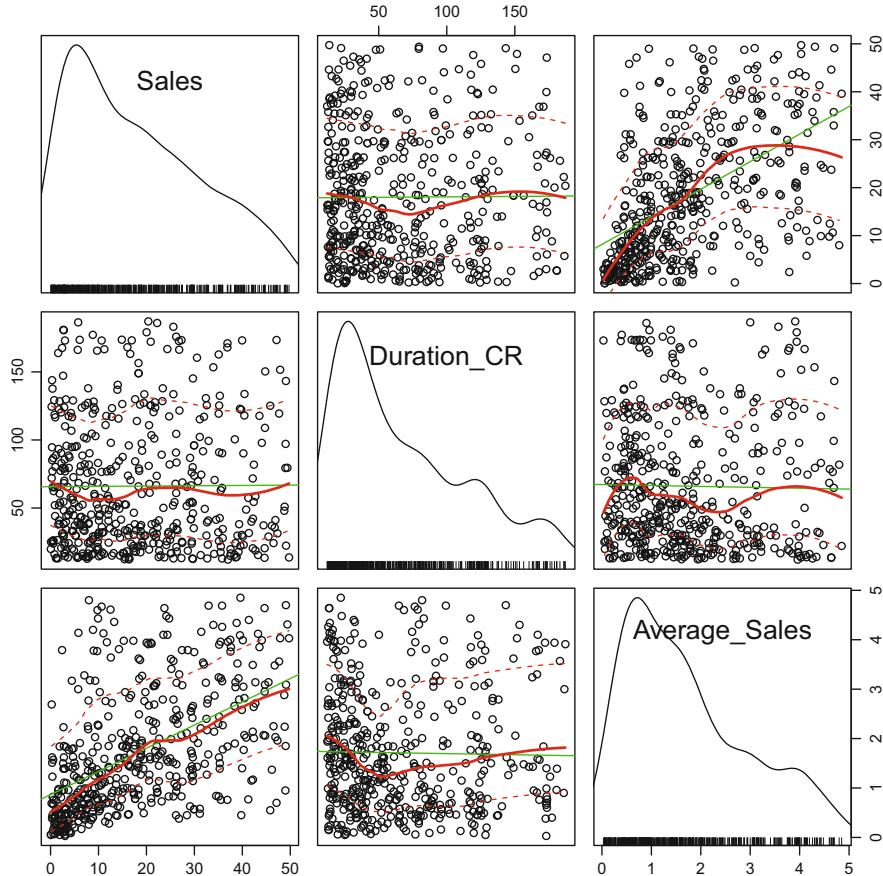


the relationships between  $k$  quantitative variables, the visualization of all  $k(k - 1)$  scatter plots can be done by a *scatter plot matrix*. The visualization of the scatter plots can be augmented by different layers. A popular additional layer in the plots are smoothing curves showing the relationship between the variables. Different types of smoothing curves are possible and will be discussed briefly in Chap. 5. Sometimes, confidence bands for the smoothing curves are displayed.

In the case of data grouped by a qualitative variable, one can use different colors or shapes for distinguishing the groups. Another informative element is a plot of the frequency distribution of the variables in the diagonal cells of the scatter plot.

#### CRM Use Case: Scatter plot for Sales Characteristics

Figure 4.14 shows the relationship between actual sales (`Sales`) of customers, the duration of the business relationship in months (`Duration`), and an indicator for past sales (`Average_Sales`). The densities in the diagonal show that all variables are skewed to the right. This is not surprising, because most private customers use the services only occasionally and have only a short-term relationship to the company and a rather moderate sales volume. The scatter plots itself show a positive relationship between average sales and actual sales. However, the relationship is rather scattered for larger sales. This can be explained by the fact that professional customers have a high variability in using the services. With respect to duration, there seems to be almost no correlation between the duration of the customer relationship and sales. In each scatter plot, there is a linear trend line and a smoothed trend line. The relation between average sales in the past and actual sales seems to be linear only for moderate sales, but for larger average sales, there seems to be almost no relationship. This fact is reflected by the bended shape of the smoothed curve.



**Fig. 4.14** Scatter plot matrix for Sales, Average\_Sales, and Duration of the business connection (R package car)

## Projections

Projections are useful for the representation of multivariate data in two or three dimensions. The basic idea of *principal component analysis* can be explained as follows. Given  $k$  variables  $X_1, X_2, \dots, X_k$ , one defines  $k$  new variables, so-called *principal components* with the following properties:

- Each new variable is a linear combination of the observed variables, i.e.,

$$PC_i = \alpha_{i1}X_1 + \alpha_{i2}X_2 + \cdots + \alpha_{ik}X_k,$$

- For the first variable, the coefficients are determined in such a way that  $PC_1$  explains as much as possible from the overall variance of the observations.

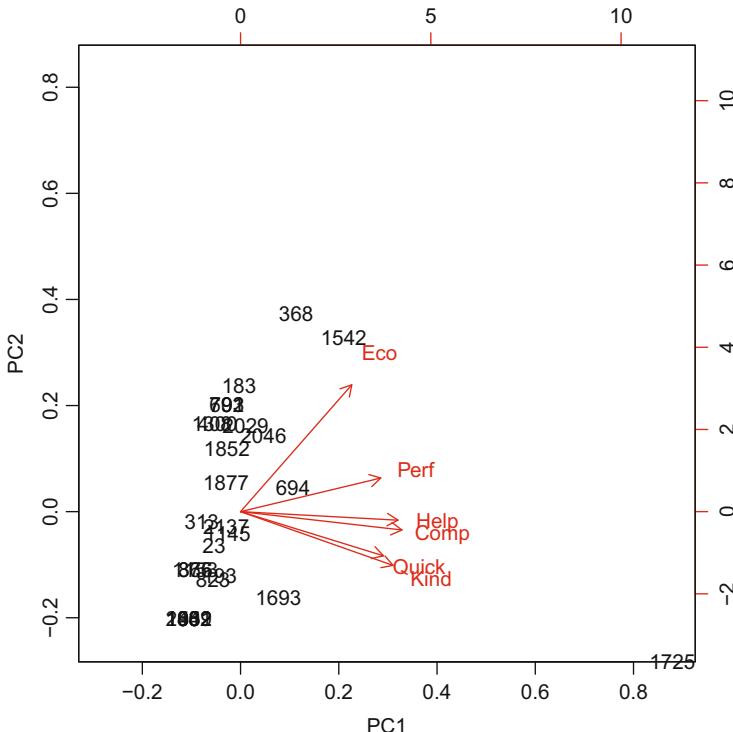
- (c) Given the new variables  $PC_1, PC_2, \dots, PC_i$ , the variable  $PC_{i+1}$  is defined in such a way that it is orthogonal to all previous variables  $PC_1, PC_2, \dots, PC_i$  and explains as much as possible from the overall variance of the observations.

Usually, the first two principal components capture a substantial percentage of the overall variability, say about 80 %. This allows a representation of the observations in two dimensions defined by the first two principal components. Plotting the data as a scatter plot with the first two principal components often gives an impression of the structure of the data. Even more instructive is a *biplot* which displays the observation points as well as the variables in the coordinate system defined by the first two principal components.

**CRM Use Case: Principal Components for Survey Questions**  
Let us demonstrate the ideas above by using the questions of the customer survey in the CRM use case. The correlation matrix is shown in Fig. 4.13. The biplot of the principal component analysis is represented in Fig. 4.15. The first principal component captures 74 % of the variability, the second one 12 %. The biplot shows that customers evaluate helpfulness and competence in a similar way. Also, the answers about quickness and kindness are rather similar. Because the vector for the variable `Perf`, i.e., performance, is shorter than the other ones, we conclude that the evaluation of performance is not so well represented by the first two principal components. Evaluation of the eco-friendliness of the service seems to be rather independent from the other evaluations. If one wants to interpret the new variables, one could say that the first component represents the evaluation of service quality, whereas the second component represents more the evaluation of customers with respect to ecology.

#### 4.4.4 Description and Visualization of Temporal Data

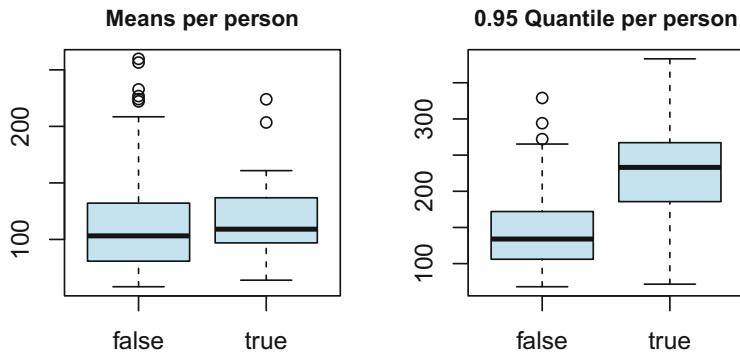
Temporal data is elaborated in detail in Chap. 6. Here we will discuss two different ways for their description and visualization. The first one is to define summary measures for each instance and use visualization techniques for cross-sectional data for these time-independent summary measures. The second approach is visualization of the state variable for each process instance as a function of time. The first approach is rather simple and its usefulness depends on the data. The second approach is closer to the structure of the measurements but more complicated in further analysis. We will illustrate both strategies using data from the pre-eclampsia use case (cf. Sect. 1.4.1):



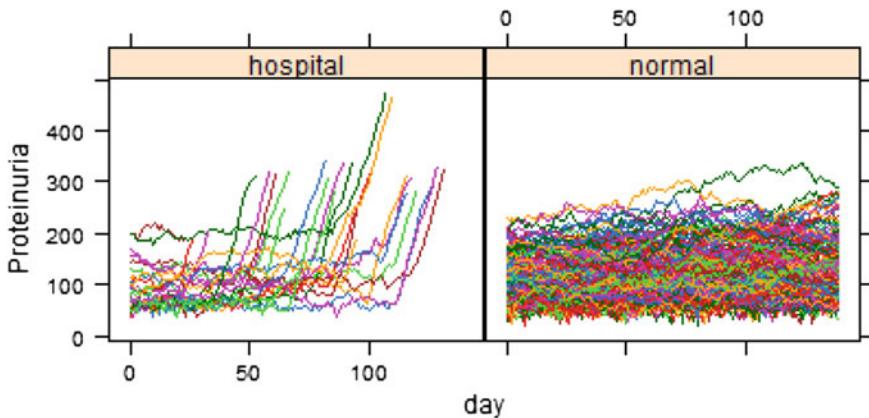
**Fig. 4.15** Biplot: principal component analysis for survey questions (R statistics)

**Pre-eclampsia Use Case: Description of Proteinuria**  
 Let us consider the variable Proteinuria which is measured several times during pregnancy. Figure 4.16 depicts boxplots for two different summaries of the Proteinuria measurements for the two groups of women defined by the decision about hospitalization. The boxplots for the means show that differentiation between the two groups defined by Hospital = true and Hospital = false is not obvious. If we produce boxplots for the individual 0.95 quantiles for each person, the differences in the two groups are clearly visible.

Figure 4.17 shows a plot of the values of Proteinuria along the time axes. For each person, a line plot is displayed. The left panel shows the development of Proteinuria for those persons who had to go to the hospital and the right panel for the other cases. Although individual curves for the normal cases are rather difficult to identify, it is obvious that there are different regimes in the development of the measurements for the two groups. For the group of women admitted to the hospital, the Proteinuria values start a steep increase between day 48 and day 100. In contrast, the women who experienced a normal pregnancy, did not show a steep increase, but a moderate course of the Proteinuria values.



**Fig. 4.16** Boxplots for means and 0.95 quantiles of the Proteinuria for persons in the two groups (R graphics)



**Fig. 4.17** Time sequences of Proteinuria for the persons in the two groups (R package lattice)

Note that such a representation does not depend on the time structure. We can apply it for a regular temporal structure with equally spaced points of measurement as well as time sequences with arbitrary time points of measurement.

#### 4.4.5 Interactive and Dynamic Visualization

All methods for visualization described up to now can be a starting point for interactive graphics. In this section, we will introduce *parallel coordinates* as an alternative plotting method for displaying multivariate data. This method is of interest in the case of high-dimensional data for interactive use. In a parallel coordinate plot, one positions the variables of interest equally spaced on the  $x$ -axis

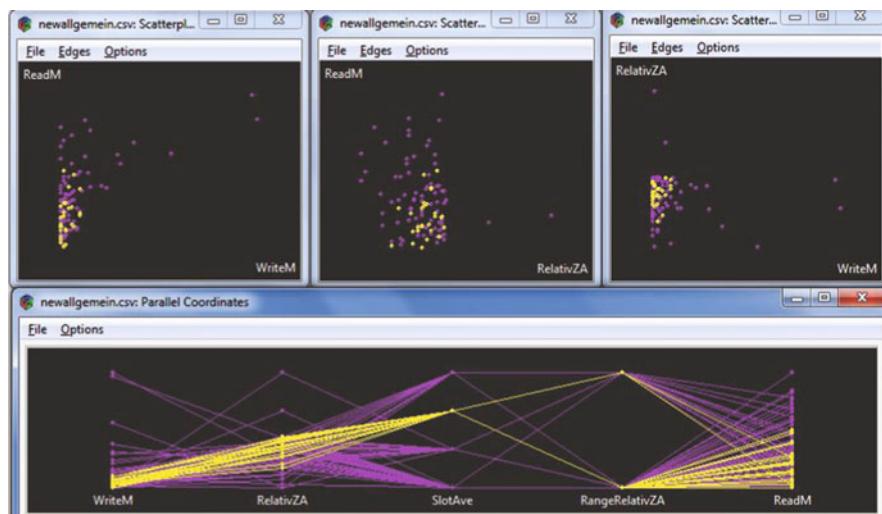
and graphs for each variable an ordinate. The observed values of the variables define the ordinate, and the instances are connected by a polygonal line. The interpretation of such a representation depends on the ordering of the variables. For example, a parallel connection of two adjacent dimensions for all observations suggests a positive correlation, and crossing of the lines suggests no correlation.

#### HEP Use Case: Parallel Coordinates

In order to illustrate parallel coordinates and linking, we use the forum activities, in particular, reading messages (ReadM) and writing messages (WriteM). For each student, the variables are defined as counts of these activities. Further, the variables are the effort taken for the lecture measured by the time students were online, RelativeZA, the registration behavior SlotAve with four different values (very early, early, late, very late), and Sex. Figure 4.18 shows the linking between parallel coordinates and scatter plots. The group with a very late registration is highlighted in all plots.

#### 4.4.6 Summary: Basic Visualization Techniques

Application of visualization techniques depends essentially on the type and number of variables and the complexity of the data structure. For qualitative variables, bar charts and mosaic plots are the most important techniques. In the case of visualization of continuous distributions, one can use histograms and boxplots



**Fig. 4.18** Linking of different views with GGobi

for one-dimensional distributions and contour or level plots for two-dimensional distributions. Boxplots are very useful for comparison of distributions within groups. For understanding the relationship between a number of quantitative variables, the scatter plot matrix is an important tool. In the case of many variables, dimensionality reduction based on principal components frequently allows an interesting interpretation of the data from a subject matter point of view. Another technique for visualization of many variables are parallel coordinate plots. In particular, for dynamic visualization, this technique is of interest.

For data in the state view, two different strategies for visualization can be used. The first is displaying the data as time series and the second is calculation of summary measures over time, which allows the application of visualization techniques for cross-sectional data.

## 4.5 Reporting

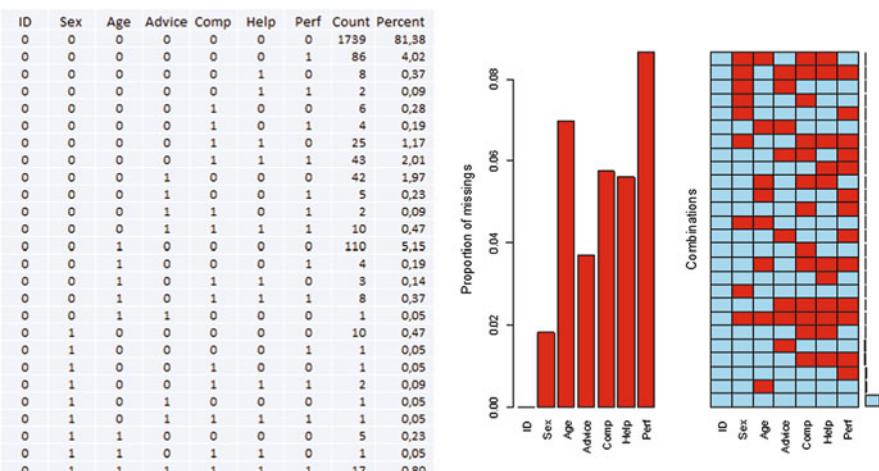
In this section, we discuss three different elements of reporting. The first concerns reporting about data quality with main emphasis on the visualization of the structure of missing values and summaries for different quality dimensions. The second considers aspects of high-level reporting (see Sect. 4.5.2), and finally we discuss briefly infographics in Sect. 4.5.3.

### 4.5.1 *Description and Visualization of Metadata*

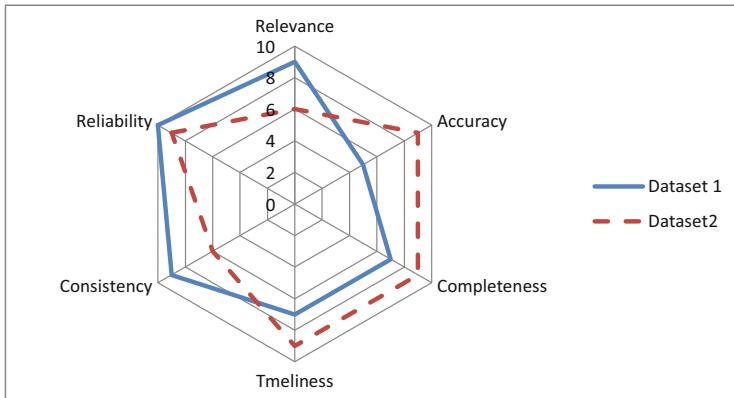
As already stated in Sect. 2.5.3, the quality of data is of utmost importance for BI applications. In order to obtain an impression of data quality, the visualization of different quality dimensions, also called quality criteria, can be useful. Crucial for analysis is knowledge about the structure of missing values, because they are a main reason for the lack of data quality. Further, they influence the quality dimension completeness and accuracy. Besides the percentage of missing values for individual variables, the combinations of missing values for different variables are of interest. In the following, we show for the CRM use case how one can organize information about such combinations.

**CRM Use Case: Description of Missing Values**

Figure 4.19 shows the structure of missing values for six questions in the survey about customer satisfaction. The data table on the left side can be interpreted as follows: the columns correspond to the variables ID, Sex, Age, Advice, Comp, Help, and Perf of the questionnaire. The meaning of ID, Sex, and Age is obvious; Advice provides the answers of customers to the question about advice given by the staff, Comp the answers to the question about the competence of the staff in the shops, Help the answers whether the staff is helpful, and Perf the answers to the question about the performance of the services. Each question was answered on a scale from -4 to 4. The rows represent the different combinations of missing answers to the questions, together with the counts of how frequently the combination occurred and the percentage of occurrence. 0 indicates that the value is not missing in the respective combination, and 1 means that answers of customers are missing. The most cases, i.e., 1,739, occur with no missing values for all questions (0 for all variables in this row). A percentage of up to 18 % of missing values is not unusual in such surveys. The bar chart in the middle shows the frequency of occurrence of missing values for the different variables. As we can see, the question about performance Perf has with 9 % the highest frequency of missing values. A visualization of the combinations of missing values is shown in the right plot. The rows are ordered from bottom to top along the number of cases for the different combinations. The bottom line reflects the 1,739 cases where no values are missing for all questions (all cells are colored in light blue). The next most frequent occurrence is for the combination of age with value 1 and all indicators with values 0, i.e., the cell for age is colored in red. From the table, we learn that this combination occurs in 110 or 5.15 % of all cases.



**Fig. 4.19** Visualization of the structure of missing values (R package vim)



**Fig. 4.20** Description of data quality with Excel radar plot

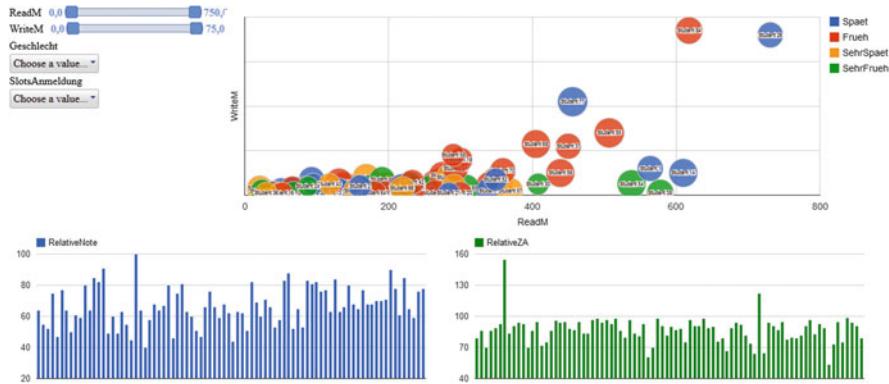
With respect to other quality criteria, a frequently useful representation is a *radar plot* as shown in Fig. 4.20. Each quality dimension of interest is represented by a corner in the radar plot. In Fig. 4.20, we used the six basic dimensions mentioned in Sect. 2.5.3. For each dimension, we define a scale for measuring quality, in our example, a scale from 0 to 10. This scale defines a spider net. Within this spider net, we insert for each data set under consideration the values of the quality dimensions and obtain in this way a polygon for each data set. Figure 4.20 shows polygons for two data sets, DataSet1 and DataSet2, represented by the blue and red lines, respectively. The figure shows that compared to DataSet 1, DataSet2 is better in the dimensions accuracy, completeness, and timeliness, but it has a lack of relevance and consistency, respectively. This shows that the decision about the usage of different data is often not easy and needs the judgment of the analyst.

### 4.5.2 High-Level Reporting

For deployment and usage of the outcomes of BI activities in decision making, one has to transfer the results into high-level reports for nonexperts, which summarizes the important analytical findings in a well-arranged and easily understandable way. Such summaries are frequently called *dashboard* or *business cockpit*. Basically, a dashboard can be defined as a graphical summary of the results of an analysis putting together the actual values of the quantities of interest and the behavior in the past. Using interactive and dynamic graphics also allows to focus on more details.

#### *Example 4.1 (Dashboard for Student Performance)*

Figure 4.21 shows a proposal for an interactive dashboard in connection with the HEP use case for lectures. The scatter plot at the top shows the read-and-write activities of the students in the forum. Students are represented as bubbles



**Fig. 4.21** Dashboard for lecture performance with HighCharts

with size defined by their performance in the course. We used a standardized performance evaluation based on percentages. The colors of the bubbles define the four registration groups (very early, early, late, very late). The left bar chart at the bottom shows the performance of the students, and the right bar chart shows the time effort of the students for the course. Two sliders allow selection of students with respect to read-and-write activities, and two drop-down lists allow selection of students with respect to gender and registration behavior.

Besides the results of BI analyses, which are mainly oriented towards analytical goals for understanding the behavior of KPIs in dependence of other quantities, such reports have to put the findings in the context of background information about the business, for example, financial information, organizational details, or strategic objectives. In the last two decades, the idea of control and measurement of performance has gained popularity under the terms business performance management (BPM) or corporate performance management (CPM), activities closely related to business intelligence [4]. The difference between BI and CPM is that BI is more analytically oriented with a focus on the past—you could say reactive—whereas CPM is more strategically, operationally, and proactively oriented. But BI and CPM have to support each other, and the used tools frequently are the same.

From a theoretical point of view, the most developed method for high-level reporting is the *balanced scorecard*. The balanced scorecard stood at the beginning of performance reporting and was designed primarily for reporting to the top-level management. The main focus was on four topics: financial information, information about customers and their perception of the business, information about internal business processes, and measures for the improvement of the business. In each topic, indicators were defined for measuring the relevant business goals in an appropriate way. This basic definition has been developed further to the so-called third-generation balanced scorecard, which has four main components [11]. For details, we refer to [12].

### Components of a Balanced Scorecard

1. *Destination statement:* It describes the organization at present and at a defined point in the future (midterm planning) in the four perspectives: financial and stakeholder expectations, customer and external relationships, process activities, and organization and culture.
2. *Strategic linkage model:* This topic contains strategic objectives with respect to outcome and activities, together with hypothesized causal relationships between these strategic objectives.
3. *Definitions of strategic objectives.*
4. *Definitions of measures:* For each strategic objective, measures are defined, together with their targets.

#### 4.5.3 Infographics

In the last 10 years, *infographics* have become popular. According to [19], an infographic is defined as “a visualization of data or ideas that tries to convey complex information to an audience in a manner that can be quickly consumed and easily understood.” Corresponding to this broad definition, there is ubiquitous use of infographics, and one can find examples in the public (e.g., maps for public transport), in news (e.g., weather forecasts or stock prices), in social media, or in scientific publications. The social network Pinterest,<sup>13</sup> originally designed as an exchange platform for pictures, gives a good impression of the multitude of infographics for different topics. These different examples impressively demonstrate that infographics target at different audiences with different intentions in mind. Common to all these applications are the following three goals:

1. *Appeal:* An infographic should engage the intended audience.
2. *Comprehension:* The viewer of an infographic should understand the information easily.
3. *Retention:* The information provided by an infographic should be remembered by the viewer.

The importance of these goals depends on the application. For BI applications comprehension is probably the most important goal, followed by retention and by appeal.

In connection with infographics for business information, three main topics are defined in [19]: numbers and concepts, how things work and are connected, and

---

<sup>13</sup><https://de.pinterest.com/>.

visualization of time series and maps. Let us show how the ideas of this chapter are related to these topics:

- *Numbers and concepts*: For presentation of numbers, the basic methods for the visualization of statistical distributions are the contents of Sect. 4.3. For the presentation of concepts defined by text data, we will show visualization techniques in Chap. 8 and an application in Chap. 9. What is missing are representations of concepts by techniques like metaphors or cartoons.
- *How things work and are connected*: The visualization of business processes treated in Sect. 4.2 is a good example for this theme. Besides business processes, we have introduced tree maps as the basic tool for the presentation of hierarchies in data. Other methods for the presentation of the organizational perspective will be introduced in Chap. 8.
- *Visualization of time series and maps*: For the visualization of temporal data, one can find only limited information for visualization of temporal information in Sects. 4.3.2 and 4.4.4. For visualization of map data, we refer to [7].

Hence, for most of the topics, a number of basic techniques are discussed in this chapter and in Chap. 8. However, we presented only the basic elements of simple infographics and to a limited extent the design of dashboards and reports as examples of more complex graphics. Customizing infographics for a specific audience needs additional design considerations. An example for such design guidelines can be found in [21].

An open source tool for creating infographics is Piktochart.<sup>14</sup> Other tools are ManyEyes,<sup>15</sup> Tableau Public,<sup>16</sup> or Gapminder.<sup>17</sup>

#### 4.5.4 Summary: Reporting

For high-level reporting, two aspects are of main interest. The first is representation of data quality which is basic for data profiling. Important aspects in this case are understanding the structure of missing values and assessment of the data with respect to the different quality dimensions discussed in Sect. 2.5.3. The second aspect is reporting for management decisions. Two standard tools are dashboards for visualization of KPIs and the balanced scorecard for understanding the results of a BI project in the framework of the overall business. A more visual representation of facts can be given by infographics.

---

<sup>14</sup><http://piktochart.com/>.

<sup>15</sup><http://www-969.ibm.com/software/analytics/maneyes/#/>.

<sup>16</sup><http://www.tableausoftware.com/public/>.

<sup>17</sup><http://www.gapminder.org/>.

## 4.6 Recommended Reading

A pioneering book for the visualization of quantitative information is Tufte (2002). For visualization of data in the cross-sectional view and in the state view, we used, besides the traditional high-level graphics functions of R, the packages `lattice` and `ggplot2`. For the theory behind `ggplot2`, we refer to Wickham (2009). An introduction to the use of `ggplot2` with many examples is [3]. Kriglstein (2012) provides a literature review and overview on layout and visualization of business processes.

- Chang W (2013) R graphics cookbook. O'Reilly Media, Sebastopol, CA
- Kriglstein S, Rinderle-Ma S (2012) Change visualizations in business processes—requirements analysis. In: Richard P, Kraus M, Laramee RS, Braz J (eds) GRAPP/IVAPP'12: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, SciTePress, pp. 584–593
- Tufte E (2002) The Visual display of quantitative information. Graphics Press, Cheshire, Connecticut
- Wickham H (2009) ggplot2: Elegant graphics for data analysis. Springer

## References

1. Bobrik R, Reichert M, Bauer T (2005) Requirements for the visualization of system-spanning business processes. In: International workshop on database and expert systems applications, DEXA'05. IEEE, Los Alamitos, California, Washington, Tokyo, pp 948–954
2. Bobrik R, Reichert M, Bauer T (2007) View-based process visualization. In: Alonso G, Dadam P, Rosemann M (eds) International conference on business process management. Lecture notes in computer science, BPM'07, vol 4714. Springer, Heidelberg, pp 88–95
3. Chang W (2013) R graphics cookbook. O'Reilly Media, Sebastopol
4. Frolick MN, Ariyachandra TR (2006) Business performance management: one truth. *Inform Syst Manage* 23:41–48
5. Gallery of data visualization: The best and worst of statistical graphics (2014). <http://www.datavis.ca/gallery/>. Accessed 25 Nov 2014
6. Heer J, Shneiderman B (2012) Interactive dynamics for visual analysis. *ACM Queue* 10(2):30
7. Heer J, Bostock M, Ogievetsky V (2010) A tour through the visualization zoo. *Commun ACM* 53(6):59–67
8. Hildebrandt T, Kriglstein S, Rinderle-Ma S (2012) Beyond visualization: on using sonification methods to make business processes more accessible to users. In: International conference on auditory display, ICAD'12, pp 248–249
9. Kabicher S, Kriglstein S, Rinderle-Ma S (2011) Visual change tracking for business process models. In: Jeusfeld MA, Delcambre LML, Ling TK (eds) International conference on conceptual modeling. Lecture notes in computer science, ER'11, vol 6998. Springer, Heidelberg, pp 504–513
10. Kabicher-Fuchs S, Kriglstein S, Figl K (2012) Timeline visualization for documenting process model change. In: Rinderle-Ma S, Weske M (eds) EMISA 2012: Der Mensch im Zentrum der Modellierung. Lecture notes in informatics, vol 206, GI, pp 95–108

11. Kaplan RS, Norton DP (1992) The balanced score card—measures that drive performance. *Harv Bus Rev* 70(1):71–79
12. Kaplan RS, Norton DP (2004) Strategy maps: converting intangible assets into tangible outcomes. Harvard Business School Press, Boston
13. Kriglstein S, Rinderle-Ma S (2012) Change visualizations in business processes—requirements analysis. In: Richard P, Kraus M, Laramee RS, Braz J (eds) International joint conference on computer vision, imaging and computer graphics theory and applications, GRAPP/IVAPP'12. SciTePress, pp 584–593
14. Kriglstein S, Mangler J, Rinderle-Ma S (2012) Who is who: on visualizing organizational models in collaborative systems. In: International conference on collaborative computing: networking, applications and worksharing, CollaborateCom'12. IEEE, Los Alamitos, California, Washington, Tokyo, pp 279–288
15. Kriglstein S, Wallner G, Rinderle-Ma S (2013) A visualization approach for difference analysis of process models and instance traffic. In: Daniel F, Wang J, Weber B (eds) 11th International conference business process management. Lecture notes in computer science, BPM'13, vol 8094. Springer, Heidelberg, pp 219–226
16. Müller D, Reichert M, Herbst J (2007) Data-driven modeling and coordination of large process structures. In: Meersman R, Tari Z (eds) OTM confederated international conferences CoopIS, DOA, ODBASE, GADA, and IS 2007, OTM'07. Lecture notes in computer science, vol 4803. Springer, Heidelberg, pp 131–149
17. Rinderle S, Reichert M (2007) A formal framework for adaptive access control models. *J Data Semant IX*:82–112
18. Rinderle S, Bobrik R, Reichert M, Bauer T (2006) Business process visualization—use cases, challenges, solutions. In: Manolopoulos Y, Filipe J, Constantopoulos P, Cordeiro J (eds) International conference on enterprise information systems, ICEIS'06, pp 204–211
19. Smiciklas M (2012) The power of infographics: using pictures to communicate and connect with your audiences. Que Corp, Pearson Education Inc., Indianapolis
20. Smirnov S, Reijers HA, Weske M, Nugteren T (2012) Business process model abstraction: a definition, catalog, and survey. *Distrib Parallel Databases* 30(1):63–99
21. The Anatomy of Infographics: 5 steps to create a powerful visual (2014). <http://spyrestudios.com/the-anatomy-of-an-infographic-5-steps-to-create-a-powerful-visual/>. Accessed 2 Jan 2014
22. Tufte E (2002) The visual display of quantitative information. Graphics Press, Cheshire
23. Unwin A, Theus M, Hofmann H (2006) Graphics of large datasets: visualizing a million. Springer, New York
24. Wickham H (2009) ggplot2: Elegant graphics for data analysis. Springer, New York
25. Wilkinson L, Wills D, Rope D, Norton A (2005) The grammar of graphics. Springer, New York
26. Young FW, Valero-Mora PM, Friendly M (2006) Visual statistics: seeing data with dynamic interactive graphics. Wiley, New York

# Chapter 5

## Data Mining for Cross-Sectional Data

**Abstract** This chapter investigates methods for analyzing cross-sectional data, i.e., data which are represented in matrix form, where each row represents one process instance. The analysis methods can be grouped into supervised learning methods, also known as predictive analysis, and unsupervised learning. Under the term predictive analysis, we summarize analytical techniques for regression and classification, whereas in case of unsupervised learning, we present methods for cluster analysis. Section 5.1 gives an introduction to supervised learning and Sects. 5.2 and 5.3 present a number of techniques for regression and classification. Section 5.4 treats principles of unsupervised learning and techniques for cluster analysis.

### 5.1 Introduction to Supervised Learning

In this section, we will follow the approach of Chapter 7 in [14] where one can find a more detailed exposition of the subject. Our knowledge for supervised learning is provided by data in the cross-sectional view of process instances. Given this fact, we distinguish two types of variables: a number of *input variables*  $X = (X_1, X_2, \dots, X_p)$  and one *output variable*  $Y$ . The variables define the columns of the data matrix, and lowercase letters  $(y_i, x_i)$  are used for the rows of the matrix representing observed values of the  $i$ -th process instance. Note that  $x_i$  is a  $p$ -dimensional vector, but we omit double indices. The number of observed instances is denoted by  $N$ , and we frequently use the term *cases* or *observations* for the observed process instances. As denotation for an  $N$ -dimensional vector of the output values for all cases, we use  $\mathbf{Y}$ .  $\mathbf{X}_r$  stands for the observed values of the  $r$ -th input variable  $X_r$  and  $\mathbf{X}$  for the  $N \times p$  matrix of all observed values of the input variables. Scalar multiplication for  $p$ -dimensional vectors  $u$  and  $v$  will be denoted by  $u^T v$ .

The analytical goals in supervised learning are the goals defined in Sect. 1.2.4 as estimation and classification. The distinction in supervised learning is formally defined by the scale type of the output variable  $Y$ .

### Analytical Goals in Supervised Learning

- *Regression:* In this case, the output variable  $Y$  is a quantitative variable interpreted as *response*, and we want to learn a regression function  $Y = f(X)$  from the data  $(\mathbf{Y}, \mathbf{X})$ . The inputs are frequently called *explanatory variables*.
- *Classification:* Here, the output variable  $Y$  is a qualitative variable interpreted as class identifier, and we want to learn a classification function  $Y = f(X)$  from the data  $(\mathbf{Y}, \mathbf{X})$ , which allows a group assignment for the inputs.

In the formulation above, we used capital letters for denotation of the generic relationship between input variables and output variables. In case of the application of this relationship to a specific instance, we use small letters, i.e.,  $y = f(x)$ . The term supervised learning refers to the fact that we have data from the outputs which allow us to “learn” the function  $f$  from the data  $(\mathbf{Y}, \mathbf{X})$ . The learned function can be used afterwards to predict the output value for a new case with input values  $x_{\text{new}}$ . In case of estimation, the term *predictive modeling* is sometimes used for supervised learning. This indicates that one aims at predicting an output for a given input.

Successful learning of the function has to take into account that we only have partial information, because the data usually are a sample of all possible process instances (cf. Sect. 2.4.3). Moreover, as explained in Chap. 1, the cases are mostly generated by customers and show a variability due to incomplete knowledge about customer behavior and motivations for customer decisions. This implies that we have to use statistical or probabilistic structures in the modeling task. The standard approach in modeling is the definition of a class of model structures  $\mathcal{F}$  as candidates for the function  $f$ . Typical examples of model classes are linear functions in the input variables, decision trees based on the values of input variables, or classes of probability distributions with unknown parameters. In case of estimation, the standard interpretation for the model class is that the model function  $f(X)$  represents the expected (average) behavior of customers. In case of classification, the class assignment  $Y = f(X)$  is usually not deterministic but defines a conditional probability  $p(Y|x)$  for the classes given the input variables  $x$ .

Basically, we can understand the analysis task as finding the “best” function  $f$  within the class to achieve the analytical goal. The term “best” is made precise by using the concepts of statistical decision theory. This means that we define a loss function  $L(Y, f(X))$  which measures the error when we predict the output  $Y$  by the function  $f(X)$  and define the risk as expected loss  $E[L(Y, f(X))]$ . A well-known example of a loss function is the squared loss function  $L(Y, f(X)) = (Y - f(X))^2$ . In this case, the risk is the expected square error. Using the risk as a quantitative measurement for the error, the best model within the class is defined by that function  $f$  which minimizes the risk.

Because our data represent only a sample of all possible cases, we cannot calculate the exact risk. We have to use the empirical risk defined by

$$R_{\text{emp}} = \frac{1}{n} \sum L(y_i, f(x_i)). \quad (5.1)$$

The analysis task can be formulated as the following minimization problem: Find a function  $\hat{f}$  such that

$$\frac{1}{N} \sum L(y_i, \hat{f}(x_i)) = \min_{f \in \mathcal{F}} \frac{1}{N} \sum L(y_i, f(x_i)). \quad (5.2)$$

The empirical risk is also called *training error*, because it depends on the data from which we learn the function.

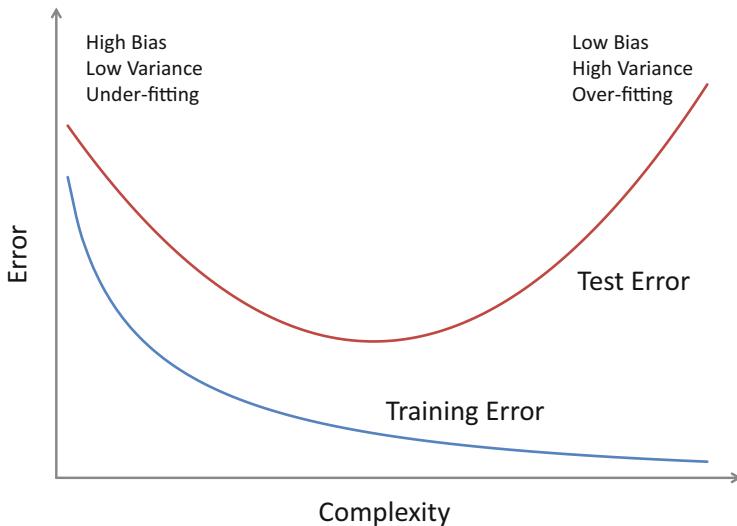
In BI applications, we are not so much interested in minimizing the empirical risk but in finding a realistic estimate for the true risk  $E[L(Y, \hat{f}(X))]$  of the model that measures the predictive power of the model. This true risk is also called *test error* or *generalization error* of the model. The two terms can be used interchangeably, but we prefer the term test error. The relation between training error and test error depends not only on the data and the definition of the loss function in the minimization problem in (5.2) but also on the choice of the model class defined by the analyst. We can use rather small model classes, defined by few parameters or large model classes characterized by certain properties of the function. Examples of such classes have been defined in case of regression in the introduction to statistical modeling in Sect. 2.4.4.

An important concept for the distinction of different model classes is *model complexity*. Intuitively, model complexity is defined by the degrees of freedom of the model, i.e., the number of independent parameters which have to be estimated. For example, if our model is defined as linear function in the input variables, the complexity can be measured by the number of input variables used for prediction. In the case of decision trees, the complexity can be defined by the number of nodes of the tree.

To understand the influence of model complexity on the test error, let us consider the regression goal and the squared loss function. Due to the fact that we do not know the class in advance, the test error consists of three components: the first component, called the *irreducible error*, is caused by the variability of the data; the second component, called the *variance*, is caused by the fact that our data is only a sample; and the third component, called *squared bias*, is caused by the choice of the model class.

### Model Complexity, Underfitting and Overfitting

- *Underfitting*: A model with low complexity can be estimated with high accuracy from the data, if we measure accuracy by the variance of the estimate. On the other hand, we have to expect that the simple model is not adequate for describing the data, i.e., we have a rather large model bias.
- *Overfitting*: A model with high complexity will be estimated with low accuracy from the data if we measure accuracy by the variance of the estimate. On the other hand, we can expect that the complex model is adequate for describing the data, i.e., we have a low model bias.



**Fig. 5.1** Bias-variance trade-off in model selection

The irreducible error cannot be influenced, but complexity of the model class influences the second and the third component and may cause effects known as *overfitting* and *underfitting*. Balancing overfitting and underfitting in such a way that the test error is minimized is of utmost importance if we want to use the model for prediction of the output variable for new values of the input variables. It is also called *bias-variance trade-off* and schematically shown in Fig. 5.1.

In the best case, the test error can be estimated from theoretical considerations, which give bounds for the generalization error. In the following sections, we will mention results in this direction, which often rely on large sample approximations and are many a time of limited use for practical applications. Hence, one is interested in the empirical evaluation of different possible models with respect to the test error. Looking at this problem from the modeling perspective in Sect. 2.1, we can understand supervised learning as an application of the modeling approach called *models of data*.

If sufficient data are at hand, a general method for model selection and model assessment is splitting the data as follows:

### Model Selection and Estimation of Test Error

1. *Data splitting*: Divide the data into a training sample, a validation sample, and a test sample. Frequently, 50 % for the training sample, 25 % for the validation sample, and 25 % for the test sample are recommended.
2. *Model learning*: Use the training data to learn the parameters of models.
3. *Model selection*: Use the validation sample to select an appropriate model.
4. *Model evaluation*: Use the test sample to estimate the test error.

Splitting in three subsamples can be simplified into splitting into two data sets by using techniques which allow the combination of the training and validation step in one single step. These techniques are based either on analytical techniques or on the reuse of the data. In such cases, a splitting rule of 70 % training sample to learn the model and 30 % test sample is frequently recommended. Moreover, in the best case, such techniques allow an estimate of the generalization error. We will explain the different techniques in connection with the models in the following sections.

## 5.2 Regression Models

After general considerations, we will discuss linear regression models as the most important standard model, introduce the basics of neural networks, and investigate two basic smoothing methods.

### 5.2.1 Model Formulation and Terminology

Linear regression models are used if we want to predict a metric response variable  $Y$ , for example, a KPI, depending on a number of predictor variables  $X = (X_1, X_2, \dots, X_p)$ . The model is defined by the equation

$$Y = f(X) + \varepsilon \quad (5.3)$$

where the deterministic component  $f(X)$  is the model for the expected behavior of the response  $Y$  and  $\varepsilon$  is a random component describing the variation of the observed data not explained by the model. As model classes for the function  $f$ , one can define the classes introduced in Sect. 2.4.4 in connection with the introduction of regression models. The standard assumption for the random component  $\varepsilon$  is a normal distribution with zero mean. With respect to the class of model functions, the most important class are linear functions in the input variables. This case is treated in Sect. 5.2.2. Alternative analytical techniques allowing more general specifications of model functions are so-called *nonparametric models*. A well-known technique are *neural nets* introduced in Sect. 5.2.3.

Another technique for nonparametric models is *nonparametric regression*. Such methods are of special interest if one wants to visualize the relationship between one input and one output by a smooth function, sometimes called *smoother*. Consequently, software for data visualization offer different smoothers for the visualization of trends in scatter plots (cf. Sect. 4.4.3). We will briefly discuss kernel estimates in Sect. 5.2.4 and smoothing splines in Sect. 5.2.5 as essential techniques, which are frequently used in other models. Generalizations to functions of more than one input variable use the concept of additive models, which define the regression function as an additive model of functions of the individual variables [13]. This

approach avoids the curse of dimensionality. Other approaches to nonparametric regression are functional models. For a comprehensive account, see [22].

The following template contains a summary of the standard procedures for regression models.

**Template: Regression Analysis**

- **Relevant Business and Data:** Customer behavior represented as cross-sectional data for process instances
- **Analytical Goals:**
  - Prediction of the response function describing the relationship between input variables and output variables
  - Prediction of output values for new input values
- **Modeling Tasks:** Define a model class, for example, linear models or nonparametric models as considered in Sect. 2.4.4
- **Analysis Tasks:**
  - *Splitting data:* Split the data randomly into one set for training and validation and one set for testing the model
  - *Model estimation:* Estimate candidate models by solving the minimization problem for the empirical risk for the training data
  - *Model assessment:* Assess the quality of the model using residual analysis
  - *Model selection:* Select the best model from the candidates using either theoretical considerations or data-oriented methods
- **Evaluation and Reporting Task:** Evaluate the selected model using the test error for the test data

The estimation of the models under consideration is usually defined as a minimization problem for the empirical risk. The estimated response function is denoted by  $\hat{f}(X)$  and the predictions for the observations are denoted by  $\hat{y}_i = \hat{f}(x_i)$ . The deviations  $r_i = y_i - \hat{y}_i$  of the observed values from the predictions are called *residuals*. The definition of the empirical risk of the estimate is, in most cases, based on the squared loss function

$$R_{\text{emp}} = \frac{1}{n} \sum L(y_i, \hat{f}(x_i)) = \sum_{i=1}^N (y_i - \hat{y}_i)^2. \quad (5.4)$$

However, other specifications are possible, for example, absolute deviations if one is interested in a robust estimation in order to reduce the influence of outliers in the data (cf. [20]).

After the estimation of a model, one has to do model assessment which evaluates the empirical risk and checks the properties of the residuals. The methods for model selection depend on the chosen model and can be based either on theoretical considerations or on the data. We will treat these methods in connection with the models.

For the evaluation of the model, the standard procedure is splitting the data into a training set and a test set. The basic quantities for evaluation are the residuals of the test data is the *mean square error* of the test data. If we use a set of  $M$  test data  $(y_i^{\text{test}}, x_i^{\text{test}})$ , the residuals of the test data are defined by  $r_i^{\text{test}} = y_i^{\text{test}} - \hat{y}_i^{\text{test}}$ , and the mean square error is defined by

$$\text{MSE}_{\text{test}} = \frac{1}{M} \sum L(y_i^{\text{test}}, \hat{f}(x_i^{\text{test}})). \quad (5.5)$$

### 5.2.2 Linear Regression

Linear regression is the most prominent predictive model for finding the value of an output variable  $Y$  in dependence on the input variables  $X$ , and it is well investigated. We sketch only few important aspects for applications. A detailed application-oriented treatment with emphasis on computational aspects is included, for example, in [7].

#### Modeling Task

Given the input variables, the model class is defined by linear functions in the input variables, i.e.,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon. \quad (5.6)$$

The interpretation of the models and the parameters is obvious in the case of quantitative input variables. The coefficients measure the effect per unit of the variable on the response. In the case of qualitative inputs (e.g., gender), the coefficients measure the effect of the categories represented by the dummy variable (cf. the description of dummy variables in Sect. 2.4.4). The selection of models within this class can be formulated as selection of relevant input variables, i.e., we understand the input variables as possible candidates for the model and select a subset by defining the coefficients for the variables not present in the model by 0.

#### Model Estimation

The estimation task is accomplished by the method of least squares; i.e., based on the observations  $(\mathbf{Y}, \mathbf{X})$  for the training data, we define the estimate of the parameters in such a way that the empirical risk, defined as squared distance of

the estimated responses  $\hat{y}_i$  from the observed responses  $y_i$ , is minimized:

$$\sum_{i=1}^N (y_i - \hat{y}_i)^2 = \min_{\beta} \sum_{i=1}^N (y_i - x_i^T \beta)^2. \quad (5.7)$$

## Model Assessment

Model assessment investigates the explanatory power of the model and checks the compliance of the model assumptions.

### Assessment of Explanatory Power

The assessment of the explanatory power is based on the decomposition of the sum of squares of the overall deviation from the mean into the explained sum of squares and the residual sum of squares (or “errors”):

$$\sum_{i=1}^N (y_i - \bar{y})^2 = \sum_{i=1}^N (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^N (y_i - \hat{y}_i)^2. \quad (5.8)$$

Here,  $\bar{y}$  denotes the overall mean of the data, corresponding to a model with no explanatory variables.

Parallel to this decomposition, we have a decomposition of the *degrees of freedom*, which is interpreted as follows: Altogether,  $N$  observations define  $N$  degrees of freedom, and the estimation of each parameter decreases the degrees of freedom by 1. Hence, for the squared deviations from the mean, we have altogether  $N - 1$  degrees of freedom; for the explained sum of squares based on  $p$  variables,  $p$  degrees of freedom; and for the residual sum of squares,  $(N - p - 1)$  degrees of freedom.

This decomposition defines two standard measures for model assessment: the first one is the *multiple R-squared*, which measures the proportion of the explained sum of squares from the total sum of squares. The second one is the *F-statistic*, which is an overall measure in how far the average explained sum of squares exceeds the average residual sum of squares. If the model assumptions hold, the well-known *F*-test can be used for testing the null hypothesis that the model gives no significant explanation for the data against the alternative hypothesis that the model gives a significant explanation for the data.

Besides these two overall measures, one can test hypotheses about the effects of input variables. For each parameter component, we test the null hypothesis that the variable has no significant contribution to the explanation of the output (i.e., the parameter has value 0) against the alternative hypothesis that the variable has a significant contribution to the model (i.e., the value of the parameter in the model differs from 0).

## Assessment of Model Assumptions

The tests about explanatory power rely essentially on the following model assumptions which guarantee a number of optimal properties of the least squares estimate:

### Assumptions for Linear Regression Models

1. *Model specification*: The mean of the function is correctly represented by the model.
2. *Independence*: The observations are independent variables.
3. *Homogeneous variances*: The variances of the error terms are the same for all observations.
4. *Normal distribution*: The error terms are normally distributed.

Hence, an important issue in all practical applications is checking the model assumptions. There exist many methods known under the term regression diagnostics which are based on the residuals  $r_i = y_i - \hat{y}_i$ .

The basic visual method for model checking is a plot of the residuals against fitted values. Under the assumptions stated above, this plot should show a symmetric scatter around 0. Any kind of visible trend in this plot is an indication that the specification of the model is incorrect. Also the inhomogeneity of variances can be detected, and diagnostics for outliers are possible.

For checking the assumption of the normal distribution of the residuals, a qq plot for the quantiles of the residuals against the quantiles of the normal distribution is useful. If the model assumptions hold, the qq plot should show approximately a straight line.

Another important issue for the quality of the model is multicollinearity which refers to a correlation of the input variables. In the case of models with highly correlated input variables, the variance of the estimates is inflated and results in low prediction power. For visualization tools detecting multicollinearity and further diagnostic methods, we refer to [8].

## Model Selection

In Sect. 5.1, we explained that in the case of unknown model specification, balancing between underfitting and overfitting is of utmost importance. A number of strategies are known for the model selection in linear regression. This avoids splitting the data into a training and a validation set. One frequently used method, which is, strictly speaking, not a method of model selection, is variable selection. Variable selection tries to find a subset of explanatory variables which have good explanatory power for the training data. In many practical problems, a complete search for finding the best selection of the input variables is not feasible due to the *curse of dimensionality*. The term curse of dimensionality refers to the computational problems that occur for high-dimensional data. For example, if we have 10 possible explanatory variables, we would have to search the best model out of  $2^{10}$  models. Hence, we have to define specific model selection strategies.

There exist various strategies for variable selection, which, to some extent, also handle the problem of multicollinearity. *Forward selection* is a strategy which builds models stepwise by starting with a model with no explanatory variable, i.e., the model is defined only by the mean. In each selection step, the variable which gives the best additional explanation to the data is added to the model. The process stops if we cannot add any variable with significant additional explanation. *Backward selection* starts with the full model and deletes variables with not significant explanatory power. *Stepwise selection* combines the two approaches by testing after the augmentation of the model with one additional variable, whether anyone of the already existing variables should be removed from the model. In connection with variable selection, it is recommended not to look at the multiple R-squared but on the *adjusted R-squared* as a measure for explanatory power which penalizes the number of parameters in the model.

More theoretically oriented methods for model selection are based on modifying the loss function by a penalization term which balances between fitting and the complexity of the model which is measured by the number of input variables. Two important criteria are the *Akaike information criterion* (AIC) and the *Bayes information criterion* (BIC) defined by

$$\begin{aligned} \text{AIC} &= -2 \sum_{i=1}^N (y_i - \hat{y}_i)^2 + \frac{d}{N} \\ \text{BIC} &= -2 \sum_{i=1}^N (y_i - \hat{y}_i)^2 + d \cdot \log N. \end{aligned} \quad (5.9)$$

The idea behind AIC is measuring the information loss of a model compared to the “true” model. If we have a number of models at hand, the best choice would be a model which minimizes information loss. In the case of normally distributed errors, an equivalent criterion to AIC is known as Mallows  $C_p$ . BIC starts from a prior probability for all models and calculates the posterior probability for the models. We should select the model with the highest posterior probability. A detailed treatment is included in [3].

The above-mentioned strategies for model selection define a model with a definite number of parameters and are available in many statistical software packages as options in regression procedures. Besides these methods, so-called *shrinkage methods* can be used. Shrinkage methods are of special interest in case of a large number of input variables caused by the definition of dummy variables for qualitative input variables (cf. Sect. 2.4.4). In the last years, the *Lasso* became a popular shrinkage method which can be interpreted as a method for continuous variable selection. Instead of minimizing the sum of squares, the Lasso minimizes the penalized loss function

$$\hat{\beta} = \arg \min_{\beta} \sum_1^N (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^p |\beta_j|. \quad (5.10)$$

The parameter  $\lambda$  can be interpreted as complexity parameter. In case of  $\lambda = 0$ , the solution corresponds to the least squares estimate; in case of  $\lambda = \infty$ , the regression function is estimated by the mean. Another shrinkage method is *ridge regression*, which is probably the oldest shrinkage method. Ridge regression uses the squares of the coefficients for penalization. Details of the Lasso and ridge regression can be found in [14].

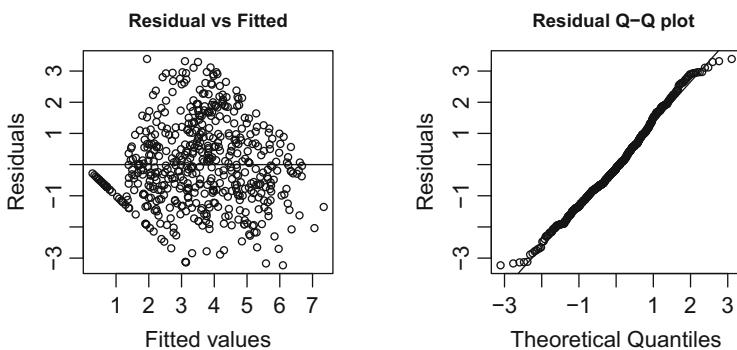
We conclude this section with application of regression in context of the CRM use case.

#### CRM Use Case: Prediction of Sales

Suppose we want to predict the sales of customers from previous sales measured by the sales index `Average_Sales`, duration of the relationship to the enterprise `Duration` and usage intensity of the different services measured by an activity index `Intensity`. As we have already seen in Chap. 4, sales seem to be significantly correlated with the sales index, whereas duration seems to be less important. The results of the model using these variables were nonsatisfying, and we tried to fit a model using the transformed variables  $\sqrt{\text{Sales}}$ ,  $\sqrt{\text{Average}_\text{Sales}}$ , `Duration`, and `Intensity`. Using variable selection with AIC dropped `Duration`. The fit of the model is shown in Fig. 5.2. The result is not completely satisfactory, mainly due to the fact that some customers actually show no sales.

Details of the results and the predictive power can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)



**Fig. 5.2** Residual plot and qq plot for sales model (R graphics)

### 5.2.3 Neural Networks

Neural networks are a modeling technique for the approximation of functions having its origin in the perceptron, one of the first models for learning behavior of the human brain. The model is of interest if one has to find models for problems with limited knowledge about the relation between input and output variables. A comprehensive overview about the application of neural nets for learning may be found in [15].

#### Modeling Task

Various network topologies and structures have been proposed for different applications, but we consider only the standard backpropagation network with one hidden layer for the approximation of a function  $Y = f(X)$ . The model is defined by a layered graph with three layers. The first layer is called *input layer* with a number of nodes corresponding to the number of input variables. Usually, a so-called *bias node* is included for modeling the intercept. The middle layer is called *hidden layer* with an arbitrary number of nodes, and the third layer is the *output layer* with a single node representing the response variable. The edges are defined between two adjacent layers bearing weights. Referring to the origin of the model, the nodes of the graph are called *neurons* and the connections between the nodes *synapses*.

#### CRM Use Case: Prediction of Sales

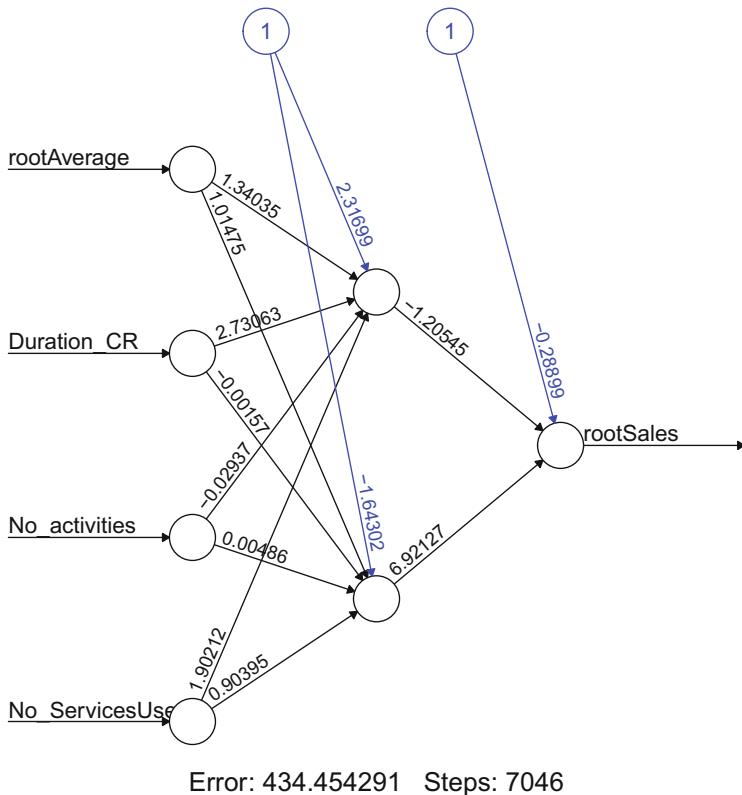
Figure 5.3 shows a network with three input nodes, two nodes in a hidden layer, and one output layer for modeling the relationship between the output  $\sqrt{\text{Sales}}$  and the input variables `rootAverage`, `Duration_CR`, `No_activities`, and `No_ServicesUse`. Additionally, the figure shows the bias nodes for modeling the intercepts in the input layer and the hidden layer.

The model works as follows: we take the input values of the variables and propagate the input to the nodes in the hidden layer by using an activation function  $f$  applied to the linear combinations of the inputs for the two nodes in the hidden layer:

$$\begin{aligned} z_i &= g_1(w_{0i} + w_{i1}\text{rootAverage} + w_{i2}\text{Duration\_CR} + w_{i3}\text{No\_activities} \\ &\quad + w_{i4}\text{No\_ServicesUse}) \end{aligned} \tag{5.11}$$

Next, the values  $(z_1, z_2)$  are used for calculation of the output layer by using an output function  $g$ :

$$\sqrt{\text{Sales}} = g_2(v_0 + v_1z_1 + v_2z_2). \tag{5.12}$$



**Fig. 5.3** Neural net for prediction of sales (R package *neuralnet*)

For the function  $g_1$ , the most popular choice is the *sigmoid function*  $f(t) = 1/(1 + \exp(-t))$ . For  $g_2$ , in case of regression, the identity function is used. If one wants to model an output with values in  $[0, 1]$ , for example, the prediction of probabilities, the sigmoid function is used.

### Model Estimation

For a given number of nodes in the hidden layer, the weights are the parameters which are determined in such a way that the empirical risk of the estimated values for the training data is minimized. With respect to the weights, a local minimum can be found as the zeros of the partial derivatives of the empirical risk by using classical calculus. This problem is solved by the method of the steepest descent, which means that we start from the initial weights and change the weights in the direction given by the negative derivatives. The algorithmic solution uses the following interpretation of the derivatives given by the structure of the problem: The derivatives with respect to the weights  $v$  in the second layer can be interpreted as errors caused by the

output layer and the derivatives with respect to weights  $w$  in the first layer as errors at the hidden layer depending on the errors at the output layer. This structure allows the realization of the steepest gradient algorithm by the *backpropagation* algorithm:

---

**Algorithm 1:** Backpropagation
 

---

```

1 Initialization: Start with initial weights, usually defined by a random values;
2 repeat
3   Forward pass: Calculate the predictions for the outputs;
4   Backward pass: Calculate the errors;
5   begin
6     Calculate errors at the output layer and derivatives with respect to
      weights  $v$ ;
7     Calculate errors at the hidden layer using errors at the output layer and
      derivatives with respect to weights  $w$ ;
8   end
9   Adapt weights in negative direction of the derivatives using a step-size
      parameter;
9 until convergence is reached;
```

---

## Model Assessment

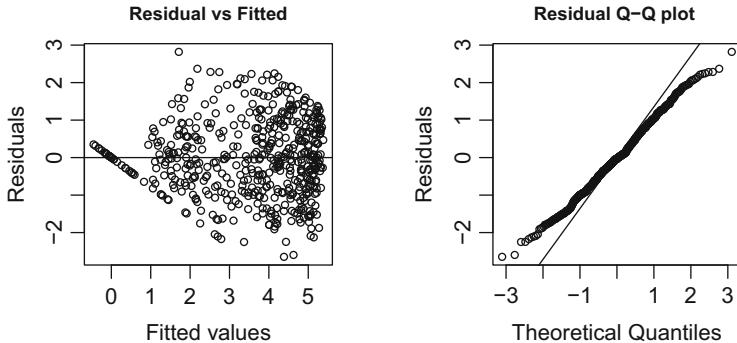
For model assessment, we use the residual sum of squares. For model checking, the basic methods are, as in case of linear regression, a plot of the residuals against the fitted values and a qq plot for checking the distribution of the residuals.

## Model Selection and Model Evaluation

From a computational point of view, the configuration of the network is a major task. A well-known result in the theory of neural networks is that one hidden layer is sufficient [16]. For the number of nodes in the hidden layer, the experimentation with different numbers of hidden nodes ranging from 2 up to 100 is proposed. This number depends on the available amount of data. A formal method for avoiding overfitting is *weight decay*, which, similar to ridge regression, uses a penalization of the weight size. For measuring the importance of the different input variables, one can use the concept of *generalized weights*, which measures the importance of an input variable for the estimated output (cf. [10]). One can also use AIC for a comparison between models.

For model evaluation, the standard procedure is splitting the data into a training set and a test set and evaluating the model using the mean square error of the test data.

Let us also mention that in the case of no hidden layer and the identity function as activation function, the neural net corresponds to linear regression. In the case of a sigmoid activation function, a network with no hidden layer corresponds to logistic regression (see Sect. 5.3.2).



**Fig. 5.4** Residuals for neural net (R graphics)

#### CRM Use Case: Prediction of Sales

We fitted a neural net for the prediction of the sales data. The resulting net is shown in Fig. 5.3. Looking at the residuals in Fig. 5.4, we see that the residuals have a similar structure as in the regression model; however, the normal assumptions for the residuals do not hold.

Details of the calculation can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

### 5.2.4 Kernel Estimates

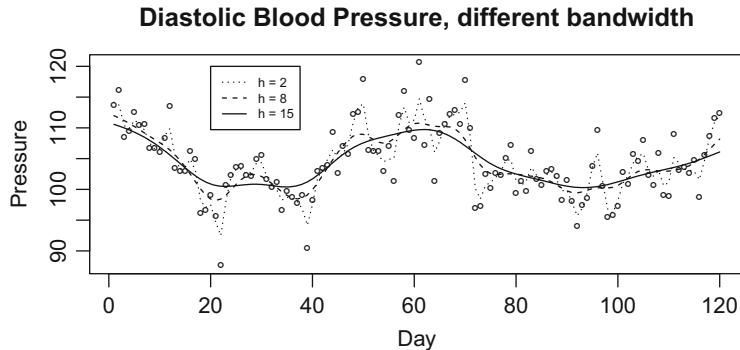
Kernel smoothing defines a model for the relation between the response and one real-valued input variable as a locally weighted mean of neighboring points. The formula of the estimate is given by

$$\hat{f}(x) = \frac{\sum w_j y_j}{\sum w_j} \quad w_j = \frac{1}{h} K\left(\frac{x - x_j}{h}\right). \quad (5.13)$$

The function  $K(x)$  is called *kernel function* which has the properties of a probability density for a distribution with mean 0 and finite variance. One frequently used specification is the normal kernel defined by the normal probability density.

The parameter  $h$  is called *bandwidth* which defines the trade-off between smoothness and fit. It can be interpreted as the complexity parameter of the model, which controls the fit to the data by the model. Small values of  $h$  will result in a function following close to the data, whereas large values of  $h$  will give smooth functions, in extreme cases, the mean.

This model explicitly defines the estimation function and does not use a loss function, but the background of the definition uses a quadratic loss function. For



**Fig. 5.5** Kernel estimates for blood pressure with different bandwidths (R graphics)

a more comprehensive introduction to the topic, see [1], where one may also find formulas for the calculation of confidence bands for smoothers.

Let us show the application in an example. Note that this example is, strictly speaking, not an example of cross-sectional data but an example of temporal data. This points to the fact that methods for regression can be applied to data of process instances in the state view.

#### Preeclampsia Use Case: Prediction of Blood Pressure

A person's blood pressure often shows high volatility. Thus, a representation of its course over time is of interest. Figure 5.5 shows kernel estimates with different values of bandwidth  $h$  for 120 daily measurements of one person's blood pressure. Note that due to the distances between the observations, we have chosen a rather large value for the bandwidth. A bandwidth  $h \leq 1$  would almost follow the data points.

Details of the results and the predictive power can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

## Model Selection

The determination of an optimal bandwidth in the sense of the predictive power of the model can be obtained by the method of *cross-validation*. Cross-validation calculates the prediction error of an observation  $y_i$  which predicts the value by a model  $\hat{f}_{(i)}(X)$  using all data points except  $(x_i, y_i)$ . Obviously, in the case of kernel estimates, this prediction error depends on the choice of the bandwidth  $h$ . The optimal parameter  $h$  is defined by the minimization of the cross-validation prediction error:

$$\hat{h} = \arg \min_h \sum_{i=1}^N (y_i - \hat{f}_{(i)}(x_i))^2. \quad (5.14)$$

It should be mentioned that in some cases, this method for bandwidth determination can produce solutions which are at the boundary of the interval for the bandwidth. In particular, if the data show outliers or if the number of observations is rather small (say less than 100), the method recommends a small bandwidth. This means that the estimate follows close to the data.

### 5.2.5 Smoothing Splines

#### Modeling Task

Smoothing splines start from the idea that the relation  $Y = f(X)$  between two real variables  $X$  and  $Y$  should be a smooth function. Smoothness of a function  $f$  is measured by the second derivative of a function: a function with absolute small second derivative at a peak has a low curvature at the peak, whereas a function with absolute large second derivative has a high curvature at a peak. More precisely, the estimate  $\hat{f}$  is defined as the solution of the following minimization problem:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda \int ((f''(x))^2 dx. \quad (5.15)$$

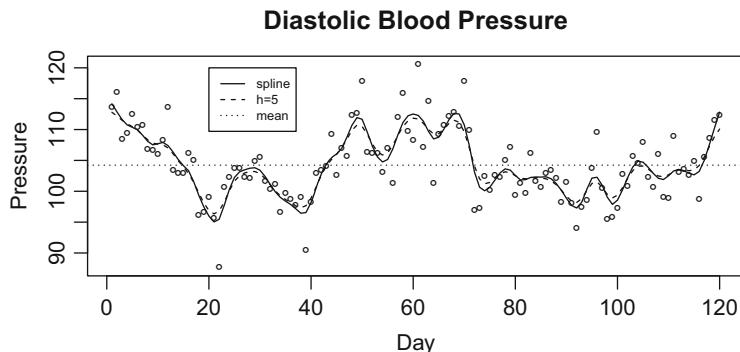
The interpretation of the equation is that the first term measures the fit of the function to the data, and the second term measures the degree of smoothness. The parameter  $\lambda$  gives the importance of the smoothness and has to be chosen in an appropriate way. If  $\lambda$  is close to 0, we do not care much about smoothness and would approximate the function by a polygon through the data points. If  $\lambda$  is rather large, we approximate the data by a linear regression function, which has second derivative 0. In that sense, we can interpret the choice of  $\lambda$  as a method for controlling model complexity as explained in Sect. 5.1.

#### Model Estimation

It can be shown that, independent of the value of  $\lambda$ , the solution of the minimization problem is a cubic spline with knots defined by all data points. A cubic spline over the interval  $[a, b]$  with  $K$  knots,  $A = a_1 \leq a_2 \leq \dots \leq a_K = b$ , is defined as a piecewise polynomial of degree 3 in each interval  $[a_i, a_{i+1}]$ . These polynomials are connected at the nodes in a smooth way which means that the values and the first and the second derivatives of the polynomials on both sides of the knot are identical. Details may be found in [5]. The choice of the parameter  $\lambda$  corresponds to the idea of model selection in regression.

#### Model Selection

As in the case of kernel estimation, the standard method for choosing  $\lambda$  is cross-validation. For a given  $\lambda$ , we calculate for each observation the prediction error of a



**Fig. 5.6** Comparison of smoothing spline and kernel estimate (R graphics)

model  $\hat{f}_{(i)}(X)$  estimated from all data points except  $(x_i, y_i)$ . The optimal parameter  $\lambda$  is defined by the minimization of the cross-validation criterion:

$$\hat{\lambda} = \arg \min_{\lambda} \sum_{i=1}^N (y_i - \hat{f}_{(i)}(x_i))^2. \quad (5.16)$$

**Preeclampsia Use Case: Prediction of Blood Pressure**  
 We applied spline smoothing for the data of diastolic blood pressure and obtained an optimal smoothing parameter  $\lambda = 1.3e - 6$ . Choosing an optimal bandwidth for the kernel estimates did not work. Figure 5.6 shows a smoothing spline for a certain person and a smoothing spline with bandwidth  $h = 5$ . As one can see, the results are rather comparable. From the graphics, we conclude that the person has a rather constant blood pressure with occasional peaks that are kept in the model. These peaks explain to some extent that the automatic choice of an optimal bandwidth does not work.

Details of the results and the predictive power can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

### 5.2.6 Summary: Regression Models

Regression models are used for the prediction of the value of a response variable in dependence of a number of explanatory variables. After an assessment of the data by descriptive measures and by visualization, the first step in the analysis is the choice of an appropriate model class. Three different model classes were considered in this section: linear regression models, neural network models, and

nonparametric regression. Using the training data, the possible candidate models are estimated. The criterion for estimation is the minimization of the empirical risk. After estimation, the model assessment has to be done. The most important input for model assessment are the residuals. The selection of a final model out of a number of candidate models is done by different methods. In the case of linear regression, a number of methods are available. From a theoretical point of view, the utilization of information criteria is advisable. In the case of neural networks, the ideas used in linear regression have to be modified and are of more heuristic character. An alternative could be splitting the training data in a training set and a validation set. In the case of nonparametric regression, the method of cross-validation can be used. After the selection of the final model, the predictive power of the model has to be checked by evaluation of the empirical risk for the test data.

## 5.3 Classification Models

After general considerations on classification methods, this section describes a number of frequently used techniques such as Bayes classifiers, logistic regression, tree-based methods, nearest-neighbor classification, support vector machines, and boosting.

### 5.3.1 Model Formulation and Terminology

For classification models, the data structure is the same as in regression, but the response variable has only a finite number of values  $Y \in \{g_1, g_2, \dots, g_K\}$  defining the classes. Here, we want to learn a rule how the class membership of an observation can be predicted using the explanatory variables  $X$ .

The analytical goal in classification can be formulated in two different ways:

#### Analytical Goals in Classification

1. *Class assignment*: Estimate the output value for the data by a function of the inputs  $Y = f(X)$ .
2. *Probability for classes*: Estimate the probability distribution of class membership for an observation given the input variables  $x$ :  $p_g(x) = P(Y = g|x) = p_g(x)$ .

In the case of the second formulation, the assignment to a class is usually done by assigning the observation to the most likely class.

There are different model classes which can be used in classification, which are discussed in the following subsections. We will use probabilistic models, trees, models based on distances, support vector machines, and models based on combination. Probabilistic models use the second formulation of the analytical goals; the other models use the first formulation.

For the determination of the empirical risk, two different loss functions can be used. The first one is the indicator function or 0 – 1-loss:

$$L(y, \hat{y}) = \begin{cases} 0, & \text{if } y = \hat{y} \\ 1, & \text{otherwise,} \end{cases} \quad (5.17)$$

and the second one is the *loglikelihood function*, which is also called *cross entropy* or *deviance*:

$$L(y, \hat{f}(X)) = -2 \log(\hat{p}(X)). \quad (5.18)$$

The advantage of 0 – 1-loss is that generalizations to different costs for different misclassification can be done easily by replacing 1 by a weight  $w_{k\ell}$  which defines the cost if data from true class  $k$  are assigned to class  $\ell$ . Such considerations are of interest in many applications, in particular in medicine, but we will not go into details. All software packages for classification have options for allowing such weighting.

The representation of the empirical risk for 0 – 1-loss is done by the so-called *confusion matrix* which summarizes the correct and incorrect counts for a certain data set. The rows of the matrix correspond to the actual classes of the data and the columns to the predicted classes:

$$C = \begin{pmatrix} g_1 & g_2 & \dots & g_K \\ \hat{g}_1 & n_{11} & n_{12} & \dots & n_{1K} \\ \hat{g}_2 & n_{21} & n_{22} & \dots & n_{2K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{g}_K & n_{K1} & n_{K2} & \dots & n_{KK} \end{pmatrix}$$

A correct classification rate corresponds to the percentage of the values in the diagonal of the matrix. In case of more general loss functions, this matrix can be used as basic information about the classification, but in order to find the loss, the values have to be multiplied with the weights for the different misclassification costs.

For the computation of the decision function, one usually starts with a classification problem with two classes. In this case, it is convenient to characterize the classes internally either by the class labels  $g = 0$  and  $g = 1$  or by  $g = -1$  and  $g = 1$ . Some methods allow immediate generalization from two to more than two

**Table 5.1** Terminology in connection with binary classification problems

Prediction	Actual class		
	Positive	Negative	
Positive	<b>True positive (TP)</b>	<i>False positive (FP)</i>	<i>Precision = TP/(TP+FP)</i>
Negative	<i>False negative (FN)</i>	<b>True negative (TN)</b>	<i>Negative predicted value = TN/(TN+FN)</i>
	<i>Sensitivity = TP/(TP+FN)</i>	<i>Specificity = TN/(FP+TN)</i>	

Other terms used:

Precision = Positive predictive value

Recall = Sensitivity

False positive rate (false alarm) = 1 – Specificity

False discovery rate = 1 – Precision

Accuracy =  $(TP + TN)/(TP + TN + FP + FN)$

classes, other methods use one of the following strategies for solving problems with more than two classes:

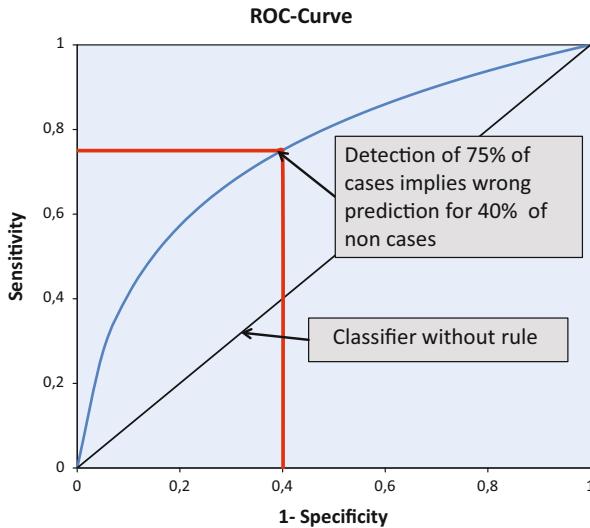
### Extending Classification to an Arbitrary Number of Classes

- *One versus the rest*: Perform for each class a classification versus all other classes, and classify the observation to the class with the highest probability in the  $K$  classifications.
- *Classification of all pairs*: Perform  $\binom{K}{2}$  classifications of all pairs of classes, rank the classes according to the number of assignments, and choose the class with the highest rank.

For model assessment and for the overall evaluation of a classification method, a number of criteria based on the classification matrix are used. Table 5.1 summarizes criteria and terminology in use for the case of two classes. This terminology makes also clear that weighting the different kinds of misclassification is a problem of interpretation of the importance of errors in the domain.

In the case of two classes, the *ROC curve* is another useful tool for assessing the quality of classification. Figure 5.7 shows the typical shape of an ROC curve.

The idea behind the ROC curve is to evaluate a classification method for two classes in dependence of the trade-off between the two types of misclassification. Instead of making the class assignment according to the group with  $g = 1$ , if  $P(1|x) \geq 0.5$ , we can use another threshold  $t \neq 0.5$ . This threshold parameter can be interpreted as a different cost for misclassification for the two classes. If we vary this threshold and draw a curve of the sensitivity against 1-specificity of the classification result, we obtain the ROC curve. The straight line corresponds to the classification without any method. The overall quality of the method can be measured by the area under the curve. In the example depicted in Fig. 5.7, the area under the curve is  $a = 0.73$ .



**Fig. 5.7** ROC curve for classification of two classes

As already explained in Sect. 5.1, an important issue in classification is model choice in order to avoid overfitting to training data. If sufficient data are available splitting into subsets for training and validation is the best method. In cases of small data sets, a standard method for model selection is  $k$ -fold cross-validation, which generalizes the idea of cross-validation introduced in Sect. 5.2.4.

---

#### Algorithm 2: $k$ -Fold cross-validation

---

```

1 begin
2   Divide the training data into  $k$  disjoint samples of roughly equal size;
3   For each validation sample use the remaining data to construct the
   estimate and estimate the empirical risk for the left out data;
4   Compute the prediction error by averaging the empirical risk of the
   validation data;
5 end

```

---

A frequently used value for  $k$  is  $k = 10$ . For many classification methods, software packages offer options for integrating cross-validation into the calculations of the classification function.

A summary of the different steps in the analysis of classification problems is included in the following template.

**Template: Classification**

- **Relevant Business and Data:** Customer behavior represented as cross-sectional data for process instances with a matrix ( $\mathbf{Y}, \mathbf{X}$ )
- **Analytical Goals:**
  - Determination of a function for group membership for the observation
  - Determination of the probability of class membership for the observations given the input values
- **Modeling Tasks:** Definition of a model class for the classification function (cf. Sects. 5.3.2–5.3.6)
- **Analysis Tasks:**
  - *Splitting Data:* Split the data randomly into one set for training and validation, and another set for testing the model
  - *Model Estimation:* Estimate candidate models
  - *Model Assessment:* Assess the quality of the model using the confusion matrix and, if possible, the ROC-curve
  - *Model Selection:* Select the best model from the candidates using either k-fold cross-validation or splitting training data into a training set and a validation set
- **Evaluation and Reporting Task:** Evaluate the selected model using the test error for the test data based on the confusion matrix

In the following sections, we will present different methods. In the description, we will use only the model class, algorithms for finding the classifier, and methods for model selection. Evaluation will be discussed in Sect. 5.3.7.

### **5.3.2 Classification Based on Probabilistic Structures**

We will consider two different methods for solving analytically the classification problem: the Bayes approach and logistic regression.

#### **The Bayes Approach**

This approach defines a model for the probability distributions in the classes and formulates the classification problem in the context of Bayes decision theory. If we have two classes labeled by  $Y = 0$  and  $Y = 1$ , we assume that the explanatory variables have common distributions in both classes characterized by the densities  $p(x|0)$  and  $p(x|1)$ . Moreover, we know the prior probabilities of the two classes  $\pi_0 = P(Y = 0)$  and  $\pi_1 = P(Y = 1)$ . Given these probabilities, we can define the joint probability of the class and the input variables  $p(x, g) = p(x|g)\pi(g)$

and calculate the posterior probability of a class given the input variables using the Bayes theorem (cf. Sect. 2.4.2):

$$P(Y = g|x) = \frac{p(x|g)\pi_g}{p(x)} \quad g = 0, 1. \quad (5.19)$$

The decision about the class for a new attribute vector  $x_{\text{new}}$  is given by maximizing the posterior class probability:

$$\hat{y} = \begin{cases} 0, & \text{if } \frac{P(Y=0|x_{\text{new}})}{P(Y=1|x_{\text{new}})} > 1; \\ 1, & \text{if } \frac{P(Y=0|x_{\text{new}})}{P(Y=1|x_{\text{new}})} < 1 \end{cases}. \quad (5.20)$$

It can be shown that this decision is optimal with respect to the risk defined by the misclassification rate. Note that this rule automatically provides the probability of the classes for the vector of explanatory variables. Provision for different costs of misclassification can be included easily by changing the threshold for deciding for group 1 in the decision rule in (5.20). An application for more than two classes is also straightforward by an assignment of an observation to the class with the highest posterior probability.

The crucial point in the application of this rule is the knowledge of the probability densities and the prior probabilities. For the prior probabilities, a standard approach is using the relative frequency of the classes in the training sample. For the probabilities of the explanatory variables, direct estimation from the training data is only feasible if we assume a specific parametric model for the joint densities of all input variables. This approach can be used in the case of normally distributed input variables with the same covariance structure in both groups. It leads to the well-known linear discriminant analysis, which historically stands at the beginning of classification (cf. [14]).

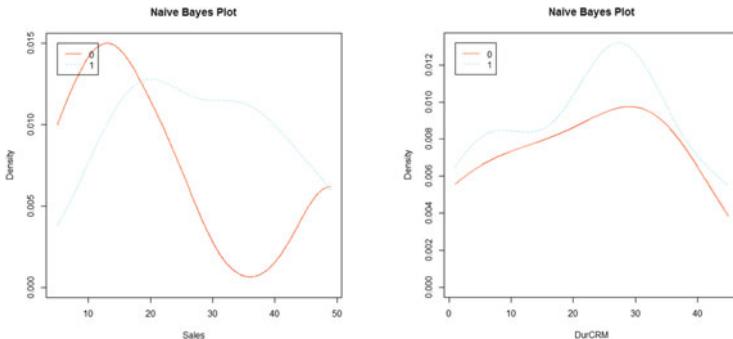
In most practical applications, this approach is not feasible, because the explanatory variables are a mixture of qualitative and quantitative variables. Further, the estimation of the joint probabilities runs into the curse of dimensionality. A standard approach to overcome these problems is the *naive Bayes* method.

Naive Bayes assumes that the input variables are all independent. In this case, the probability distribution is estimated separately for each input variable, and the joint probability is obtained by multiplying all components. The empirical frequencies of the different categories can be used for estimation of qualitative explanatory variables. For quantitative variables, one can use either a parametric model for each variable or density estimation.

Let us show how the approach works for a simple artificial example in the context of a question about customer relationship management (CRM).

*Example 5.1 (Customer Behavior in Service Usage)* The table in Fig. 5.8 shows the data of 11 customers in the shaded area with respect to the usage of a specific Service together with the amount of Sales, the Duration of customer relationship, and the Type of customer (professional or private).

CR-Dur	Sales	User Type	UseService	$P(\text{no} \mathbf{x})$	$P(\text{yes} \mathbf{x})$	Decision
10	12	private	yes	0.5004	0.4996	no*
24	36	business	yes	0.0865	0.9135	yes
28	48	business	yes	0.5999	0.4001	no*
45	20	private	yes	0.3121	0.6879	yes
30	34	private	yes	0.0423	0.9577	yes
3	21	private	yes	0.3337	0.6663	yes
1	5	business	no	0.8300	0.1700	no
23	23	business	no	0.5414	0.4586	no
12	49	business	no	0.6672	0.3328	no
35	12	private	no	0.5080	0.4920	no
33	15	private	no	0.4389	0.5611	yes*
12	25	private	??	0.2804	0.7196	yes



**Fig. 5.8** Naive Bayes approach for classification (R.graphics)

Our goal is the prediction of the behavior of the customer in the last line. A possible solution is using a nonparametric density estimate for the two quantitative variables and the relative frequency approach for the customer type. Thus, we obtain the posterior probabilities for the classes and the decision about class assignment. As we can see from the training data, eight cases are correctly classified and three are misclassified. The density estimates for Sales and Duration are shown at the bottom of the figure. Keeping the small number of data in mind, the estimates of the densities are by no means accurate, but the assumption of a normal distribution seems not very realistic in this case.

The naive Bayes approach has been applied successfully in many problems, for example, spam detection which is a simple example of text mining. Note that the approach assumes that the true probability model is within the family of distributions. Model selection is done by the selection of the variables used as predictors for the classes. In general, there is no formal method for model selection as in regression, and we have to use different sets of explanatory variables. Splitting into training and test set is always necessary. A formal analysis of naive Bayes can be found in [11].

## Logistic Regression

Logistic regression is a method for modeling the probability of the class of interest  $p = P(Y = 1)$  in dependence of a number of explanatory variables. In the case of independent and identically distributed observations of the process instances, the number of instances falling into the class  $\{Y = 1\}$  follows a binomial distribution with parameter  $p$ . Instead of modeling the dependence of the probability  $p$  from the explanatory variables directly, we transform the probabilities into *odds* defined by  $\text{odds} = P(Y = 1)/P(Y = 0) = p/(1 - p)$ , which are an alternative for defining probabilities in the context of betting (cf. Sect. 2.4.2). Given the odds, one can immediately calculate probabilities by the formula  $p = \text{odds}/(1 + \text{odds})$ . Next, we transform the odds into the so-called *logits* by  $\text{logit} = \ln(\text{odds})$  and define a linear model for the logits by using the explanatory variables:

$$\text{logit} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p. \quad (5.21)$$

The estimation of the parameters  $\beta = (\beta_0, \beta_1, \dots, \beta_p)$  is done by the method of maximum likelihood, i.e., we choose the parameters in such a way that the likelihood of the data defined by the binomial distribution is maximized. The numerical calculation and assessment of the estimates is accomplished within the framework of generalized linear models. These models transfer the ideas of linear regression to models where a linear model in the explanatory variables is defined for a function of the mean. This function is called *link function*. The overall assessment of the model is based on the deviance as loss function. As in the case of linear regression, a number of diagnostic tools are available for assessing the model. For further details, see [17]. Moreover, techniques for model selection, that are similar to those described for regression in Sect. 5.2.2 can be used.

Predictions for the logits can be transformed into probabilities by using the formula

$$p(X) = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p)} \quad (5.22)$$

and the classification rule for new explanatory variables  $x_{\text{new}}$  is given by

$$\hat{y} = \begin{cases} 1, & \text{if } p(x_{\text{new}}) \geq \text{tr} \\ 0, & \text{if } p(x_{\text{new}}) < \text{tr} \end{cases}. \quad (5.23)$$

The standard choice for the threshold  $\text{tr}$  is  $\text{tr} = 0.5$ , but other values can be used in the case of different costs of misclassification for the groups. For an overall assessment of the classification, the ROC curve is used. The generalization to more than two classes is usually achieved by one against the rest, the first method described in Sect. 5.3.1.

**Table 5.2** Demonstration example for logistic regression

Duration	ActInd	UserType	Quit	Duration	ActInd	UserType	Quit
5.63	1.93	office(0)	no (0)	6.43	7.6	office(0)	yes(1)
6.39	9.47	office(0)	no (0)	5.55	3.53	private(1)	yes(1)
5.31	9.23	office(0)	no (0)	6.68	3.6	private(1)	yes(1)
5.76	11.67	office(0)	no (0)	3.35	0.23	private(1)	yes(1)
7.12	8.9	office(0)	no (0)	4.31	0.53	private(1)	yes(1)
8.13	9.9	office(0)	no (0)	2.06	2.33	private(1)	yes(1)
4.1	7.27	office(0)	no (0)	3.03	2.5	private(1)	yes(1)
4.29	10.8	office(0)	no (0)	4.78	5.37	private(1)	yes(1)
1.55	4.97	office(0)	no (0)	5.89	1.13	private(1)	yes(1)
0.81	7.2	office(0)	no (0)	4.78	3.83	private(1)	yes(1)
5.25	9.0	private(1)	no (0)	3.83	1.47	private(1)	yes(1)
4.26	8.57	private(1)	no (0)	1.25	2.87	private(1)	yes(1)

A strong point of logistic regression is the interpretation of the parameters  $\beta$  in the model. In the case of a dichotomous input variable,  $\exp(\beta)$  measures the change of the odds if the corresponding input variable has the value 1 compared to the odds if the value is 0. This can be interpreted as the increase (or decrease) of the risk for the event of interest if the event defined by  $X = 1$  occurs. In the case of qualitative variables defining more than two categories, the interpretation is the change of the risk for a category compared to a reference category. For quantitative input variables,  $\exp(\beta)$  measures the change of the odds if the corresponding input variable changes by one unit.

Let us demonstrate logistic regression by a simple example for churn management.

*Example 5.2 (Churn Management)* Suppose we have 24 observations of customers from which 12 quit their relationship to the company in the last year and 12 are still customers. In addition, we know the customers' UserType (private user or office user), activity index ActInd, and the Duration of the customer relationship. The data are shown in Table 5.2.

Logistic regression for the churn rate showed that Duration does not have a significant influence, and the following model was estimated:

$$\text{logit}(Quit) = 1.385 + 3.058\text{UserType} - 0.577\text{ActInd} \quad (5.24)$$

The coefficient has the following interpretation: The risk for churning for private users is  $\exp(3.058) = 21.3$  times the risk of office users. An increase in activities by one unit reduces the risk of churning by a factor  $\exp(-0.577) = 0.56$ .

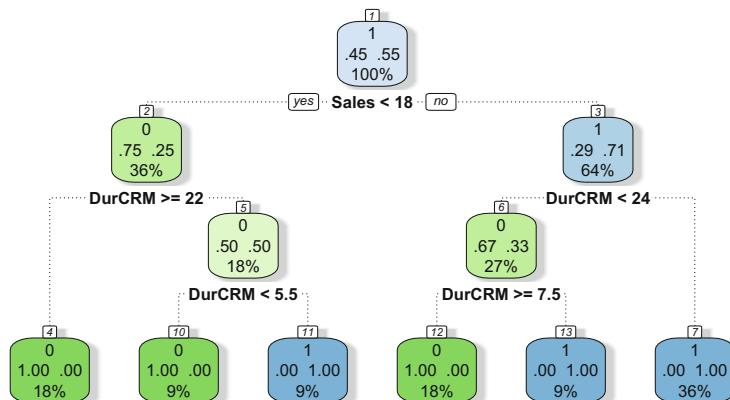
### 5.3.3 Methods Using Trees

The basic idea of tree classification resembles the strategy frequently used in guessing games for terms. By asking a sequence of questions that are answered with yes or no, the candidate tries to limit the number of possible terms until he/she can make a guess with high confidentiality. In combination with this strategy for finding the class membership of the training data, we use a binary tree for modeling purposes. All cases of the training data belong to the root node. In each node of the tree, the data belonging to that node is split into subsets according to the values of one input variable.

*Example 5.3 (Customer Behavior in Service Usage)* Figure 5.9 shows such a partition for the training data given in Fig. 5.8 for the demonstration of the naive Bayes method.

The root node represents all 11 cases. Using the variable `Sales`, the data are split into cases 1, 7, 10, 11 with `Sales < 18` going to the left node and the other cases going to the right node. The data at the left node are split further according to the values of `DurationCRM`. Cases 10 and 11 have a value `DurationCRM > 22` and go to the right, cases 1 and 7 to the left. At the right node, no further split is necessary, because both cases do not use the service and the terminal node is labeled with 0 indicating `UseService = no`. At the left node, a further split is done according to the value of the variable `DurationCRM` for separating cases 1 and 7. In a similar way, the other cases are partitioned at the right node, and we obtain a perfect classification for the training data.

The classification of the new case in the table in Fig. 5.8 is retrieved according to the decision tree. Because `Sales = 25 > 18`, we first go to the right; afterwards, `DurationCRM = 12` indicates to go the left and subsequently to go to the right. Hence, we decide that the customer wants to use the service.



**Fig. 5.9** Classification tree for simple example (R package rattle)

In this simple example, one can obtain a decision tree probably by experimenting. Defining such a model for realistic training data requires the formal definition of two strategies:

### Strategies for the Definition of a Tree Model

1. *Splitting rules*: A strategy for growing the tree defining in each node which variable should be used for splitting together with the threshold for the split.
2. *Pruning rules*: A strategy for pruning the tree in order to avoid overfitting to the training data.

Note that the strategy for defining the model is also the algorithm for defining the classification rule. The most frequently used algorithm for both strategies is *CART* which is the acronym for *classification and regression trees*. The name indicates that the model can also be applied to regression problems. The first systematic treatment of the method was given in [2], and we will briefly discuss the ideas behind splitting and pruning.

Splitting is based on measures for the *impurity* of a node. A node has impurity 0 if only input data of one class belong to the node. If  $\hat{p}(g|t)$  denotes the relative frequency of class  $g$  at node  $t$ , the impurity  $Q(t)$  of the node is defined by one of the following two measures:

$$Q(t) = \begin{cases} \sum \hat{p}(i|t) \hat{p}(j|t) = 1 - \sum (\hat{p}(j|t))^2 & \text{Gini Index} \\ Q(t) = -\sum \hat{p}(j|t) \ln(\hat{p}(j|t)) & \text{Entropy} \end{cases}. \quad (5.25)$$

For the split, the variable is used, which minimizes the impurity in the child nodes. Finding this variable can be done by greedy search. In the case of a metric variable, the split is defined by a decision rule in the form ( $X < \text{tr}$  OR  $X \geq \text{tr}$ ); in the case of nominal variables, we decide according to the rule ( $X = a_t$  OR  $X \neq a_t$ ). The process continues as long as the child nodes contain different classes or the number of cases in the node is too small. In the example tree in Fig. 5.9, the entropy measure was used.

After growing, the tree is pruned for obtaining a simpler decision rule which avoids overfitting. The basic strategy for pruning is to remove splits which lead only to a small improvement of the empirical risk. CART uses a more complex strategy, which allows the pruning of sub-trees by using a penalized risk instead of the empirical risk:

$$R_{\text{pen}}(\alpha) = R_{\text{emp}} + \alpha|T|. \quad (5.26)$$

Here,  $|T|$  denotes the number of nodes in the tree measuring the complexity of the model and  $\alpha$  is a penalization parameter for complexity.  $\alpha = 0$  corresponds to the most complex tree after complete growth, and  $\alpha = \infty$  corresponds to a simple model with only a root node. It can be shown that there is a finite number of penalization parameters  $\alpha_0 = 0, \alpha_1, \dots, \alpha_M$  such that for all  $\alpha \in (\alpha_i, \alpha_{i+1}]$ ,

there exists one unique tree with a minimum number of nodes which minimizes the penalized risk  $R_{\text{pen}}(\alpha)$ . This fact can be used for the efficient computation of a number of candidate tree models with minimum number of nodes from the training data. For selection of the final model, a test set is used or k-fold cross-validation is applied. Details may be found in [23]. The model corresponding to the penalization parameter which minimizes the penalized risk is chosen as the final model.

Some additional features make CART an attractive procedure for applications.

- CART has a mechanism for internal re-weighting of data in case of unbalanced classes, which occurs quite frequently in BI applications.
- CART has a mechanism for handling missing values. Obviously, the tree can be applied only if the variables used in the splitting decisions are known. In case of missing values, CART offers so-called surrogate splits. This means that in tree building, for each knot of the tree, a degree of missingness is computed and alternative variables for the split decision are defined.
- CART automatically adapts to the number of classes and does not require a special mechanism.
- The estimation of the classification probabilities for the training data is rather simple, because we have to calculate only the frequency of observations of the different classes in the terminal nodes.

As weak points of tree classifiers, it is sometimes mentioned that tree building is rather sensitive to the ordering of the input variables and unstable with respect to changes in the training data. In order to overcome such problems, one can perform many classifiers and average the results of these classifiers, for example, by using a majority vote. Usually, only one training data set is available, and additional samples have to be generated. This technique is known as *bootstrap aggregation* or *bagging*. Bagging applies the general method of *bootstrap* to classification problems. The interested reader is referred to [6]. The basic steps in the bagging algorithm are as follows:

---

**Algorithm 3:** Bagging algorithm

---

```

1 begin
2   Given the training data  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ , generate  $B$ 
      bootstrap samples of size  $N$ 
      
$$(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \dots, (\mathbf{x}^{(B)}, \mathbf{y}^{(B)})$$

      from the data by sampling with replacement;
3   For each bootstrap sample  $(\mathbf{x}^{(b)}, \mathbf{y}^{(b)})$  learn a classification tree  $T_b$ ;
4   Define the final classification by taking the majority vote of the classifiers;
5 end
```

---

Random forests modify this procedure in such a way that in each split of a bootstrap classification only a random subset of the input variables is used. Such a random selection helps to overcome the problem that the different tree classifiers

may be correlated due to the fact that important predictors are used in all bootstrap classifications.

Sometimes it is mentioned as a disadvantage of the method that the interpretation of complicated trees may be tricky and that trees cannot cope with linear combinations of the variables. With respect to predictive power, other methods, such as neural nets, may be superior. However, in general, CART is considered as one of the most useful methods for classification.

Besides CART, other methods for tree classification exist. The oldest method is CHAID (chi-square automatic interaction detection) developed in the 1960s for partitioning data according to the values of nominal variables. This method does not require pruning. More popular are C4.5 or C5 which also allow growing trees with more than two child nodes. The ideas behind these methods, together with their implementation in Python, may be found in [21].

### 5.3.4 K-Nearest-Neighbor Classification

This classification method uses a very simple rule for assigning class membership. Given an observation  $x_{\text{new}}$ , the  $k$  closest points to the new observation are determined, and the class assignment is done according to the most frequent class in the  $k$ -nearest neighbors. Using instead of the absolute frequencies of the classes in the  $k$ -nearest neighbors the relative frequencies allows another interpretation of the classification rule: the classifier is based on estimates of the posterior probabilities of the classes. This shows that nearest-neighbor classifiers have a close connection to Bayes classifiers.

The idea of nearest-neighbor classification is intuitively appealing but depends on two decisions: first we have to define how to measure the distance between the observations and second we have to determine the value of  $k$ . The decision about the distances has to take into account the values of the variables. Sometimes it may be advantageous to standardize the variables. Another problem is measurement of the distance between qualitative variables. We will discuss this problem in more detail in Sect. 5.4.

The determination of  $k$  has to be done under consideration of the bias-variance trade-off discussed in Sect. 5.1. A small value of  $k$  makes the estimate rather unstable, and we can expect a large variance. A large value of  $k$  reduces the variance, but the estimate of the posterior probability will be closer to the priors of the classes. Note that if we take  $k$  equal to the number of observations of the training set, the estimate will be equal to the sizes of the classes in the training set. This may cause an additional bias for the decision. As it is shown in [14], the decision about  $k$  depends on the classification problem and may have substantial influence on the results. Cross-validation is proposed for finding the right  $k$ .

The nice property of nearest-neighbor classifiers is that they do not need an explicit training phase. This fact gives the method the name *lazy* classifier. The price for this laziness is the computational effort in finding the  $k$ -nearest neighbors. All

the data points of the training set must be kept in the memory and searched for the  $k$ -nearest neighbors. This fact justifies the name *memory-based* classifier.

$k$ -nearest-neighbor classifiers have been successfully applied for problems with many different prototypes, for example, in connection with classification problems of time series. We will show an application of that type in Chap. 6.

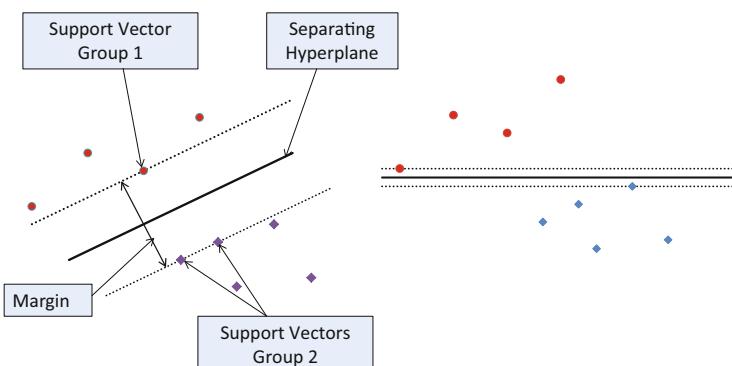
### 5.3.5 Support Vector Machines

To understand the ideas behind support vector machines for two classes, it is convenient to label the classes by  $Y \in \{-1, 1\}$ . For the separation of the classes, we use a linear function in the input variables  $f(x) = \sum w_k^T x_k$ . The classification rule is then defined by the sign of the function  $f(x)$ . The weights are defined in such a way that the following two properties hold:

- Explain the training data well
- Achieve maximum separation for correctly classified data

The first property corresponds to the idea of minimizing the training error if we measure explanatory power by the misclassification rate. The second property can be interpreted as generalization error: if we maximize the separation of the classes in the training data, we can expect that new data with a similar behavior will also be classified correctly. In [4, p. 409], an additional explanation of this idea in terms of Popper's principle of falsification is given.

Let us demonstrate this idea by a simple example shown in Fig. 5.10 for two-dimensional input data. The figure shows two possible solutions for a separating line between the two groups. For the evaluation of the two classifications, the concept of *margin* is used. The margin is defined by the points that are closest to the line in the two groups. In the figure, the margin corresponds to the distance between the two



**Fig. 5.10** Separation of two classes by hyperplanes. *Left*, a separating line with large margin; *right*, with small margin

parallel dotted lines. Points lying on the dotted lines are called *support vectors*. In the left panel in Fig. 5.10, we have one support vector for group 1 and two support vectors for group 2. In the right panel, there is only one support vector for each group. Obviously, the margin essentially depends on the slope of the separating line defined by the weights and the intercept. The best separating line is defined by that one which maximizes the margin for the training data.

Using standard linear algebra, the task of finding a linear function defined by the weight vector  $w$  and the intercept  $b$ , which maximizes the margin, is transformed into the following minimization problem:

$$\begin{aligned} \min \frac{1}{2} \|w\|^2 \\ y_i w^T x + b \geq 1 \quad i = 1, 2, \dots, n \end{aligned} \quad (5.27)$$

Support vectors are all those points that fulfill the constraint with equality.

The solution of the problem is found by transferring this quadratic optimization problem into its dual problem, which is also a quadratic optimization problem. Solving the problem is numerically demanding, because the number of variables corresponds to the number of training data points. A standard method for obtaining the solution is the *sequential minimization method*, which solves the problem iteratively considering only two multipliers in each iteration.

Using the solution  $(w^*, b^*)$  gives the classification rule

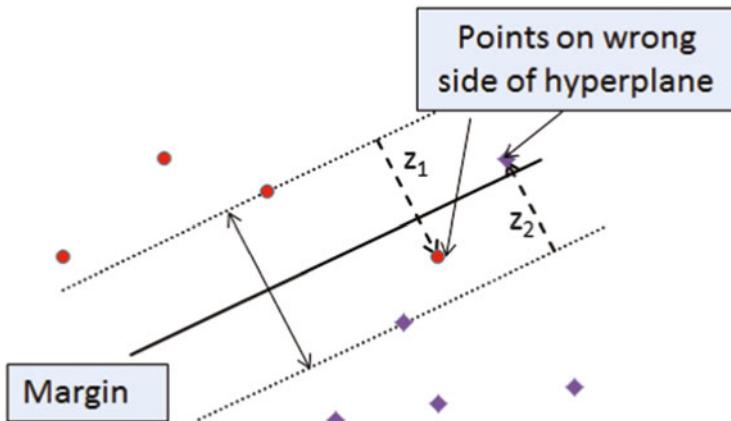
$$\hat{y} = \text{sign}(w^T x + b^*). \quad (5.28)$$

Up to now, we made the assumption that for training data, the groups can be separated by a linear function in the space of input variables. The answer whether such a separation is possible leads to the definition of the *Vapnik–Chervonenkis dimension* (VC dimension) for a class of functions, which is defined as follows:

**Definition 5.1 (VC Dimension)** Given a class of functions  $f(x, \theta)$ , the VC dimension is defined by the maximum number of points in any configuration that can be shattered by a function in this class.

The interpretation of the term “shattered” is as follows: if we choose points in an arbitrary position and assign the group labels 0 and 1 randomly to the points, then the groups can be separated perfectly by some function of the class. It can be easily seen that for linear functions in the plane, the VC dimension is 3. In case of four or more points, this is not always possible. Generally, the VC dimension of linear functions in  $p$  variables is  $p + 1$ .

This fact shows that using the concept of margin for separation, with linear functions for classification, needs some modifications. Two strategies can be applied. The first one are support vector machines with soft margin as demonstrated in Fig. 5.11. If we use a solid line for separation with margins defined by the dotted lines, then two points are on the wrong side of the separating line. The



**Fig. 5.11** Support vector machine with soft margin

size of violation of separation is given by  $z_1 + z_2$ . The idea is now to modify the minimization problem in Eq. (5.27) in such a way that misclassification is allowed, but the violation of the constraints penalizes the minimization function. This can be done by introducing new variables  $\xi_i$  that measure the misclassification and define the following minimization problem:

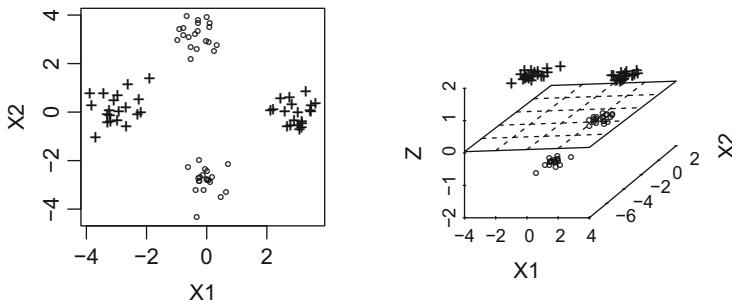
$$\begin{aligned} & \min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum \xi_i \\ & y_i \mathbf{w}^T \mathbf{x}_i + b \geq 1 - \xi_i \quad \xi_i \geq 0, i = 1, 2, \dots, n \end{aligned} \quad (5.29)$$

$C$  is a parameter measuring the trade-off between the complexity of the machine and separability.

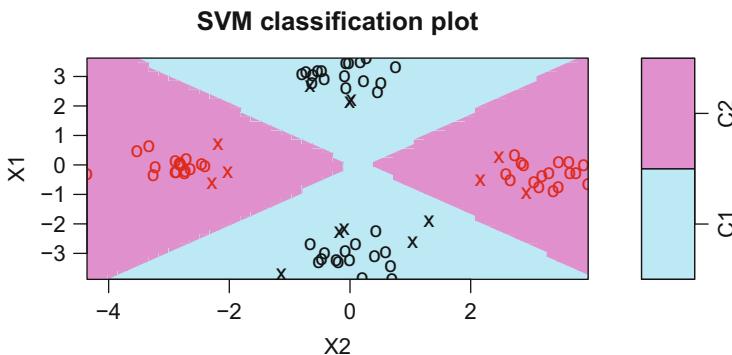
The second strategy allows nonlinear boundaries in the space of input variables by transforming the problem into a higher dimension. Figure 5.12 shows how this method works for a so-called XOR problem. The left panel shows data in two classes generated by normal distribution with different means. Obviously, a separation by a linear function is not possible. The right panel transforms the data into a three-dimensional space which allows separation in the three-dimensional space by a plane parallel to the  $(x, y)$  plane.

For the calculation of this transformation one uses the so-called *kernel trick*. This method changes the definition of the inner product by transforming the data with an inner product kernel. Frequently used kernel functions are *radial basis kernels* defined by

$$H(x, x') = \exp \left\{ \frac{-\|x - x'\|^2}{\sigma^2} \right\}. \quad (5.30)$$



**Fig. 5.12** XOR problem and transformation for linear separation (R package `scatterplot3d`)



**Fig. 5.13** Solutions of the XOR problem with radial basis kernel (R package `e1071`)

The centers of these functions  $x'$  and the number of functions correspond to the support vectors. The scaling factor  $\sigma$  is the same for all functions and depends on the data dimension. Other choices of kernels are polynomial kernels. The advantage of the kernel method is that it allows the calculation of the inner products and the distances between observations without explicit transformation of the data. For theoretical details, we refer to [24].

Figure 5.13 shows the solution for the XOR problem with radial basis kernel. The classification uses 14 support vectors, marked by  $\times$ , and gives a perfect solution. It should be mentioned that the solution depends on the used kernel functions. In this example, the solution with polynomial kernels is much worse.

Using support vector machines for classification allows the determination of the generalization error according to the VC theory. Results may be found in [4]. For model selection, one can use the technique of cross-validation.

The support vector machine is defined for two classes. A standard method for more than two classes is the method one against one. This means that we solve the classification for all pairs and assign that class which has the majority vote.

The calculation of the classification probabilities has to be done separately. One way to do this is by using a logistic distribution which fits best the function defining the decision boundary.

### 5.3.6 Combination Methods

Combination methods are motivated from the observation that there exists no single best method for all classification problems. One method to overcome this problem is to use random forests as described briefly in Sect. 5.3.3. Another frequently applied method is *boosting*. The theoretical background of boosting is the question on whether it is possible to boost a weak classifier to a strong classifier by repeated application. Under the term weak classifier, we understand a method with classification error only slightly larger than 0.5, i.e., the random guess of the classes taking into account the size of the classes. In terms of the ROC curve for two classes, this means that the area under the curve is only slightly larger than 0.5. Boosting shows that the question has a positive answer using the following basic algorithm:

#### Outline of the Boosting Algorithm

1. Start with a classifier  $f(x)$  for the training data using equal weights  $w$  for all observations.
2. Compute modified weights  $w^*$  for the data in such a way that misclassified data get a higher weight and correctly classified data get a lower weight.
3. Compute a new classifier  $f^*(x)$  for the data with the new weights  $w^*$  with the same method.
- 4 Repeat steps 2 and 3  $T$  times, and define a final classification function as a combination of the  $T$  classifiers.

As in the case of support vector machines, the starting point is a classification problem for two classes with training data  $((y_1 \mathbf{x}_1), (y_2, \mathbf{x}_2) \dots (y_N, \mathbf{x}_N))$  and class identifiers  $Y = \{-1, 1\}$ . The classification error for a classifier  $F$  is defined by the misclassification rate  $\varepsilon = \text{card}\{y_i \neq F(\mathbf{x}_i)\}/N$ . The rationale behind the update for the weights and the weighting of the individual classifiers is based on the *exponential loss* of a classification method which is defined by

$$L_{\text{exp}} = \exp(-yF(x)). \quad (5.31)$$

To find the optimal weights for the individual classifiers in the combination, we consider the loss of an observation if we use a linear combination  $F + \alpha f$  of the already existing classifier  $F$  and the new classifier  $f$ . The parameter  $\alpha$  is chosen in such a way that the exponential loss is minimized, and the new weights of the observations are calculated by a Taylor expansion for the exponential loss. A detailed derivation may be found in [25] where one may also find theoretical results

about the generalization error and references to the connection between boosting and logistic regression.

Usually, tree classification with few nodes is used as a weak classifier. This choice allows the usage of cross-validation for the test phase. The generalization for more than two classes is achieved by the method one against the rest.

### 5.3.7 Application of Classification Methods

In this section, we will apply the above introduced methods to data from the CRM use case (cf. Sect. 1.4.4).

#### CRM Use Case: Classification of Service Usage

For the comparison of the different methods, we applied the algorithms to predict the usage of a specific service in the CRM use case. As predictors, we used the variables `Sales`, the sales index `SalInd`, the duration of customer relationship `CR-Dur`, the indicators `Student` and `Office` for student users and office users, and the personal characteristics `Age_group` and `Sex`. There were missing values in the variables `Age_gr` (5 %), `Sex` (1 %), `Student` (7 %), and `Office` (7 %). Only customers with a relationship to the company longer than 12 months were considered. The data set encompassed 1,311 customers, from which 748 (75 %) used the service and 563 (43 %) did not use the service. We randomly selected a training data set of 905 customers (approximately 70 %). In the training data set 532 (59 %) used the service and 373 (41 %) made no use of the service. For classification, we used the R procedures for naive Bayes, logistic regression, tree classification, support vector machines, and AdaBoost with the following specifications:

- *Naive Bayes*: All variables were used, and for the quantitative variables, nonparametric density estimation was used because it seemed more realistic than a normal distribution (cf. the results about data visualization in Chap. 4). With respect to missing values, we used the standard strategy, i.e., the omission of cases with missing values.
- *Logistic regression*: For the training data, a model with the variables `Sales`, `Age_gr`, and `CR-Dur` was selected. This selection was supported by the significance of the coefficients as well as by the AIC criterion.
- *CART*: A model for the training data was learned using tenfold cross-validation. It turned out that the splits used the variable `Sales`, the sales indicator `SalesInd`, and the indicator for office.
- *Support vector machines*: We used a specification of a support vector machine with soft margin and a radial basis kernel. For the evaluation of the learned machine from the training data, tenfold cross-validation was used.
- *Boosting*: We trained a model with a maximum number of 10 training rounds using a version of AdaBoost with cross-validation for model selection.

Method	Accuracy	Sensitivity	Specificity	Area under ROC
NaiveBayes, Train	0.782	0.703	0.743	0.817
NaiveBayes, Test	0.665	0.662	0.668	
Logistic, Train	0.714	0.714	0.713	0.790
Logistic, Test	0.685	0.690	0.679	
CART, Train	0.762	0.832	0.662	0.781
CART, Test	0.729	0.824	0.621	
SVM, Train	0.707	0.705	0.710	0.791
SVM, Test	0.672	0.685	0.659	
AdaBoost, Train	0.874	0.705	0.804	0.895
AdaBoost, Test	0.782	0.595	0.695	

**Fig. 5.14** Summary of results using different classification methods

The results of the different classifications are shown in Fig. 5.14. For each method, we report sensitivity, specificity, overall rate of correct classification, area under the ROC curve for the training data, and classification results for the test data.

Details of the results can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

### 5.3.8 Summary: Classification Models

Classification aims at learning a decision rule for the class membership of a new observation from training data. This rule can be formulated either as an estimate of the class indicator or as an estimate of the class probabilities. For the decision of the class membership, a threshold for these probabilities is used. For the evaluation of a decision rule, two methods are the confusion matrix and the cross entropy or deviance. In the case of two classes the ROC curve is a useful tool for assessing the quality of the decision rule.

There are numerous methods for learning the decision rule which are based on different principles. We discussed probabilistic models like naive Bayes or logistic regression, tree-based methods, distance-based methods like nearest-neighbor classifiers, support vector machines which minimize the empirical risk, and combination methods which start with a weak classifier and iteratively improve the decision rule.

Rather independent from the used method, the analysis steps have to follow the general template for supervised learning. We start with descriptive methods for data understanding, split the data in a training and test set, learn a rule from the training set, and assess the decision rule with the test set. The model selection in the training phase depends on the classification method. There are methods like CART or logistic regression that allow integration of model selection in the development of the model. For other methods, the validation of the model has to be done by  $k$ -fold

cross-validation or by splitting the training data into a training set and a validation set.

In applications, different methods are applied and tuned to the problem. Afterwards, the most eligible method is used in the deployment phase of the project. In practice, the decision about the classification method depends on a number of factors. In the case of data with missing values, CART may be an interesting option because it offers a mechanism for handling missing values. If one is interested in the interpretation of the rule in subject matter terms, logistic regression has some advantages compared to other methods.

## 5.4 Unsupervised Learning

The term unsupervised learning refers to analysis goals without training data for the evaluation of the analysis results. In this section, we focus on unsupervised learning problems which aim at a segmentation of the data. They are summarized under the heading cluster analysis. From the numerous approaches to cluster analysis, we consider in this section hierarchical methods, partitioning methods, and model-based clustering. Clustering for temporal data will be treated in Chap. 6 and cluster analysis for text mining in Chap. 8.

### 5.4.1 Introduction and Terminology

In the matter of unsupervised learning, our data structure is similar to supervised learning, but we have only observed input variables  $X = (X_1, X_2, \dots, X_p)$  and no observed output variable. The analytical goal is finding a grouping of the observations, so-called *clusters*, that can be used later on for explaining the structure of the observations in the context of the domain. We will discuss two different modeling approaches for defining the clusters: *distance-based methods* and *model-based methods*.

In the first case, the main prerequisite for modeling is the definition of a distance between observations, which measures the similarity or dissimilarity of observations. In the case of quantitative variables, the most important distance is the Euclidean distance defined for two  $p$ -dimensional vectors  $x$  and  $z$  by the equation

$$d^2(x, z) = \sum_{j=1}^p (x_j - z_j)^2 = \|x - z\|^2. \quad (5.32)$$

In the case of qualitative variables, the most frequently used distance is the *Hamming distance*. This distance uses dummy coding for the different values of the qualitative variable, i.e., each value corresponds to a dummy variable with the

values 0 and 1. Using such coding, we obtain a bit string for each observation, and the distance between bit strings is defined by the number of different bits. Note that this distance corresponds to the Euclidean distance for the dummy variables.

Measuring the distance between observations with variables of different magnitude puts an emphasis on the distance between the variables with larger scale. Hence, it is usually recommended to standardize the variables into a value range [0, 1]. Such standardization is of utmost importance if we want to measure the distances between observations with qualitative and quantitative variables. A detailed description of a general method for defining distances can be found in [9] and is implemented in R<sup>1</sup> as the distance function *daisy*.

Two different approaches are commonly used for clustering observations based on distances. The first one comprises so-called *hierarchical methods* which define a hierarchy tree for the observations. Each node in the tree represents a possible subset (cluster) of the observations, the root defines one cluster containing all observations, and the leafs of the tree are the observations representing  $N$  different clusters. Cutting the tree at a certain level provides a possible cluster solution. We will treat such methods in Sect. 5.4.2. The second approach consists of so-called *partitioning methods* which define a number of clusters in advance and assign the observations iteratively to the clusters. Such methods will be treated in Sect. 5.4.3.

Besides the formulation of clustering based on distances, we can formulate the problem in the context of finding the distribution of the observations as described in [12] under the topic *descriptive methods*. Such methods assume that the distribution of the observations is a mixture distribution:

$$f(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \dots + \alpha_p f_p(x). \quad (5.33)$$

Here,  $p$  is the number of clusters,  $\alpha_i$  are the proportions of the clusters in the observations, and  $f_i(x)$  are the densities of the distributions in each cluster. The number of clusters  $p$ , the mixture probabilities  $\alpha_i$ ,  $\sum \alpha_i = 1$ , and the distribution in the components are unknown. In Sect. 5.4.4, we will present a solution for this approach under the assumption that the distribution in the different clusters is a multivariate normal distribution.

As soon as we have defined a cluster solution, we are also interested in the representation of the clusters by a typical element of the cluster. This characterization connects cluster analysis with the method of *vector quantization*, which is a more decision-theoretic approach for dimensionality reduction. The goal in vector quantization is finding  $p$  representatives for the clusters, for example, a customer with typical behavior. Details on the difference between vector quantization and clustering may be found in [4]. The main difference to vector quantization is that clustering does not use a decision-theoretic-oriented criterion for the overall evaluation of a solution, but more application-oriented criteria, such as reliability and validity, as discussed in 2.1.4.

---

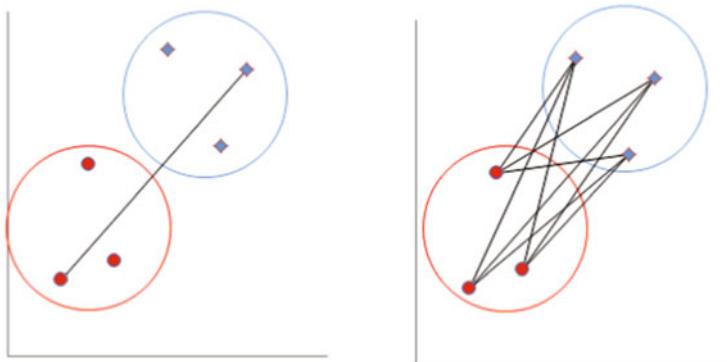
<sup>1</sup><http://www.r-project.org/>.

The validity and reliability of the solution are also of utmost importance for model assessment, model selection, and model evaluation. Besides this, we will consider a number of diagnostic tools for model assessment. With respect to model testing, the method of using an independent set of test data is usually recommended. The following template summarizes the main steps in cluster analysis:

Template: Cluster Analysis
• <b>Relevant Business and Data:</b> Customer behavior represented as cross-sectional data for process instances with a matrix $\mathbf{X}$ of explanatory variables
• <b>Analytical Goals:</b>
– Find a segmentation of the data into clusters which allows an interpretation from domain point of view
– Determine representatives for the clusters
• <b>Modeling Tasks:</b> Definition of a model for data description either based on the distances between the observations or by a mixture model for the distribution
• <b>Analysis Tasks:</b>
– <i>Splitting Data:</i> If necessary split the data randomly into one set for training and another for testing the model
– <i>Model Estimation:</i> Estimate the cluster solutions
– <i>Model Assessment:</i> Assess the quality of models with respect to homogeneity of the clusters, separation between clusters, validity and reliability
– <i>Model Selection:</i> Select a model by specifying the number of clusters
• <b>Evaluation and Reporting Task:</b> Evaluate the selected model using test data

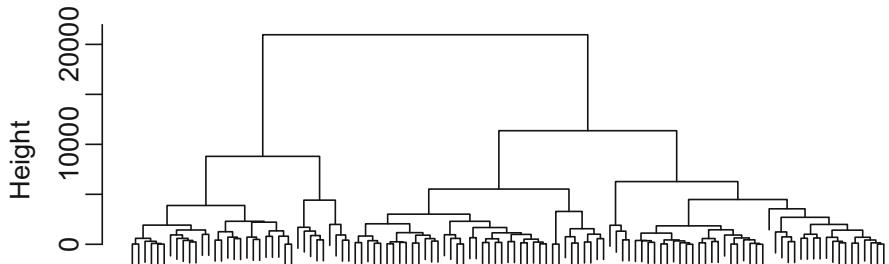
### 5.4.2 Hierarchical Clustering

Starting from a distance matrix between observations, we can define a distance  $D(C_j, C_k)$  between sets  $C_j$  and  $C_k$  representing clusters of observations, called *linkage*. Figure 5.15 shows two frequently used techniques, i.e., the *average linkage* in the right panel and *complete linkage* in the left panel. Average linkage is defined as the mean distance between all observations in the two clusters and complete linkage as the maximum of the distances between points in the two clusters. A third important technique is the Ward method, which measures the distance between two clusters by comparing the total within-cluster sum of squares for the two clusters separately with the within-cluster sum of squares resulting from merging the two clusters.



**Fig. 5.15** Distances between clusters using average linkage (*left*) and complete linkage (*right*)

### Dendrogram, complete linkage



**Fig. 5.16** Cluster tree with complete linkage (R package *cluster*)

Based on a definition of the distance between clusters, one can construct a cluster tree using *agglomerative methods* as outlined in the following algorithm:

---

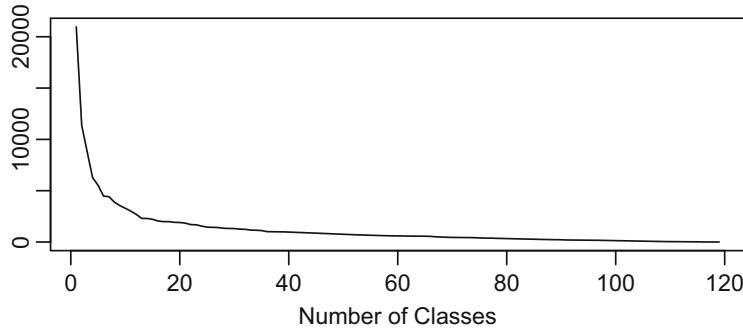
#### Algorithm 4: Agglomerative clustering

---

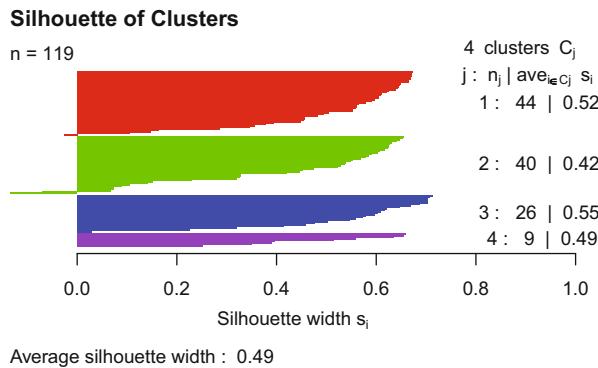
- 1 Define clusters  $C_k$ ,  $1 \leq k \leq N$  by the observations,  $N_{cl} = N$ ;
  - 2 **for**  $k = 1$  to  $N - 1$  **do**
  - 3     Merge clusters  $C_r$  and  $C_s$  for which  $d(C_r, C_s) = \min_{(l,k)} D((C_l, C_k))$ ;
  - 4      $N_{cl} = N_{cl} - 1$ ;
  - 5 **end**
- 

The resulting hierarchy can be visualized by using a tree diagram or *dendrogram* as shown in Fig. 5.16. The length of the branches called height corresponds to the distance between the merged objects.

Using the tree, one can decide about the number of clusters according to the height. Merging two clusters is not advisable if the height of the branches measured from the merged cluster to the individual clusters is a substantial value. The evaluation of distances is easier using a *scree plot*, which shows the distance between the clusters plotted against the number of clusters. Figure 5.17 shows the



**Fig. 5.17** Scree plot for cluster fusion (R graphics)



**Fig. 5.18** Silhouette plot for cluster solution (R package cluster)

scree plot for the cluster tree depicted in Fig. 5.16. As one can learn from both figures, a solution with four clusters seems reasonable.

As already mentioned in Sect. 5.3.1, the evaluation of a cluster has to be mainly accomplished in the context of the domain problem. The interpretation of the clusters can be supported by descriptive statistics for variables. A formal evaluation of quality of the cluster assignment can be done using *silhouettes*. For constructing the silhouette, two different distances are calculated: the average distance from each observation to the observations in the same cluster and the minimum distance from the observation to observations in other clusters. From these two distances, silhouette values for each observation are calculated by standardization into the interval  $[-1, 1]$ . The silhouette values of the observations are plotted for each cluster in decreasing order as shown in Fig. 5.18. Large positive values indicate that the observation is well located in the cluster. Negative values are an indicator that the observation may not be well assigned. Besides this visual inspection of the cluster solution, one can calculate a silhouette coefficient as the overall quality measure of the solution. Details may be found in [18].

In the following, we will demonstrate the principle of clustering in the context of the HEP use case (cf. Sect. 1.4.2).

#### HEP Use Case: Clustering of Student Performance

In practical exercises about algorithms, students have to solve programming examples which are uploaded and tested. The following data were collected from 109 students attending the course: the phase of the course in which the solution was uploaded; the number of trials the student made; the size of the program; the minimum, maximum, and average length of the messages written in the forum; and the number of read-and-write activities in the forum. Cluster analysis was done with and without standardization of the variables. For not standardized variables, the cluster tree is shown in Fig. 5.16. The dendrogram and the scree plot in Fig. 5.17 suggest a solution with four clusters. Due to the fact that the variable size of the program has a larger scale than the other variables, the division into the clusters is dominated by the variable size. The interpretation of the clusters is as follows:

- The first cluster comprises 44 students with short programs uploaded in the first phases. In average, these students needed only 14 trials. With respect to the forum activities, these students were not very active.
- The second cluster consists of 40 students who uploaded longer programs, needed about 30 trials in average, and were more active in the forum.
- The third cluster of 26 students uploaded longer programs than the first cluster, and they were much more active in writing in the forum.
- Finally, the fourth cluster comprises nine students with very large programs, which were uploaded very late, after almost 40 trials in average.

Figure 5.18 shows the silhouette plot of the solution. The silhouette coefficient is 0.49.

As expected, the analysis of the standardized data showed that the size of the program does not dominate the classification. More important are the activities in the forum. This solution also shows two clusters which are dominant in size, and two clusters encompass the students with exceptional behavior.

This example shows that both approaches provide results which can be interpreted from a subject matter point of view.

Details can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

### 5.4.3 Partitioning Methods

Partitioning methods define a cluster solution for observations for a given number of clusters in an iterative way. The most popular method is *K-means* which defines the solution by the following algorithm:

---

**Algorithm 5:** *K*-means algorithm
 

---

**Data:** Observation matrix  $\mathbf{X}$  and distance for the objects; number of clusters  $K$ .

**Result:** Cluster solution for observations

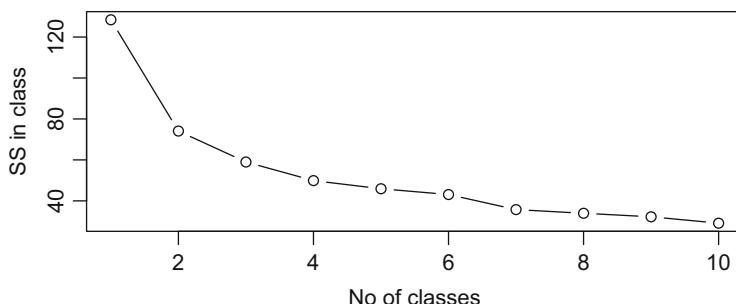
```

1 begin
2   Define an initial solution for the cluster centers  $(c_1, c_2, \dots, c_k)$ ;
3   Assign each observation  $x$  to the cluster which center is closest to the
      observation;
4   Compute new centers for the clusters as means of the assigned
      observations;
5   Repeat steps 2 and 3 as long as there is no significant change in the centers;
6 end
```

---

Two decisions of the user are important for the application. The first one is deciding about the number of clusters. For this decision, it is useful to plot the sums of squares within the clusters for different solutions and use this diagram in a similar way as the scree plot in the case of hierarchical clustering. Figure 5.19 shows such a graph for solutions between one and ten clusters. Further, a solution with four or five clusters seems to be a reasonable choice in this case. The second decision is about the initial centers. The standard approach is to choose the centers at random and to compute different solutions. In most implementations, one can also provide starting solutions.

The simple structure of the algorithm makes *K*-means an attractive procedure for large data sets, and it can be implemented easily on parallel architectures. The iterative structure also allows modifications for applications with rapidly changing



**Fig. 5.19** Sum of squares within clusters (R graphics)

data from transactional databases.  $K$ -means is also frequently used for vector quantization.

One disadvantage of the procedure is the lack of robustness with respect to outliers. In the case of clustering, outliers can be interpreted as small groups of observation with rather irregular behavior. Such data may cause solutions which are rather sensitive regarding the starting values.

A more robust alternative to  $K$ -means is the *PAM algorithm*, which stands for partition around medoids. The concept of a *medoid* is a multidimensional generalization of the concept of the median, i.e., a central point in the multivariate data. Contrary to the mean, a medoid is not computed by a formula, but it is an existing data point. For a precise definition, we refer to [18]. The algorithm also works iteratively from a starting set of medoids. One iteration consists of the assignment, the swapping, and the calculation of new medoids. In the assignment step, observations are allocated to the closest medoid. The swapping step tries to find observations in the clusters with smaller distances from the data points in the cluster than the existing medoid. The new medoids are defined by the points with the smallest distance from the actual solution. For details, consider [18].

In the following, we will demonstrate the application of both methods in the context of the CRM use case (cf. Sect. 1.4.4).

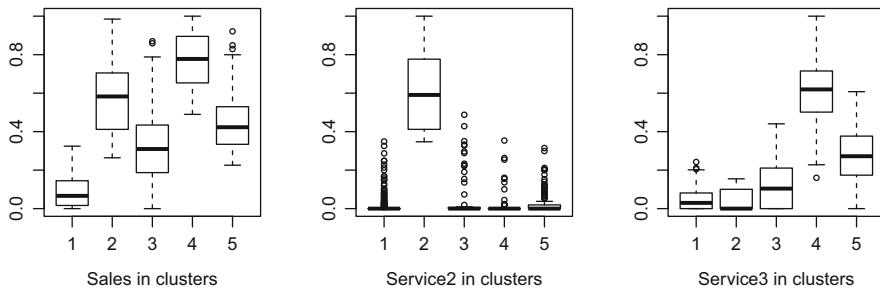
#### CRM Use Case: Clustering of Customers

We are interested in finding user profiles of customers with respect to the different services. Four services are of main interest: Service 2, Service 3, Service 4, and Service 9. Furthermore, the variables Sales for the actual overall sales, the activity index ActInd as measure for actual performance, and the variable SalesInd for sales in the past are used. A first data inspection showed that the variables have numerous outliers, mainly due to the fact that there are extreme power users. Hence, we decided to perform two different analyses. The first one used the data of 528 regular users, where  $K$ -means was used for standardized data. Figure 5.19 shows the decrease in sum of squares for solutions with 1 up to 10 clusters. We chose a solution with five clusters. The 528 observations are distributed to the clusters as follows:

Cluster 1, 43 %; Cluster 2, 4 %; Cluster 3, 25 %; Cluster 4, 14 %; and Cluster 5, 14 %.

For the interpretation of the cluster, we used the distribution of the variables in the clusters shown in Fig. 5.20 for Sales, Service 2, and Service 3. This leads to the following interpretation:

- Cluster 1 may be characterized as “small users.”
- Cluster 2 is a small group of intensive users, who are the main users of Service 2.
- Cluster 3 are moderate users using mainly Service 3.
- Cluster 4 is a group of heavy users, with high usage in Service 3.
- Cluster 5 are average users, mainly using Service 3.



**Fig. 5.20** Distribution of variables within clusters (R graphics)

For the second data set with 1,311 users,  $K$ -means was not able to calculate a useful solution. Although we used standardized variables in this case, a solution with five clusters showed one major cluster comprising more than 75 % of all cases of average users and four clusters differentiating between different types of extreme power users. This problem was independent from the chosen starting values. The PAM algorithm produced a more reliable solution. It differentiated between 519 low-level customers using mainly Service 2 and Service 3, 490 average customers using mainly Service 3, a group of 199 heavy users with a high activity index and high usage of Service 3, and two small groups of 53 and 50 customers being power users, one mainly in Service 2 and the other mainly in Service 3.

More results can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

Besides these two clustering methods, one can find many other clustering algorithms in the literature. A well-known and frequently used method are *self-organizing maps* (SOM), which define a neural network for clustering. In [14], it is shown that SOMs can be also interpreted as a special kind of  $K$ -means clustering, where the clusters are defined by a distorted grid. In IBM/SPSS<sup>®2</sup> two-stage clustering is used, which is a combination of hierarchical methods and  $K$ -means clustering based on the idea of a cluster feature tree. Details may be found in [26].

#### 5.4.4 Model-Based Clustering

Model-based clustering reformulates the problem of finding groups into a problem of estimating a mixture model for the distribution of the observations. We will consider here only the case of mixtures of normal distributions. The model starts

<sup>2</sup><http://www-01.ibm.com/software/analytics/spss/>.

from the assumption that observed data  $\mathbf{X}$  are generated from a population defined by  $p$  subpopulations  $G_1, G_2, \dots, G_p$ . For the clusters in the population, the term strata is frequently used. The proportion of subpopulation  $G_j$  in the population is denoted by  $\alpha_j$ . In each subpopulation, the observed variables are multivariate normally distributed with mean vectors  $\mu_j$  and covariance matrices  $\Sigma_j$ . Under these assumptions, we can denote the density function of an observation  $x$  as a mixture of normal distributions

$$f(x) = \sum_{j=1}^k \alpha_j \phi(\mu_j, \Sigma_j), \quad \alpha_j \geq 0, \quad \sum_{j=1}^k \alpha_j = 1. \quad (5.34)$$

This model transforms the cluster problem into the problem of estimating the unknown parameters  $\mu_j$  and  $\Sigma_j$  of the normal distributions and the proportions  $\alpha_j$ . If we know the parameters of the model, we can determine for an observation  $x_i$  the values of the densities  $f_j(x)$  and assign the observation to the most plausible cluster defined by the group with the highest value for the density. Note the similarity to a classification problem.

The major challenge of this approach is that we have no training data for learning the probabilities, and we do not know in advance how many different clusters exist. The solution is achieved in two steps similar to the ideas for model selection in supervised learning. In a first step, the parameters of the model are estimated given a fixed number of clusters, and in a second step, a model is selected according to a model selection criterion as introduced in Sect. 5.2.2.

For the estimation of the parameters  $\mu_j$ ,  $\Sigma_j$ , and  $\alpha_j$  for a given number of clusters, the *EM algorithm* is used. This algorithm is a general method for computing maximum likelihood estimates in case of missing information. We will explain the basic idea of the algorithm only for the case of a mixture of two normal distributions with the same covariance structure. A general treatment of the EM algorithm may be found, for example, in [25] or [14].

In the case of a mixture of two normal distributions with the same covariance structure, the density of the observations is defined by

$$f(x, \theta) = \alpha \phi(x, \mu_1, \Sigma) + (1 - \alpha) \phi(x, \mu_2, \Sigma), \quad \theta = (\alpha, \mu_1, \mu_2, \Sigma) \quad (5.35)$$

For the group membership of the observation, we introduce a dummy variable  $Z$

$$Z(X) = \begin{cases} 1 & \text{if } X \in \text{Group } G_1 \\ 0 & \text{if } X \in \text{Group } G_2 \end{cases} \quad (5.36)$$

for class membership. Using this dummy variable, we can write the distribution of an observation as

$$f(x, z, \theta) = \alpha^z \phi(x, \mu_1, \Sigma)^z + (1 - \alpha)^{1-z} \phi(x, \mu_2, \Sigma)^{1-z}, \quad \theta = (\alpha, \mu_1, \mu_2, \Sigma) \quad (5.37)$$

and understand the distribution in (5.32) as the marginal distribution of the variables  $(X, Z)$  with respect to  $X$ . If we would know the values of  $Z$  for each observation, the estimation of means  $\mu_j$  and covariances  $\Sigma_j$  could be done by standard formulas. The EM algorithm uses this fact and defines the solution in two steps. The basic iteration is shown in the following algorithm:

---

**Algorithm 6:** EM algorithm
 

---

**1 repeat**

2     Expectation-Step: Given estimates  $\hat{\theta} = (\hat{\mu}_1^{(r)}, \hat{\mu}_2^{(r)}, \hat{\Sigma}^{(r)}, \hat{\alpha}^{(r)})$ ) for the parameters, compute the expected values of the loglikelihood given the data and the parameter estimate:

$$J(\theta, \hat{\theta}^{(r)}) = E[\log(\frac{f(x, z, \theta)}{f(x, z, \hat{\theta}^{(r)})}) | (x, \hat{\theta}^{(r)})] ;$$

3     Maximization-Step: Find new parameter values

$$\hat{\theta}^{(r+1)} = \arg \max_{(\theta)} J(\theta, \hat{\theta}^{(r)}) ;$$

4 **until** convergence is reached;

---

The idea behind the expectation step is replacing the unknown variable  $Z$  by the expected value. In the case of normal distributions, the calculation can be done using Bayes' theorem. It can be shown that this procedure converges, although the iteration could be rather slow.

In model selection, these calculations are done for different numbers of clusters and for different possible assumptions about the covariance structure for the observations. These covariance may be the same in all groups, but they may also differ between the classes. For the selection of a model, the BIC criterion (cf. Sect. 5.2.2) is used. The chosen model has the smallest value for BIC.

#### 5.4.5 Summary: *Unsupervised Learning*

One important goal in unsupervised learning is finding clusters of similar observations. The primary modeling task in cluster analysis is the definition of the similarity, or distance, between the observations using the observed variables. The most frequently used distance for quantitative variables is the Euclidean distance; in the case of qualitative variables, one can use the Hamming distance after dichotomization of the variables by dummy variables. An important decision in modeling is whether the variables should be standardized. In general, standardization is preferred. Based on the distances, two basic analysis techniques can be used. The first one are hierarchical methods which define clusters by aggregation procedures. Important decisions for these algorithms is the definition of the distance between the clusters and the number of clusters. The second approach are partitioning methods which start with a predefined number of clusters and assign iteratively the observations to the clusters.

The model selection task in cluster analysis is the determination of the number of clusters. Various descriptive tools can be used which support the selection of the number of clusters. Also for the evaluation of the cluster solution, these descriptive tools can be applied. For assessment of the generalization of the cluster solution, splitting the observations in training data and test data is appropriate.

Besides these two basic techniques, there are numerous algorithms which combine the two approaches. An alternative to distance-based methods is using a model-oriented approach that defines a mixture model for the observations. The estimation of the parameters of the mixture model uses the EM algorithm.

## 5.5 Conclusion and Lessons Learned

In this chapter, we introduced different techniques for the analysis of cross-sectional data. Depending on the analysis goal, we distinguished between methods for the achievement of predictive goals and for descriptive goals. Predictive goals are formulated as regression problems or as classification problems. In both cases, the definition of an appropriate loss function is essential. From the methodological point of view, balancing between overfitting and underfitting has to be done using a training and a test sample. In the case of unsupervised learning, we considered the formulation as clustering problem based on distances and as estimation problem for mixture distribution.

## 5.6 Recommended Reading

There exist many excellent books about data mining and machine learning which cover the material of this chapter in more detail. An introduction which emphasizes business applications is Linoff and Berry (2011) ([19]). For a more theoretical oriented approach from the statistical perspective, we recommend Hastie et al. (2009). A concise description of the most important data mining algorithms can be found in Wu and Kumar (2009). For readers interested in the algorithmic perspective and using Python for data mining, we recommend Marsland (2009).

- Hastie T, Tibshirani R, Friedman J (2009) *The elements of statistical learning*. Springer, 2nd edition
- Linoff GS, Berry MJA (2011) *Data mining techniques for marketing, sales, and customer relationship management*. Wiley
- Marsland S (2009) *Machine learning—an algorithmic perspective*. CRC Press
- Wu X, Kumar V (2009) *The top ten algorithms in data mining*. CRC Press

## References

1. Bowman AJ, Azzalini A (1997) Applied smoothing techniques for data analysis. Oxford Science Publications, Oxford
2. Breiman L, Friedman J, Stone CJ, Olshen RA (1984) Classification and regression trees. CRC, Boca Raton
3. Burnham KB, Anderson DA (2004) Model selection and multimodel inference: a practical information-theoretic approach. Springer, New York
4. Cherkassky V, Mulier F (2007) Learning from data—concepts, theory and methods. Wiley, New York
5. de Boor C (2001) A practical guide to splines. Applied mathematical sciences. Springer, New York
6. Efron B, Tibshirani RJ (1993) An introduction to the bootstrap. Chapman & Hall/CRC, Boca Raton
7. Faraway JJ (2004) Linear models with R. Chapman & Hall/CRC texts in statistics. CRC Press, Boca Raton, FL
8. Friendly M, Kwan E (2009) Where's Waldo? Visualizing collinearity diagnostics. *Am Stat* 63(1):56–65
9. Gower JC (1971) A general coefficient of similarity and some of its properties. *Biometrics* 27:857–874
10. Günther F, Fritsch S (2010) Neuralnet: training of neural networks. *R J* 2(1):30–38
11. Hand DJ, Yu K (2001) Idiot's Bayes—not so stupid after all? *Int Stat Rev* 69:385–398
12. Hand DJ, Mannila H, Smyth P (2001) Principles of data mining. MIT, Cambridge, MA/London
13. Hastie TJ, Tibshirani RJ (1990) Generalized additive models. Chapman & Hall/CRC, Boca Raton
14. Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning, 2nd edn. Springer, New York
15. Haykin S (2008) Neural networks and learning machines. Prentice Hall, Upper Saddle River
16. Hornik K, Stichcombe M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Netw* 2:359–399
17. Hosmer DW, Lemeshow S (2000) Applied logistic regression. Wiley texts in statistics. Wiley, New York
18. Kaufman L, Rousseeuw PJ (1990) Finding groups in data: an introduction to cluster analysis. Wiley-Interscience, New York
19. Linoff GS, Berry MJA (2011) Data mining techniques for marketing, sales, and customer relationship management. Wiley, New York
20. Maronna RR, Martin D, Yohai V (2006) Robust statistics—theory and methods. Wiley, New York
21. Marsland S (2009) Machine learning—an algorithmic perspective. CRC, Boca Raton
22. Ramsay JO, Silverman BW (2005) Functional data analysis, 2nd edn. Springer, New York
23. Therneau TM, Atkinson EJ (2014) An introduction to recursive partitioning using RPART routines. <http://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf>. Accessed 24 June 2014
24. Vert M, Tsuda K, Schölkopf (2004) A primer on kernel methods. In: Kernel methods in computational biology. MIT, Cambridge, pp 35–70
25. Wu X, Kumar V (2009) The top ten algorithms in data mining. CRC, Boca Raton
26. Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: an efficient data clustering method for very large databases. In: Jagadish HV, Mumick IS (eds) SIGMOD'96: International conference on management of data. ACM, New York, pp 103–114

# Chapter 6

## Data Mining for Temporal Data

**Abstract** In this chapter, we present analysis techniques for temporal data. First of all, we discuss the different data structures in temporal mining, introduce the different analytical goals and models, and give an overview on the corresponding analytical techniques subsequently. Section 6.2 considers time warping and response feature analysis for clustering and classification, Sect. 6.3 discusses regression models and their role in predicting the time period until the occurrence of an event, and Sect. 6.4 introduces the analysis of Markov chains. The following sections deal with analysis techniques for temporal patterns, in particular association analysis, sequence mining, and episode mining.

### 6.1 Terminology and Approaches Towards Temporal Data Mining

In Chap. 3, we have introduced log formats as the basic data structure for data with temporal reference. Given a sequence of ordered time values  $t_1 \leq t_2 \leq \dots \leq t_T$  called *observation times* and attribute values  $x_1, x_2, \dots, x_T$ , a log format defines *temporal data* as pairs  $x = ((t_1, x_1), (t_2, x_2), \dots, (t_T, x_T))$ . Besides the terms temporal data and observation time, we will use the notions *time-stamped data* for the sequence and *time stamps* for the observation times. In applications, the time stamps are often not exactly known, and we can only define ordered sequences of attributes  $\langle x_1, x_2, \dots, x_T \rangle$  which are called *sequential data*. The term *temporal data mining* summarizes methods for the analysis of time-stamped data and sequential data. Besides BI, the analysis of temporal data plays an important role in many domain areas, in particular, bioinformatics, speech recognition, or spatiotemporal information systems. Within these different application areas, various models and analysis techniques have been developed, frequently utilizing a specific terminology. An overview with an extensive bibliography may be found in [18].

For BI applications, it is important to specify the interpretation of time and the attributes of temporal data. With respect to time, we understand observation times as so-called *valid time*. This means that the time stamps refer to the point time where the information given by the attribute is true in the real world. For analysis purposes, this is the natural interpretation of the time stamps. In *temporal databases* besides the valid time, the *transaction time* is also considered. It is defined as the time when

the information is entered in the database. Data with at least two time dimensions, i.e., valid time and transaction time, are called in [21] *fully temporal data*. In this book, we abstain from a detailed discussion on temporal databases and focus on temporal analysis techniques in the context of BI projects.

With respect to the attributes, the following definitions distinguish between temporal data as information about the business process in the state view and in the event view (cf. Sect. 1.2.3).

### Definition 6.1 (Time Sequences, Time Series, State Sequence)

- a) A *time sequence* is defined as a sequence of time-stamped data for which the attribute values are the result of measurements of a quantitative real-valued state variable  $Y$ , i.e.,  $y \in \mathbb{R}$ . We denote the observations of a time sequence by  $\mathbf{y} = (y(t_1), y(t_2), \dots, y(t_T))$ .
- b) A *time series* is a time sequence with equidistant predefined observation times denoted by  $\mathbf{y} = (y_1, y_2, \dots, y_T)$ .
- c) A *state sequence* is a time sequence where the state variable  $S$  attains only a finite number of possible values given by a set  $\mathcal{S} = \{s_1, s_2, \dots, s_K\}$ . If the observation times are of minor importance, or even not known, we denote a chain simply as ordered sequence of observations of the state variable  $\mathbf{s} = < s_1, s_2, \dots, s_T >$ ,  $s_i \in \mathcal{S}$ .

The reader familiar with the notion of stochastic processes will recognize that a time sequence corresponds to observations of the path of a stochastic process at certain times  $(t_1, t_2, \dots, t_T)$ , but in order to avoid confusion with the understanding of the term *process* as business process throughout the book, we use the notion of time sequence. The term time sequence is always understood as observations of a continuous process, for example, the temperature curve taken for a container in a certain time window. The definition of a time series is in agreement with the usual understanding and emphasizes the distinction between continuous and discrete stochastic processes. An example for a time series are the temperatures taken within a sequence of days, i.e.,  $< (\text{day 1}, 20), (\text{day 2}, 25), \dots >$ . The definition of a state sequence corresponds to the terminology introduced in Sect. 2.4.4 for Markov chains.

### Definition 6.2 (Event Set, Event Sequence)

- a) Given a set  $\mathcal{E} = \{e_1, e_2, \dots, e_K\}$  of events, an *event set* is subset  $E$  of  $\mathcal{E}$ .
- b) An *event sequence* is an ordered list of events  $\mathbf{s} = < e_1, e_2, \dots, e_T >$ .

If the times of the events are known, event sequences are denoted by  $\mathbf{s} = < (e_1, t_1), (e_2, t_2), \dots, (e_T, t_T) >$ .

Note that formally event sequences and state sequences have the same structure. The difference is mainly in the interpretation as outlined in Sect. 1.2.3.

There exist numerous approaches for the representation of temporal data in modeling and analysis. These representations are obtained by transformations called frequently *feature extraction*. In [18], four approaches towards feature extraction are discussed. A similar distinction may be found in [4].

**Feature Extraction Approaches for Temporal Data:**

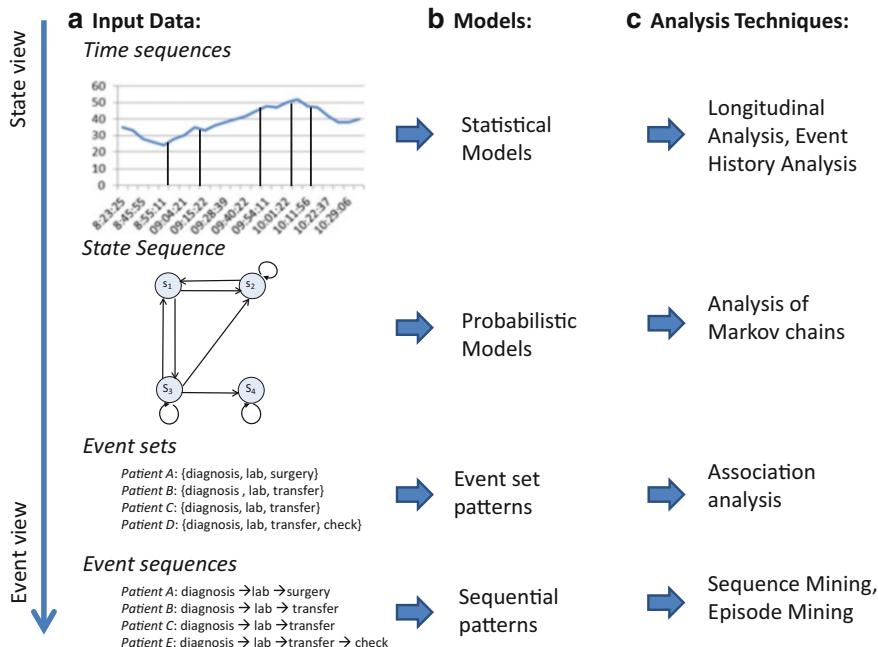
1. *Nonadaptive representation methods* transform a time sequence in a feature space allowing a representation in a lower dimension. A typical example is the *Fourier transform*, which transforms the sequence into a space of frequencies. Such representations are useful if the time sequence is interpreted as an observation of a signal composed of different basic frequencies, for example, a sound.
2. *Adaptive representation methods* extract, in dependence of the data, characteristic features of the time sequence. There exists a broad range of adaptive representation methods. Typical examples are the identification of time points with characteristic values, e.g., a maximum or a minimum, local regression models, and shape description languages which characterize the time sequence by the behavior of the gradient within predefined time windows.
3. *Data-dictated representation methods*, such as clipping, transform the time sequence in a bit string which indicates the deviations from the overall mean of the time sequence.
4. *Model-based representation methods* use the time sequence as input for statistical or probabilistic models.

In this chapter, we discuss a number of model-based analysis techniques and adaptive representation methods for the achievement of the following analytical goals.

**Analytical Goals in Temporal Data Mining**

- **Analytical goals for time sequences:**
  - Segmentation of time sequences into clusters of process instances
  - Prediction of properties of interest for a time sequence
  - Classification of time sequences
- **Analytical goals for state sequences:**
  - Understanding the structural behavior of the state sequences
  - Segmentation of instances of state sequences into clusters
- **Analytical goals for event sets and event sequences:**
  - Given a sample of event sets, find frequent event patterns and derive rules for the co-occurrence of events
  - Given a sample of event sequences, find frequent patterns in event sequences

Similar to the analysis of cross-sectional data, the analytical goals for time sequences are, in most cases, formulated as goals in the customer perspective. In the case of state sequences, event sets, and event sequences, the analytical goals may be related to all business perspectives.



**Fig. 6.1** Temporal analysis: input data, model structures, and analysis methods

Corresponding to the multitude of data specifications and analytical goals, a number of model structures can be defined, and the analyst has different analytical techniques at her/his disposal. The different specifications discussed in this chapter are summarized in Fig. 6.1. In Fig. 6.1a, the data are arranged from top to bottom along the state view towards the event view (cf. Fig. 1.2) with iconic representations. A similar distinction is also made in [15] or [18].

Figure 6.1b defines the model structures used for the different data. For time sequences, we investigate statistical models as described in Sect. 2.4.4, and for state sequences, we use Markov chains as probabilistic models. In the case of event sets and event sequences, the structure of interest are patterns as outlined in Sect. 2.1.5.

The analysis techniques treated in this chapter are listed in Fig. 6.1c. Even though we have used the general terminology introduced in Sect. 1.2.4 for the formulation of analytical goals for time sequences, the analysis techniques have to be adapted to meet the requirements of the temporal data structure. Let us explain this by the goals in connection with the use cases introduced in Sect. 1.4.

- In the logistics use case, we are interested in finding similarities between time sequences of temperature measures. Such a similarity has to take into account that the measurements at different times are not independent attributes but have a common structure. Hence, similarity has to be defined in another way as in the case of clustering for cross-sectional data in Sect. 5.4.

- In the EBMC<sup>2</sup> use case, we are interested in the survival times of patients with a certain stage of skin cancer. This means that we want to predict the duration of the illness in dependence of personal attributes, taking into account that we have observed this duration only for a subset of patients.

To achieve the goals, we have to consider techniques for supervised and unsupervised learning beyond those considered in Chap. 5. The discussed analysis methods can be applied for time sequences as well as for time series. Special techniques for time series are not considered. For the application of time series analysis for temporal data mining, the interested reader is referred to [22] and to general monographs about time series, for example, [11].

The analytical goal of finding frequent event sets is also known as *market basket analysis* referring to the prominent example from online vendors stating that “customers who have bought product A did also buy product B.” The association of products A and B is established based on their co-occurrence in observed event sets. However, the application is not restricted to such business examples. In medical applications, we could be interested in co-occurrences of certain diagnoses and lab tests. In the case of event sequences, questions about event sets can be made more precise, because using the temporal ordering allows for finding patterns in event sequences. For example, by contrast to stating that patients with a certain diagnosis require a lab test, a pattern would state that *after* diagnosing the patient, he/she requires a lab test, followed by a transfer to the hospital.

These analysis techniques summarized in the overview define the organization of the chapter.

### Analysis Techniques for Temporal Data

- *Time warping* operates on time sequences and aims at finding a similarity measure for time sequences which can be used later on for segmentation and classification.
- *Response feature analysis* reduces the problem of segmentation and classification to a problem for cross-sectional data by defining characteristics of the time sequences for the process instances.
- *Event history analysis* uses regression models which allow the prediction of the duration up to an event.
- *Analysis of Markov chains* targets at understanding the structural behavior of state sequences and the clustering of state sequences.
- *Association analysis* is a well-known technique to derive behavioral rules from event sets.
- *Sequence/episode mining* targets at finding behavioral patterns in event sequences.

Section 6.2 explains time warping and introduces into response feature analysis, Sect. 6.3 treats models and analysis techniques for the prediction of durations, and Sect. 6.4 shows how one can use Markov chains as a model for temporal data

mining. This is followed by a discussion of analysis techniques that are based on event patterns such as association analysis and sequence/episode mining in Sects. 6.6 and 6.7. In the presentation, we will focus mainly on model formulation and the algorithms used to achieve the analytical goals. From understanding the goals as special cases of supervised and unsupervised learning follows that methodological considerations about model assessment, model selection, and evaluation as discussed in Chap. 5 also apply for the analysis techniques in this chapter.

The methods considered are only a selection of the many approaches toward temporal data mining. In particular, we do not explicate temporal knowledge representation and reasoning in databases. For a more database-oriented exposition to temporal data mining, see [21].

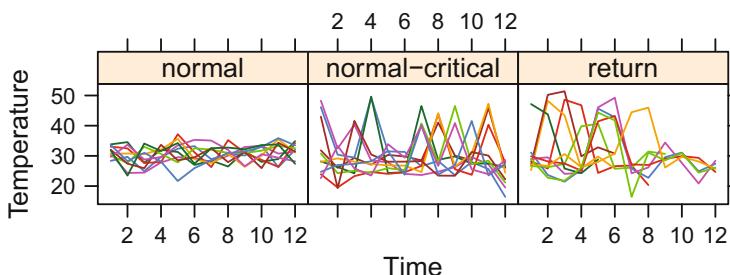
## 6.2 Classification and Clustering of Time Sequences

This section introduces dynamic time warping and response feature analysis as possible approaches towards the classification and segmentation of time sequences. To illustrate the approaches, we start with the visualization of data from the use cases.

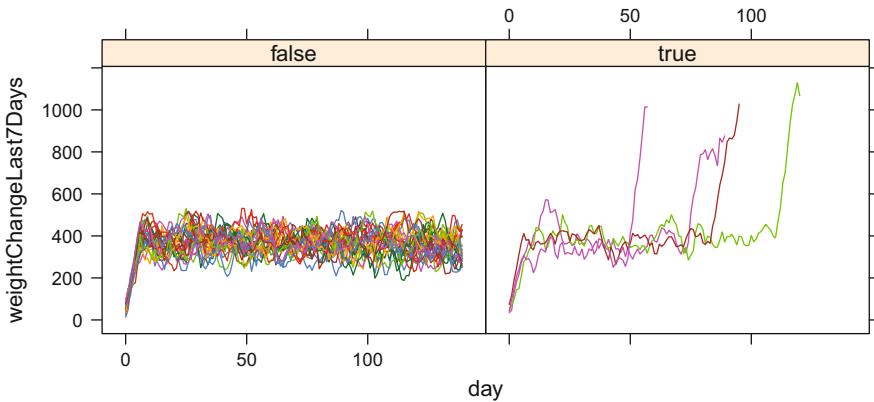
### Logistics Use Case: Container Temperature

In this use case, the main information about the process is in the time sequences of the temperature measurements. The visualization of 30 time sequences organized according to the temperature regimes is shown in Fig. 6.2.

The visualizations indicate the following characterization of the regimes: The normal regime is characterized by the fact that the temperature is always below 40 °C. The normal-critical regime shows spikes in temperature around 40 °C at different times, and the return regime shows consecutive time intervals with high temperatures around 40 °C. Moreover, due to early return, the time sequences have different lengths in case of return. The different time points for the occurrence of



**Fig. 6.2** Temperature regimes in different groups in the logistics use case (R package `lattice`)



**Fig. 6.3** Development of weight change for persons without preeclampsia (*left*) and with pre-eclampsia (*right*) (R package `lattice`)

high temperature can be interpreted as a temporal distortion of the structure of the time series. Hence, it seems reasonable to align the time sequences in order to make them more similar within the groups and more distinct between groups.

The second scenario is taken from the pre-eclampsia use case.

**Preeclampsia Use Case: Time sequences for Weight Change**  
 In this use case, the decision about hospitalization depends on the development of the measurements of weight, systolic and diastolic blood pressure, and proteinuria. Figure 6.3 shows the weight change development for 4 patients who had to go to the hospital due to pre-eclampsia and 26 patients who had no complication.

From the figure, we get the impression that there is a rather high variability in the weight change of persons without a complication during pregnancy. With respect to the increase in weight change, the right panel leads to the conjecture that in the case of pre-eclampsia, there is a strong increase in weight change in the later phase of pregnancy. Hence, it seems reasonable to use change points in the slopes as an indicator for possible pre-eclampsia.

The basic steps of the analysis are summarized in the following template which resembles the structure of the templates for supervised and unsupervised learning. Both modeling approaches have a number of tuning parameters. These have to be chosen according to the assessment of the models using mainly visualization and descriptive techniques.

**Template: Segmentation and Classification  
of Time Sequences**

- **Relevant Business and Data:** Customer behavior represented as time sequences for the process instances
- **Analytical Goals:**
  - Classification of a new time sequence into one of the possible classes
  - Segmentation of time sequences according to their structural similarity
- **Modeling Task:** The use of visualization techniques for the time sequences of the process instances is recommended to determine which of the following approaches seems more useful for the analytical goal:
  - *Time warping* for defining distances between time sequences
  - *Response features* for obtaining variables based on statistical techniques
- **Analysis Task:**
  - *Splitting Data:* If possible, split the data randomly in one set for training and one set for validation
  - *Model Estimation:* Estimate the optimal warping path or the envisaged response features
  - *Model Assessment:* Assess the quality of the model by using techniques discussed in Chap. 5
  - *Model Selection:* Select a model based on different specification of the warping algorithm or based on the chosen response features
  - use the results of model estimation for segmentation or classification
- **Evaluation and Reporting Task:** Evaluate the results of segmentation or classification either with test data or by using cross validation

The common strategy in both modeling tasks is the transformation of the learning problem for temporal data into a learning problem for cross-sectional data. This allows application of the ideas about supervised and unsupervised learning in Chap. 5 to temporal data. As we will show for modeling and analysis, the support provided by visualization methods is of utmost importance.

### **6.2.1 Segmentation and Classification Using Time Warping**

Time warping starts with data representing time sequences for a number of process instances, and the analytical goals are either the segmentation or classification of time sequences. The method is useful if the time sequences show a rather similar structural behavior, but the structure is blurred by temporal distortion in a way as shown in the logistics use case in Fig. 6.2. In agreement with Definition 6.1,

we consider only real-valued time sequences. Further, note that we need no exact knowledge of time. It is sufficient that the indices of the time sequences correspond to the temporal order of the observations.

A well-known algorithm for computing similarities of time sequences is *dynamic time warping*. In the exposition of the method, we follow [10] and [19]. The basic idea behind dynamic time warping is to stretch and compress the time sequences in such a way that the distance between the two sequences is minimized. Distance calculation for two time sequences is based on the distances between the elements of the time sequence. Given two time sequences  $x = (x_1, x_2, \dots, x_N)$  and  $y = (y_1, y_2, \dots, y_M)$ , we can define the distance for two indices by

$$d(i, j) = |x_i - y_j|. \quad (6.1)$$

In this definition of the distance, we assume that the values of the time sequences are real numbers. In the case of other values, one has to define the distance in an appropriate way (cf. Sect. 5.4). By using the distance matrix between the elements, we define a warping path as follows:

**Definition 6.3 (Warping Path)** Given two time sequences  $\mathbf{x} = (x_1, x_2, \dots, x_N)$  and  $\mathbf{y} = (y_1, y_2, \dots, y_M)$ , a warping path is a sequence  $p = (p_1, p_2, \dots, p_L)$  of index pairs  $p_\ell = (i_\ell, j_\ell)$  satisfying the following conditions:

- a) Boundary conditions:  $p_1 = (1, 1)$  and  $p_L = (N, M)$
- b) Monotonicity condition:  $i_1 \leq i_2 \leq \dots \leq i_L$  and  $j_1 \leq j_2 \leq \dots \leq j_L$
- c) Step-size condition:  $p_{\ell+1} - p_\ell \in \{(1, 0), (0, 1), (1, 1)\}$ .

The cost of the warping path is defined by

$$D_p = \sum_{\ell=1}^L d((i_\ell, j_\ell)) \quad (6.2)$$

The boundary condition implies that we map the start points and the end points of the two time sequences. The monotonicity condition guarantees that the order of the sequences is preserved. The step-size condition is sometimes called *symmetric step size* and ensures that we increase the value of the index in each step at least for one time series to the next one. This implies that multiple alignments of indices are possible. In some definitions, other step-size conditions are used (cf. [10]).

Based on the cost of the warping path, we can now define an optimal warping path by solving the following optimization problem:

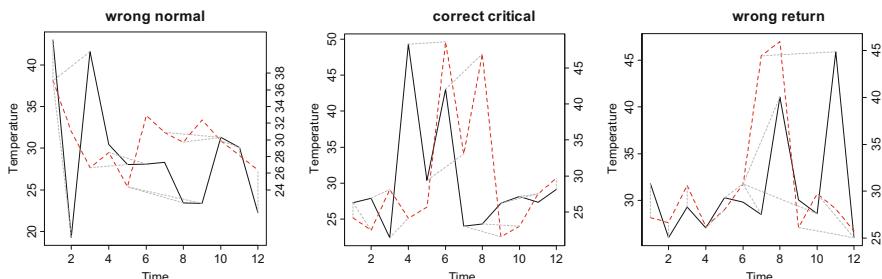
$$D(\mathbf{x}, \mathbf{y}) = \min_{p \in \mathcal{P}} D_p. \quad (6.3)$$

Here,  $\mathcal{P}$  denotes the set of all possible warping paths. The minimization problem can be solved efficiently using dynamic programming. Details may be found in [19].

As soon as we have computed a solution of dynamic warping for all pairs, we can interpret the costs of the optimal warping as a distance matrix between the time sequences. This distance matrix is now input for classification or segmentation algorithms. In the case of segmentation, any clustering method described in Sect. 5.4 can be used. For classification, the method of 1-nearest neighbors (1-NN) can be applied immediately (cf. Sect. 5.3.4). Time warping together with 1-NN classification is considered as one of the best general approaches for classification if there is no additional knowledge about time sequences. For more information and specific methods based on time warping, see [18]. Let us demonstrate the application of the method for the data in the logistics use case.

#### Logistics Use Case: Time Warping

Dynamic time warping was used for the calculation of the distance matrix between the 100 time sequences in the logistics use case. The resulting distance matrix defines the input for hierarchical clustering. Different methods were investigated and it turned out that the Ward method produced the best results. A solution with 3 clusters found the first cluster with 50 normal correctly classified cases, the second cluster with 5 normal-critical cases, and 5 return cases and the third cluster with 40 cases comprising 25 correctly classified return cases and 15 normal-critical cases. This shows that the differentiation between critical cases and return cases is not easy. An application of 1-NN classification showed similar problems with the normal-critical cases. The normal cases and the return cases were classified correctly, but only 8 normal-critical cases were identified. Eleven normal-critical cases were classified as return cases, and 1 normal-critical case was identified as a normal case. Figure 6.4 shows the result of dynamic time warping and classification for three misclassified normal-critical cases. The dashed lines show the time sequence which is the closest one, and the dotted lines indicate how the points are matched. In the wrong normal classification (left panel), it is obviously difficult to distinguish between the peaks, which are of almost the same height. In the wrong return classification (right panel), the last peak is mapped to the two consecutive peaks. From a practical point of view, it seems reasonable that a decision about return is not always easy in such cases.



**Fig. 6.4** Time warping for three sequences with the closest elements (R package dtw)

Details of the results and the predictive power can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

### 6.2.2 Segmentation and Classification Using Response Features

Response feature analysis is an example for the application of adaptive representation methods of the time sequences. The basic idea of this method is to deduce for each process instance from the observed time sequence a vector of characteristic features. These features define cross-sectional data for the process instances. In addition, the analytical goals of classification and segmentation are mapped to a corresponding analytical goal for cross-sectional data, which can be analyzed by methods considered in Chap. 5. In the following, we will briefly discuss three possible strategies for feature extraction.

#### Features Based on Frequency Distributions

Characteristic features may be based on properties of the frequency distribution of the time sequences such as the mean, median, quantiles, or variances. Derivation of such features has already been discussed in Sect. 4.4.4.

#### Features Based on Regression Models

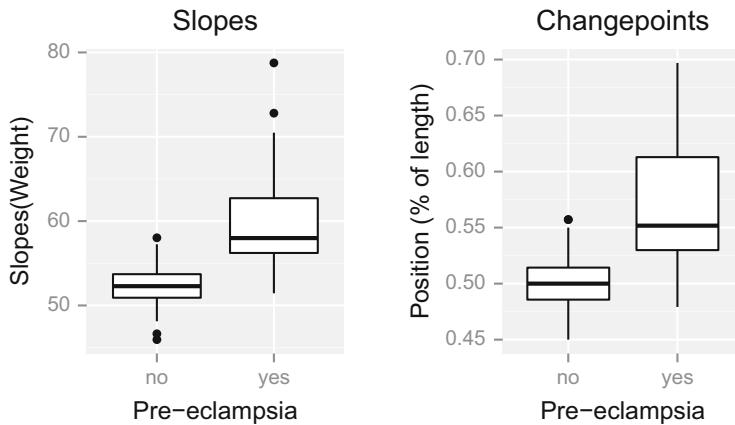
Another approach is the definition of a regression model for each observed time sequence instance by

$$Y_i(t) = f(t, X_i) + \varepsilon_i(t), \quad 1 \leq i \leq N. \quad (6.4)$$

The function  $f(t, X_i)$  describes the mean of the time sequence, and  $\varepsilon_i(t)$  represents the not-explained part of the state variable at time  $t$ . Besides time as explanatory variable, one can include additional explanatory variables  $X$ . The extracted features are based on the properties of the regression function for the mean of the time sequences. In the case of parametric regression models, the candidates for features are the coefficient of the regression function. In the case of nonparametric regression, possible response features are the maximum and the minimum of the function or the duration of exceedance of a certain threshold.

#### Features Based on Change Points

Sometimes, one regression function for the entire time sequence is not an appropriate model. For example, for the time sequences shown in Fig. 6.3, the right panel indicates that there is a change in the behavior in the time sequences at



**Fig. 6.5** Slopes of the time sequences of weight (left) and the relative position of change points (right) (R package `ggplot2`)

later time points, whereas in the left panel, such a change is not visible. In such cases, change points of the regression model can define an interesting response feature. Detecting change points in the behavior of a time sequence is an intensively investigated problem, in particular in economic time series analysis, and there exist a number of methods. A survey may be found in [23]. One basic approach for detecting such change points is the definition of a regression model for the time sequences and calculate the residuals of the regression model. If there are no change points, the residuals should show random fluctuation around zero. Any systematic deviation from this behavior is an indicator for a change point. By using statistical tests for the residuals, one can test hypotheses about a change point in the time sequence.

#### Preeclampsia Use Case: Response Feature Analysis

For identifying persons with the complication, we tried to extract features from the time sequences for weight, proteinuria, and blood pressure. Feature extraction based on the distribution has already been considered in Sect. 4.4.4. The 0.95 quantile for the time sequences for proteinuria is shown in Fig. 4.16. This figure indicates that the 0.95 quantile is a promising response feature. Feature extraction based on regression was applied for all variables. The left panel in Fig. 6.5 shows a boxplot of slopes in the two groups for the regression model

$$\text{weight} = \beta_0 + \beta_1 \text{day}$$

applied for all time sequences. Although the regression model for the group with preeclampsia is not correct, the figure indicates that the slopes may be a reasonable response feature for classification of the two groups.

For the demonstration of feature extraction based on change points, we use the residuals of the sequences for weight. For normalizing the position of the change points for time sequences with different lengths, the relative position of the change point was calculated. The right panel in Fig. 6.5 shows a boxplot for the change points in the different groups.

For these response features, different classification algorithms were applied. Classification trees with tenfold cross-validation generated a simple decision tree using only the change points of the variable weight. Alternatives for the splits were the slopes of the weight variable. From the 275 time series of persons with normal pregnancy, three were misclassified. An application of boosting and support vector machines produced similar results.

Details of the results and the predictive power can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

The method for the detection of change points can be extended to identification of multiple change points. Details and formal derivation may be found in [14].

We conclude this section with a remark about more advanced approaches towards regression models for time sequences. In the case of segmentation, the definition of cluster representatives is often of interest. For example, in the pre-eclampsia use case, we could be interested in a model for weight of all cases with normal pregnancy. Such a model can be estimated using so-called *mixed models*. Starting from a regression model  $\text{weight} = \beta_0 + \beta_1 \text{day}$  for the behavior of the state variable, random components  $\gamma_0$  and  $\gamma_1$  are introduced for capturing the individual variations of the slope and the intercept. This leads to the model

$$\text{weight}(\text{day}) = \beta_0 + \beta_1 \text{day} + \gamma_0 + \gamma_1 \text{day} + \varepsilon. \quad (6.5)$$

Here,  $\gamma_0$  and  $\gamma_1$  are random variables from a normal distribution with means 0, variances  $\sigma_0^2$  and  $\sigma_1^2$ , and independent from the errors  $\varepsilon$ . The obvious advantage of the model is that we have to estimate only four parameters,  $\beta_0, \beta_1, \sigma_0$ , and  $\sigma_1$ , instead of parameters for each time sequence. The estimation of the parameters of the mixed model can be based either on maximum likelihood estimation or on restricted maximum likelihood estimation (REML). From a computational point of view, the procedures are rather complex. Furthermore, one can apply the general methods for model selection and for prediction. A short application-oriented exposition using R for analysis may be found in Chapters 10 and 11 in [8].

### 6.2.3 Summary: Classification and Clustering of Time Sequences

Application of supervised and unsupervised learning methods for time sequences requires an appropriate representation of the time sequences. These representation methods are known as feature extraction. We considered one approach based on similarity measures and another based on response feature analysis.

Similarity measures apply the time warping algorithm for calculation of the distance between two time sequences. This algorithm can handle time sequences with arbitrary temporal structure and calculates a transformation of the time points which makes two sequences as similar as possible. The similarity measure allows the application of distance-based cluster methods. For classification, the K-nearest-neighbor classification is a frequently used tool.

Response feature analysis transforms the learning problem for the time sequence into a learning problem for cross-sectional data. Application of different methods for the calculation of response features was discussed, and the application of the methods for supervised and unsupervised learning of Chap. 5 was demonstrated.

## 6.3 Time-to-Event Analysis

In Chap. 5, we analyzed customer behavior according to some explanatory variables without taking into account the tenure of the relationship. In this section, we discuss models allowing the prediction of time until a certain event. In business applications, such events may be that a customer quits his/her relationship with a company, or in medical applications, interesting events may be the duration of a certain disease episode. Besides the notion time-to-event analysis, the terms *event history analysis* and *survival analysis* are used. The main problem in the analysis is the fact that we have to deal with so-called *censored data*. More precisely, we will consider right censored data. This means that we can record the time when a certain state starts for all instances, but we have complete information about the duration of the state only for that fraction of cases where the event has already occurred. The goal is the description of data and models which allow the prediction of the duration depending on attributes of the process instances.

We consider here only the basic concepts and methods for time-to-event analysis. More elaborated models, for example, time-varying explanatory variables, may be found in textbooks on the topic. A readable textbook is [7] which emphasizes data analysis with R.

**Template: Time to Event Analysis**

- **Relevant Business and Data:** Customer behavior represented by cross-sectional data and time sequences containing censored information about a terminal event
- **Analytical Goals:** Predict the duration up to the event for the censored time sequences from the uncensored data
- **Modeling Tasks:**
  - Definition of a survival table
  - Definition of a Cox regression model for the time to event
- **Analysis Tasks:**
  - Estimate the time up to the event using the Kaplan–Meier estimate
  - Estimation of the coefficients in the Cox regression model
- **Evaluation and Reporting Task:** Evaluate the results using a method for the evaluation of regression (cf. Sect. 5.1)

For the description and modeling of duration, the important concepts are the *survival function* and the *hazard function*. Duration up to a certain event is understood as a random variable and modeled by a probability distribution. We denote by  $T$  the variable measuring the time up to the event. The probability that the event occurs before time  $t$  is defined by the distribution function  $F(t) = P(T \leq t)$ . The survival function is defined as the probability that the duration of a customer is larger than  $t$ , i.e.,

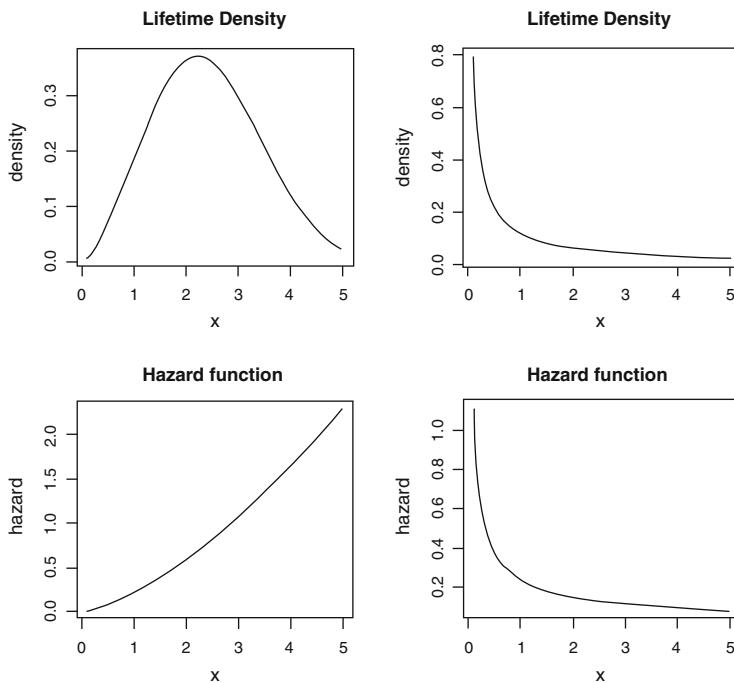
$$S(t) = 1 - F(t). \quad (6.6)$$

From the definition, it is obvious that the survival function is a decreasing function in time. The mean of the survival function is the expected survival time.

The second useful concept is the hazard, which is defined as the likelihood that the event takes place at time  $t$ , given that the event has not occurred up to time  $t$ . Formally, the hazard is related to the probability distribution of the duration according to the formula

$$h(t) = \frac{p(t)}{1 - F(t)}. \quad (6.7)$$

In most cases, the hazard function is used for describing the behavior. Depending on the domain problem, different structures for the hazard function can be defined, for example, hazard may increase or decrease in time. A convenient family of distributions for such problems is the family of *Weibull distributions* with distribution function  $F(x) = 1 - \exp[-(\alpha x)^\beta]$ . The parameters allow the adaptation to different



**Fig. 6.6** Densities and hazard functions of lifetime distributions (R graphics)

kinds of hazards. Figure 6.6 shows two Weibull distributions with increasing and decreasing hazard function.

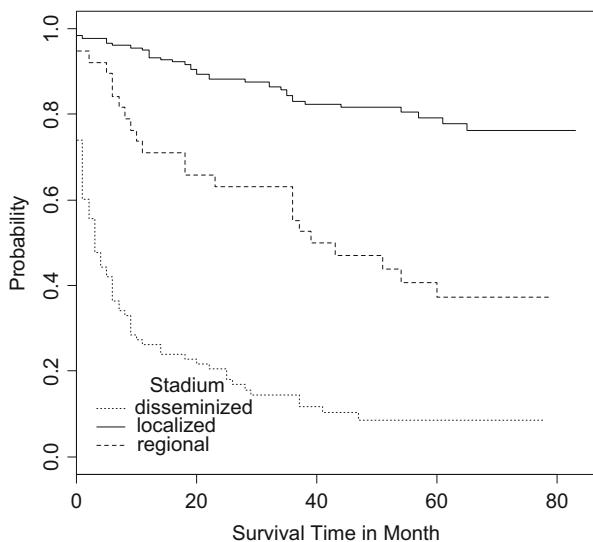
For the description of censored data, a life table is the standard method. The rows of the table are defined by predefined time intervals and characterized by the end point of the interval. In each row, there are three basic entries defining the columns of the table: the number of units entering the time interval, the number of units for which the event occurred in the time interval, and the value of the survival function at the end point of the interval. Additionally, other columns can be specified, for example, a confidence region for the estimate of the survival function. The estimation of the survival function is usually based on the *Kaplan–Meier estimate*. Let us demonstrate these ideas in the context of the EBMC<sup>2</sup> use case.

#### EBMC<sup>2</sup> Use Case: Life Table for Melanoma Patients

In order to understand the duration of melanoma until death, 305 patients who have been registered with malignant skin melanoma from 2006 to 2010 are considered. For all persons, the variables diagnosis time of malignant melanoma, age at diagnosis, sex, and stadium with three values, localized, regional, and disseminated, are known. Furthermore, the occurrence of other types of tumors for the patients is recorded. Overall 137

**Table 6.1** Life table of patients with malignant melanoma of the skin

Year	n.risk	n.event	survival	std.err	lower 95 % CI	upper 95 % CI
0	305	69	0.774	0.0240	0.728	0.822
1	236	23	0.698	0.0263	0.649	0.752
2	213	19	0.636	0.0275	0.584	0.692
3	174	16	0.578	0.0286	0.524	0.637
4	136	6	0.552	0.0292	0.498	0.612
5	86	4	0.526	0.0305	0.470	0.590

**Fig. 6.7** Survival times for patients with different stadiums (R package *survival*)

events occurred for the 305 patients in the observed time period. A life table of the persons using years as time resolution is shown in Table 6.1.

Details of the results can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

For the visualization of the time-to-event data, the survival time can be plotted for groups of the population defined by some factors. Figure 6.7 shows the survival functions for the three different values of stadium. As one can see, disseminated cases have the worst prognosis for survival time and localized cases the best.

For the prediction of the time to the event, we need a model which allows the specification of the hazard in dependence of some explanatory variables taking into account censoring of the data. The most prominent model used in practice is *Cox regression*, also known as *proportional hazard model*. The main idea of the model is the definition of a time-dependent *baseline hazard* for all cases. This baseline hazard is modified according to the explanatory variables of the cases. This idea is

made precise by the formula

$$h(t) = h_0(t)\exp(\beta_1x_1 + \beta_2x_2 + \cdots \beta_kx_k). \quad (6.8)$$

The formula justifies the name proportional hazard, because the ratio of the risks for two cases is constant over time.

Similar to logistic regression, the model defines a linear dependence for the logarithm of the hazard in the explanatory variables. This allows the following interpretation of the parameters in the model: For a quantitative explanatory variable  $X_j$ , the change of relative risk when changing the value from  $x$  to  $x + 1$  is given by  $\exp(\beta_j)$ . For a dummy variable representing a factor level,  $\exp(\beta_j)$  measures the change of relative risk for the factor level compared to the reference factor level.

**EBMC<sup>2</sup> Use Case: Cox Regression for Melanoma Patients**  
 For the data of the 305 patients described above, different specifications of the Cox model were computed. It turned out that the best model uses age at diagnosis and stadium as explanatory variables. The risk factor for age was 1.03 per year. This shows that hazard increases slightly with age. Compared to the reference category of disseminated stadium, the risk decreases for localized stadium by a factor 0.071 and for regional stadium by a factor 0.244. This is in correspondence to the survival curves shown in Fig. 6.7.

Besides the estimation of the model, one can use tools for model diagnostics. Of particular interest is the question whether the assumption of proportional hazard is correct. In the case of categorical explanatory variables, a simple visual inspection of the survival times for the different categories can be done. The curves should not cross for different categories as it is the case in Fig. 6.7.

## 6.4 Analysis of Markov Chains

For data represented as state or event sequences with known or unknown time stamps, a number of analytical goals are of interest in all three business perspectives. In the production perspective, one may be interested in finding clusters of event sequences in order to optimize services. In the organization or customer perspective, understanding the behavior in communication and cooperation networks is frequently of interest.

The basic model for these analytical goals are homogeneous Markov chains with a finite number of states introduced as probabilistic model structure in Sect. 2.4.4. As explained in Chap. 2, the interpretation is that starting from an initial state  $S_0 = s_i$ , a sequence of variables  $\langle S_0, S_1, \dots, S_T \rangle$  describes the process instances. Each variable can take values from a set  $\mathcal{S} = \{s_1, s_2, \dots, s_K\}$  which can be interpreted either as states or as events. In Sect. 6.4.1, we will interpret the variables as state variables, and in Sect. 6.4.2, we will use the interpretation as events.

For homogeneous Markov chains, we can use two types of representation. The first is the probabilistic representation as a matrix  $P = (p_{i,j})$  defined by the transition probabilities from state  $s_i$  to state  $s_j$  in one time step. The matrix is a *stochastic matrix* which means that all entries are positive and the rows sum up to one. The transition matrix after  $n$  steps is denoted by  $P(n)$ . By using the *Chapman–Kolmogorov equations*,  $P(n)$  can be calculated by matrix multiplication, i.e.,  $P(n) = P^n$ . If we denote the initial probabilities for the possible states at  $t = 0$  by  $\mu_0^{(i)} = P(S_0 = s_i)$  and by  $\mu_n^{(i)}$  the probabilities of the states at time  $T = n$ , we can calculate the probabilities of the different states after  $n$  time steps by

$$\mu_n = \mu_0 \cdot P^n, \quad \mu_0 = (\mu_0^{(1)}, \mu_0^{(2)}, \dots, \mu_0^{(K)}). \quad (6.9)$$

The second representation for homogeneous Markov chains is a graphical representation. It is obtained by interpreting the matrix of transition probabilities as weighted adjacency matrix of a directed graph with nodes defined by the states of the process.

In this section, we will primarily discuss analysis methods based on estimation and clustering techniques. With respect to evaluation and reporting, the general considerations in Sects. 5.1 and 5.4 apply in the case of clustering. Regarding the estimation problems, we will discuss the evaluation in the context of the analysis methods. The following template summarizes the most important analysis steps.

**Template: Analysis of Markov Chains**

- **Relevant Business and Data:** Process instances represented as states or event sequences
- **Analytical Goals:**
  - Estimation of state transitions from exiting instances
  - Structural behavior of state transitions in the long run
  - Segmentation of sequences into groups
  - Segmentation of the states
- **Modeling Tasks:**
  - Definition of a stationary Markov chain for state transitions
- **Analysis Tasks:**
  - Estimation of transition probabilities
  - Estimate of a stable distribution
  - Cluster analysis for instances of state or event sequences
  - Cluster analysis of the states or events
- **Evaluation and Reporting Task:** Evaluation and reporting of the results in agreement with general considerations about evaluation of learning methods in Chap. 5

Section 6.4.1 presents analysis details for estimation problems and Sect. 6.4.2 the details for clustering. We conclude this section with a short description of hidden Markov chains which are a more general model useful in many applications.

### 6.4.1 Structural Analysis of Markov Chains

By the term structural analysis of Markov chains, we understand methods for solving estimation goals, i.e., finding transition probabilities in the long run or the estimation of transition probabilities from instances of state sequences. For solving these estimation problems, a classification of the states of a Markov chain with respect to the transition behavior plays an important role. Hence, we start with a classification of the states of a Markov chain, consider afterwards the estimation of the transition probabilities based on observed state sequences, and conclude with the estimation of the stable distribution.

#### Typology of States for Markov Chains

For understanding the typology of states, a representation of the homogeneous Markov chain as graph is useful. We say that a state  $s_i$  is *reachable* from a state  $s_j$  if there is a path from state  $s_i$  to state  $s_j$ . Obviously, an edge  $(s_i, s_j)$  defines a path of length 1, and we call such states *directly linked*. States  $s_i$  and  $s_j$  are called *connected* if  $s_i$  is reachable from  $s_j$  and  $s_j$  reachable from  $s_i$ . In the graph representation, connected states define a closed path, and the property of connectedness defines a partition of all states into classes of connected states.

A Markov chain is called *irreducible* if each state can be reached from any other state in finite time, i.e., all states belong to one class. Furthermore, we define a closed set of states as states which cannot be left as soon as we have reached the states. An *absorbing* state is a closed state not connected to any other state. For an absorbing state  $s_i$ , we have  $p_{ii} = 1$ .

Figure 6.8 illustrates the typology of states. The chain is not irreducible because from  $s_4$  no other state can be reached. There are three classes of connected states:  $\{s_1, s_2\}$ ,  $\{s_3\}$ , and  $\{s_4\}$ ; the classes  $\{s_1, s_2\}$  and  $\{s_4\}$  define closed classes and  $s_4$  is an absorbing state.

For the long-term behavior of the process defined by the transition probabilities, we need three additional definitions. A state is called *transient* if there is a positive probability of not returning into the state. A state is called *recurrent* if the probability of returning into the state is 1. In the case of irreducible Markov chains, all states are either recurrent or transient. For recurrent states, we can define the *period* as the largest common divisor of all times  $t$  for which  $p_{ii}(n) > 0$ . If the period of a state is 1, the state is called *aperiodic*. A Markov chain where all states are aperiodic is called an *ergodic Markov chain*. Figure 6.9a shows an ergodic Markov chain and Fig. 6.9b a Markov chain where each state has period 3.

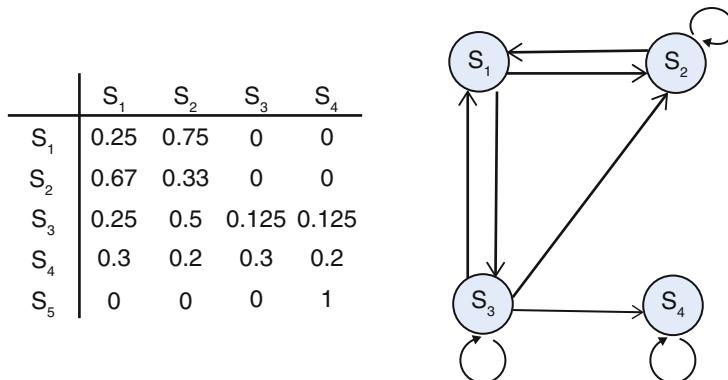


Fig. 6.8 Types of states in a Markov chain

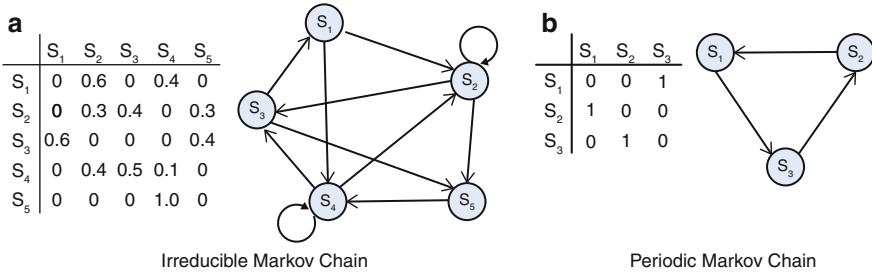


Fig. 6.9 Examples of an ergodic Markov chain (a) and a Markov chain with periodic states (b)

### Estimation of Transition Probabilities

Given  $N$  state sequences  $s_1, s_2, \dots, s_N$  of possibly different lengths, generated by a homogeneous Markov chain with  $K$  states  $s_1, s_2, \dots, s_K$ , the goal is the estimation of the matrix of transition probabilities  $p_{ij}$ . Obviously,  $p_{ij} \neq 0$  only for those transitions for which an edge between the vertices exists in the graph representation. Under the assumption that all transitions are generated independently, the distribution of the number of transitions from a state  $s_i$  to the directly linked states is a multinomial distribution. Given a number of state sequences, we denote by  $n_{ij}$  the observed number of one-step transitions from state  $s_i$  to state  $s_j$  and by  $n_i$  the observed number of occurrences of state  $s_i$ . An intuitive appealing method for the estimation of the transition probabilities is the maximum likelihood estimate defined by

$$\hat{p}_{ij} = \frac{n_{ij}}{n_i}. \quad (6.10)$$

A drawback of this estimate is that if transitions from state  $s_i$  to  $s_j$  are not observed in the data,  $\hat{p}_{ij} = 0$  although such transitions may be possible from

the structure of the Markov chain. In order to overcome such deficiencies, a Bayesian approach is frequently used. This means that a prior distribution for the transition probabilities is assumed, and the estimates of the transition probabilities are computed as means of the posterior distribution. A frequently used prior for the multinomial distribution is the *Dirichlet distribution* defined by the density

$$P(p_1, p_2, \dots, p_K) = C \prod p_i^{\alpha_i - 1}, \quad \alpha_i > 0. \quad (6.11)$$

The parameters  $\alpha_i$  are called concentrations. Means and variances of the components are given by

$$E[N_i] = \frac{\alpha_i}{\sum_{j=1}^K \alpha_j} = \frac{\alpha_i}{\alpha_0} \quad [N_i] = \frac{\alpha_i(\alpha_0 - \alpha_i)}{[\alpha_0^2(\alpha_0 + 1)]}. \quad (6.12)$$

These expressions can be interpreted in such a way that  $E[N_{ij}]$  reflects the prior belief of the researcher about the mean values of the transitions and that  $\alpha_0 = \sum \alpha_j$  measures how peaked the priors are around the means. A large value of  $\alpha_0$  indicates that there is higher confidence in the priors than for a small value of  $\alpha_0$ .

By using the prior distribution, one obtains the following estimates for the transition probabilities:

$$\tilde{p}_{ij} = \frac{n_{ij} + \alpha_i}{n_i + \alpha_0}. \quad (6.13)$$

Based on these estimates, one can make predictions of future states. An application of this method is the prediction of page requests by humans on the Internet. In such applications, one can choose the parameters  $\alpha_j$  according to the relation between the outgoing links of the pages. The choice of  $\alpha_0$  determines how many observations are necessary for a substantial change of the prior beliefs. Details may be found in [5].

### Stable Distribution of Markov Chains

An important issue in the analysis of Markov chains is the long-term behavior of the chain. The main question is whether there exists a stable distribution for the states and whether we can reach such a stable distribution independent from the initial distribution. By stable distribution, we mean a probability distribution  $\pi$  for the states fulfilling the equation  $\pi = \pi \cdot P$ , i.e., the distribution remains unchanged by transitions. In the case of irreducible aperiodic Markov chains, there exists a stable distribution, and this distribution is independent from the initial distribution of the states. This justifies the term ergodic Markov chain, because in general the term ergodic refers to the fact that the temporal iteration leads to the stable distribution of the states.

The stable distribution of an ergodic Markov chain can be approximated by the iteration of the transition matrix  $P$  as it is shown in Example 6.1.

*Example 6.1 (Stable Distribution for Markov Chains)*

Consider a homogeneous Markov chain with six states defined by the following matrix of transition probabilities:

$$\begin{array}{cccccc} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{matrix} & \left( \begin{array}{cccccc} 0 & 0.25 & 0.25 & 0 & 0.25 & 0.25 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0.33 & 0.33 & 0.34 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0.5 & 0 \end{array} \right) \end{array}.$$

Iterating the matrix gives the stable distribution for the states:

$$\pi = (0.2088, 0.1769, 0.2585, 0.3878, 0.0408, 0.0272).$$

After 20 iterations, the accuracy of the solution is of order  $10^{-4}$ , and after 30 iterations,  $10^{-9}$ .

Example 6.1 demonstrates the basic idea behind the well-known *page rank algorithm* [5] which is important for analyzing Web browsing. The pages of the Web define states of a Markov chain, and the edges are defined by outgoing links from the page. The number of outgoing links defines the *hubness* of a page. A user of the Web generates a state sequence of pages by choosing randomly a linked page. If the structure of the links between pages defines an irreducible aperiodic Markov chain, the transition probabilities between the pages generated by many users will follow the stable distribution for this Markov chain. This stable distribution for the sites can be interpreted as the importance rank of the page. Different methods have been proposed to ensure that the Markov chain is irreducible and aperiodic, for example, random choice at pages with no outgoing links.

In applications, a lot of refinement is necessary and computational effort caused by the size of the network needs additional considerations. One way is the well-known HITS model. HITS is an abbreviation for *Hyperlinked Induced Topic Search*. For answering queries, HITS does not only use page rank, but it builds also a subnet of pages which are relevant for the topic of the query. For building this subnet of pages, the algorithm considers outgoing links as well as incoming links to pages. Incoming links define the *authority* of a page. For details, see [5].

### 6.4.2 Cluster Analysis for Markov Chains

The clustering of Markov chains aims for finding groups of Markov chains with similar structure. Following the ideas proposed in [20], we will outline a cluster algorithm. The basic ingredients of the algorithm are the interpretation of an event sequence as a Markov chain and the assignment of the probability that an event sequence is generated by a Markov chain.

The interpretation of an event sequence as a Markov chain is based on the interpretation of the events as states and the definition of transition probabilities according to Eq.(6.10). For demonstration of the this idea, let us consider the following example in the context of the EBMC<sup>2</sup> use case:

**EBMC<sup>2</sup> Use Case: Markov Chain Associated to an Event Sequence**

Persons who have susceptible moles are asked to do preventive checkups in the hospital on a regular basis. If there is no change in the structure of the moles (event CN), the decision is to continue with the routine checkup. In case of suspicious changes (event CP), a histological diagnosis is done with two possible outcomes. In the case of negative histological results (event HN), the routine checkup continues; otherwise (event HP), the patients has to undergo surgery (event EX). If we add a start event and an end event, the event sequences are defined by the events  $\mathcal{S} = \{\text{CN}, \text{CP}, \text{HN}, \text{HP}, \text{EX}, \text{start}, \text{end}\}$ . A sequence

$\langle \text{start}, \text{CN}, \text{CN}, \text{CP}, \text{HN}, \text{CN}, \text{CP}, \text{HP}, \text{EX}, \text{end} \rangle$

generates a Markov chain with the following positive transition probabilities:

$$\begin{aligned} P(\text{CN}|\text{start}) &= 1 \\ P(\text{CN}|\text{CN}) &= 1/3, P(\text{CP}|\text{CN}) = 2/3 \\ P(\text{HN}|\text{CP}) &= P(\text{HP}|\text{CP}) = 1/2 \\ P(\text{CN}|\text{HN}) &= 1 \\ P(\text{EX}|\text{HP}) &= 1 \\ P(\text{end}|\text{EX}) &= 1 \end{aligned}$$

Given a Markov chain  $M$  with transition probabilities  $P_M$  and initial probability  $\pi_M$ , we can find the probability that an event sequence  $\mathbf{e} = (e_1, e_2, \dots, e_T)$  is generated by the Markov chain by

$$P(\mathbf{s}|M) = \pi(s_1) \cdot \prod P_M(s_i|s_k). \quad (6.14)$$

Obviously, this equation is only meaningful if the events in the sequence are a subset of the events defining the Markov chain  $M$ . Based on these probabilities, the

following algorithm can be used for clustering Markov chains, which resembles the structure of the  $k$ -means algorithm in Sect. 5.4.3:

---

**Algorithm 7:** Markov chain clustering
 

---

**Data:** Transition matrices  $P^{(i)}$ ,  $1 \leq i \leq n$  of event sequences; number of clusters  $K$ .

**Result:** Cluster solution for the transition matrices

- 1 **begin**
  - 2    Define an initial solution for the transition matrices of the cluster centers  $(P_1, P_2, \dots, P_K)$ ;
  - 3    Assign the transition matrices  $P^{(i)}$  of the event sequences to the cluster which leads to the highest probability according to equation (6.14);
  - 4    Compute new centers for the clusters according to the calculation of transition probabilities of equation (6.10);
  - 5    Repeat steps 2 and 3 as long as there is no significant change in the centers;
  - 6 **end**
- 

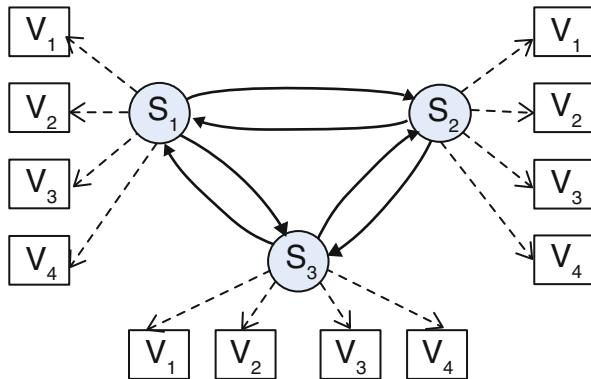
Besides the clustering of transition matrices, it is sometimes of interest to cluster the events or the states themselves. For example, in Markov chains, defined by the communication between persons involved in a business process, one would be interested in finding a subset of persons that shows many interactions. An algorithm for solving this problem was proposed in [24]. We briefly explain the basic idea of the algorithm: Its starting point is a network graph. For this graph, a transition matrix is generated which allows loops, i.e.,  $p_{ii} \neq 0$ . Next, in a so-called expansion step, the transition matrix for  $T$  time steps is generated. Here,  $T$  is a parameter which has to be set in advance. In a following inflation step, this transition matrix is modified in such a way that transitions with high probabilities are boosted and transitions with low probabilities are downgraded. Similar to the expansion step, the inflation step uses a parameter. Expansion and inflation operations are repeated until a steady matrix is obtained.

### 6.4.3 Generalization of the Basic Model

Markov models can be generalized in different ways. A generalization used in many applications are *hidden Markov chains*.

**Definition 6.4 (Hidden Markov Model)** A Hidden Markov model  $(\pi, P, B)$  is defined by the following components:

- a) A Markov process with  $K$  states  $s_1, s_2, \dots, s_K$  and a matrix of transition probabilities  $P = (p_{ij})$ .
- b) For each state, there exists an additional attribute  $V$  with values  $\{v_1, v_2, \dots, v_M\}$ , and for each  $v_m$ , the conditional probability  $b_{mi} = P(v_m | s_i)$  for the occurrence of the attribute  $v_m$  given state  $s_i$  is defined by a matrix  $B$ .
- c) An initial distribution  $\pi$  of the states.



**Fig. 6.10** Example of a hidden Markov chain

The idea behind the definition is that the observations of the system are generated by the Markov chain, but we can only observe a sequence of attributes. Figure 6.10 shows the structure of a hidden Markov model with three states and four attributes.

An example of a hidden Markov chain in a medical context could be that the states represent the health status of a person who is not directly observed, and only some attributes can be measured. Another example are mixtures of Markov models which assume that the data are generated by different Markov chains.

For hidden Markov models, three different types of analytical goals can be distinguished:

#### Analytical Goals for Hidden Markov Chains

1. *Evaluation problem:* Given a sequence of observations of the attributes and the model parameters, calculate the probability of an observed sequence. This can be done directly by applying Bayes' theorem. A more efficient way is the so-called *forward procedure*.
2. *Decoding problem:* Given a sequence of observations, we want to learn what sequence of states has generated the observations. This problem is solved by the so-called *Viterbi algorithm*, which is an application of the principle of dynamic programming.
3. *Learning problem:* Find  $P$ ,  $B$ , and  $\pi$  of the model from observation sequences. This problem can be seen as a special kind of a missing data problem and can be solved with the EM algorithm. In connection with the model, it is also known as the *Baum–Welch algorithm*.

A detailed exposition of the analysis of hidden Markov chains may be found in [6]. Here, we only point to an application of similar ideas in [9] that concern the analysis of workflows in case of process logs without identifying labels. Given a workflow model, all admissible realizations of the workflow are considered and identified by the corresponding sequence of events. These admissible sequences

define the states of the hidden Markov chain. The observed events of a workflow are the observable attributes of the system. The probabilities of occurrence of the attributes given a specific state (i.e., a realization of the workflow) are defined by the transition structure of the admissible sequence. If the event logs are not labeled, the problem can be understood as the analysis of a hidden Markov chain. The main difference is that the conditional probabilities of the observed attributes depend on the occurrence in the workflow sequence, which needs some modification in the evaluation step.

#### 6.4.4 Summary: Analysis of Markov Chains

Markov chains are a probabilistic model with a wide range of applications, provided the data are in the state view with a finite number of states. The basic analytical techniques for the analysis of the structure of a Markov chain were discussed. Two important questions are the estimation of the transition probabilities and the determination of the stable distribution of a homogeneous Markov chain. This distribution characterizes the long-term behavior of the Markov chain and is a basic tool for understanding the Web surfing behavior.

Furthermore, two applications of cluster analysis for Markov chains were delineated. For finding clusters of state sequences from different process instances, an algorithm was outlined. A second application of cluster analysis is finding a structure in the transition matrix.

Finally, hidden Markov chains were introduced as a generalization of the basic model. This model has numerous applications in different subject matter areas. The analytical goals for hidden Markov chains were delineated.

## 6.5 Association Analysis

Association analysis works on event sets. Typically, these event sets are produced by transactions such as purchase transactions in an online store or prescription transactions in a hospital. Of particular interest are the *items* that are associated with the transactions. Examples are the goods that were purchased or drugs that were prescribed. Example 6.2 describes two example event sets with the associated items, i.e., an online store, where the items nail and hammer were purchased, and a hospital setting, where items Aspirin, Marcumar, and Paracetamol were prescribed.

*Example 6.2 (Events Sets for Association Analysis)*

- Online Store—Transaction 1:
  - Event 1 = Purchase Nail
  - Event 2 = Purchase Hammer

- Hospital—Transaction 1:
  - Event 1=Prescribe Aspirin
  - Event 2=Prescribe Marcumar
- Hospital—Transaction 2:
  - Event 1=Prescribe Aspirin
  - Event 2=Prescribe Marcumar
- Hospital—Transaction 3:
  - Event 1=Prescribe Aspirin
  - Event 2=Prescribe Paracetamol
- Hospital—Transaction 4:
  - Event 1=Prescribe Aspirin
  - Event 2=Prescribe Marcumar

The goal is to find association rules in the form of “ $A \implies B$ ,” i.e., the occurrence of A implies the occurrence of B. We denote A as the *antecedent* and B as the *consequent* of the rule.

In our examples, the following rules could be of interest:

- Online store:  $R_O : \text{Purchase Nail} \implies \text{Purchase Hammer}$
- Hospital:  $R_H : \text{Prescribe Aspirin} \implies \text{Prescribe Marcumar}$

The interpretation of both rules is different. For rule  $R_O$ , the store manager can now think of offering hammers to every customer that has bought a nail. By contrast, rule  $R_H$  constitutes a means to detect adverse drug events as Aspirin and Marcumar should not be prescribed together.

In the following, we will explain association analysis along the main points of the general structure of the analysis templates. Due to the fact that association analysis is not based on models but on finding patterns of event sequences, we omit the modeling task in the description. The evaluation task in association analysis is also omitted, since it consists in the interpretation of the mined associations in the context of the business.

Definition 6.5 summarizes the input data for association analysis following the original paper [3].

**Definition 6.5 (Input for Association Analysis)** The input data for association analysis are as follows:

- $\mathcal{I} := \{i_1, \dots, i_n\}$  defines a set of items.
- $T$  defines a set of transactions; each transaction  $t \in T$  is defined as a vector  $t := < t[1], \dots, t[n] >$  with  $t[j] = 1$  if item  $i_j$  is associated with  $t$  and  $t[j] = 0$  otherwise.
- $X \subseteq \mathcal{I}$  denotes the item set of interest, i.e., we are looking for rules  $X \implies I_j$  with  $I_j \in \mathcal{I}$  and  $I_j \not\subseteq X$ .
- We say that a transaction  $t \in T$  satisfies  $X$  if  $\forall x \in X : t[x] = 1$ .

**Table 6.2** Formats for the representation of transactions

Transaction	Aspirin	Marcumar	Paracetamol
$t_1 = <1,1,0>$	1	1	0
$t_2 = <1,1,0>$	1	1	0
$t_3 = <1,0,1>$	1	0	1
$t_4 = <1,1,0>$	1	1	0

The following example illustrates Definition 6.5:

*Example 6.3 (Adverse Drug Events)* Take the set of transactions  $T$  as provided in Example 6.2. The item set is given as  $\mathcal{I}_H = \{\text{Aspirin}, \text{Marcumar}, \text{Paracetamol}\}$ . The transaction vectors using the order Aspirin, Marcumar, and Paracetamol are shown in Table 6.2 on the left side.

Alternatively, a table format can be used, e.g., for tool imports, as shown on the right side. All transactions  $t_1, \dots, t_4$  satisfy, for example, item set  $\{\text{Aspirin}\}$ .

The *analytical goal* in association analysis is finding association patterns with certain properties. Afterwards, these association patterns are formulated as rules which are of interest from a practical point of view. For the precise formulation of these goals, we first of all need measurements on how strong a given rule is supported within the transaction set. For this, we define the *confidence* of a rule  $A \implies B$ .

**Definition 6.6 (Confidence of an Association Rule)** Let  $T$  be a set of transactions,  $A \subseteq \mathcal{I}$  be an item set of interest, and  $B \in \mathcal{I}$  be an item set. Then the confidence  $c$  of rule  $r: A \implies B$  is defined as follows (based on [3]):

$$c(R, T) := \frac{|\{t \mid t \text{ satisfies } A \cup B\}|}{|A|}. \quad (6.15)$$

The confidence tells us about the rule strength [3] and can be used to evaluate rules of interest. For rule  $R_H = \text{Aspirin} \implies \text{Marcumar}$ , we obtain a confidence of  $c(R_H, T) = \frac{3}{4} = 0.75$ .

Another goal is to search for rules with a minimum confidence for a given transaction and item set, i.e., the rules of a certain strength are not known beforehand, but are mined from the transaction set. These rules can be arbitrary. However, certain restrictions on the rules are often known beforehand. These restrictions can be expressed by constraints on the rules. According to [3], two kinds of constraints can be of interest:

- *Syntactic constraints* imply some restriction on the antecedent  $A$  and/or consequent  $B$  of a rule  $A \implies B$ . An example would be the analysis questions: “Find all rules with a confidence of 0.5 where Aspirin is in the antecedent.”

- *Support constraints* refer to the support of a given rule within the transaction set  $T$ . The support  $s$  of a rule  $R: A \Rightarrow B$  in  $T$  can be calculated as follows:

$$s(R, T) := \frac{|\{t \mid t \text{ satisfies } \{A, B\}\}|}{|T|}. \quad (6.16)$$

In the adverse drug example, the support of  $R_H: \text{Aspirin} \Rightarrow \text{Marcumar}$  turns out as  $s(R_H, T) = 0.75$ .

Note that the support is independent of the rule structure. For a rule  $R_1: \{A, B\} \Rightarrow C$  the support would turn out equal as for rule  $R_2: A \Rightarrow \{B, C\}$ . Here, we can see that the support refers to the item set rather than to the rule. The relation between support and confidence for a rule  $R: A \Rightarrow B$  is given as follows:

$$c(R, T) = \frac{s(R, T)}{s(A \Rightarrow, T)}. \quad (6.17)$$

How can association rules be found in a transaction set? Basically, the *analysis task* consists of two steps (cf. [3]):

### Main Steps in Association Analysis

1. *Finding large item sets*: A minimal support is defined. Then, the task is to find all item sets for which their support is above the minimal support, so-called *large item sets*. The search might be confined by syntactical constraints. Finding large item sets can quickly become expensive as, in principle, the support of all possible combinations of item sets in the transaction set has to be determined. Optimization techniques, such as pruning, can be applied.
2. *Discover rules within large item sets based on confidence and syntactical constraints*. Given a set of large item sets, all possible rules with elements of this large item set are determined. We choose all those rules that exceed a given confidence. The result is a set of rules with a minimal support and minimal confidence that possibly respect certain syntactical constraints.

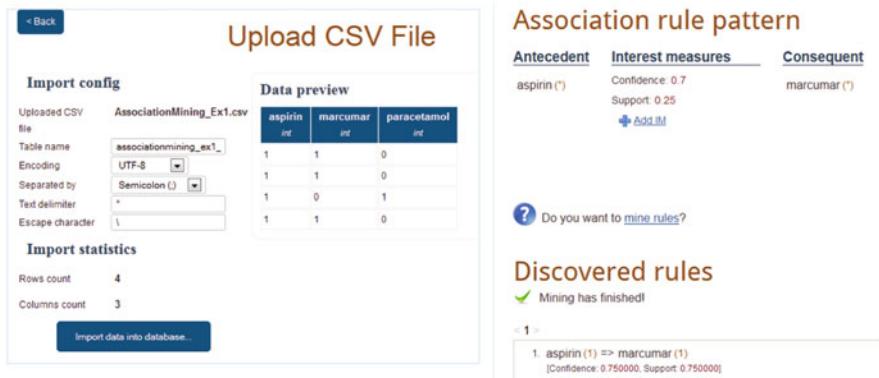
Let us illustrate these steps by the following example:

*Example 6.4 (Finding Large Item Sets for Adverse Drug Event Set)* In our example, assume a minimal support of 0.2. We have to determine the support of item sets (A: Aspirin, M: Marcumar, P: Paracetamol):

$$\begin{aligned} s(\{A\}, T) &= 1, s(\{M\}, T) = 0.75, s(\{P\}, T) = 0.25, s(\{A, M\}, T) = 0.75, \\ s(\{A, P\}, T) &= 0.25, s(\{M, P\}, T) = 0, s(\{A, M, P\}, T) = 0. \end{aligned}$$

Consequently, we obtain as large item sets:  $\{A\}$ ,  $\{M\}$ ,  $\{P\}$ ,  $\{A, M\}$ ,  $\{A, P\}$ . Respecting the additional syntactical constraints, “antecedent must contain Aspirin” and “consequent must not equal emptyset” would reduce the large item set to:

$$\text{LargeItemSet} = \{\{A, M\}, \{A, P\}\}.$$



**Fig. 6.11** Adverse drug event example (using EasyMiner)

From this set, we form all possible rules with associated confidences:

- $R_1 : A \implies M$  with  $c(r_1, T) = 0.75$ .
- $R_2 : A \implies P$  with  $c(r_2, T) = 0.25$ .

By calculating the confidence of these rules and demanding for a minimal confidence of 0.5, the result is rule  $R_1$ .

The above example has been implemented using the Web-based tool EasyMiner.<sup>1</sup> Figure 6.11 confirms the manually calculated results for rule  $\text{Aspirin} \implies \text{Marcumar}$ .

As discussed before, calculating large item sets can become complex. Hence, several variations of the basic algorithm were developed [1]. The basic idea is to reduce the candidates for large item sets. This is achieved by only considering those item sets of size  $k$ , i.e., containing  $k$  items, as candidates for large item sets that built upon a large item set of size  $k - 1$ . In our example, we would consider candidate  $\{A, M\}$  of size 2 since it builds upon large item sets of size 1, i.e.,  $\{A\}$ ,  $\{M\}$ . The relations of “builds upon” can be expressed by a lattice [12]. The item sets that are not large are pruned.

## 6.6 Sequence Mining

In Sect. 6.5, the basic technique for finding association rules in a transaction set was described. In this section, we will explain how association rules can be extended by order relations as motivated in Example 6.5.

<sup>1</sup><http://easyminer.eu>.

*Example 6.5 (Adverse Drug Data for Sequence Mining)* Assume the following transaction set (table on top). Each row describes one transaction together with a set of prescribed drugs, a time stamp when the transaction took place, and the concerned patient. The table at the bottom presents the same data aggregated for the patients (cf. example presented in [2]).

Aspirin	BetaBlock	Ibu	Antibiotics	Time stamp	Patient
1	0	0	0	10.10.2013	P1
1	0	1	0	12.10.2013	P2
0	0	0	1	13.10.2013	P2
0	1	0	0	14.10.2013	P1
1	0	0	0	15.10.2013	P3
0	0	1	0	16.10.2013	P3
1	0	1	0	17.10.2013	P4
0	1	0	1	18.10.2013	P4

Patient ID	Item set	Sequence
P1	{Aspirin, BetaBlock}	< {Aspirin}, {BetaBlock} >
P2	{Aspirin, Ibu, AntiBiotics}	< {Aspirin}, {Ibu}, {AntiBiotics} >
P3	{Aspirin, Ibu}	< {Aspirin}, {Ibu} >
P4	{Aspirin, Ibu, BetaBlock,AntiBiotics}	< {Aspirin, Ibu}, {BetaBlock,AntiBiotics} >

So far, we have mined association rules such as

$$\text{Aspirin} \implies \{\text{Ibu, AntiBiotics}\}$$

meaning that patients taking Aspirin also take Ibu and Antibiotics with a support of 0.5 and a confidence of 0.5.

What we do not know from the above result is whether Ibu and Antibiotics are prescribed in a certain order after the administration of Aspirin. It is possible that first Ibu is prescribed and then Antibiotics or vice versa. The order of taking the drugs might cause different effects and therefore plays a crucial role.

Hence, we are interested finding rules of the form “if Aspirin is taken, then Ibu is taken followed by Antibiotics.” This is what *sequence mining* [2] is about. Note that sequences may contain sets of unordered items, e.g., “after taking Aspirin and Parkemed, Ibu is taken followed by Paracetamol and Antibiotics.” For the explanation of sequence mining, we use the same structure as in association analysis.

The data structure for sequence mining is specified in the following definition:

**Definition 6.7 (Input for Sequence Mining)** According to [2], the input data for sequence mining is as follows:

- $\mathcal{I} := \{i_1, \dots, i_n\}$  defines the set of items.
- $T$  defines a set of transactions; note that to each transaction, a time stamp is assigned.
- $S := < s_1, \dots, s_k >$  denotes a sequence of item sets, i.e.,  $i < j$  means that  $s_i$  occurred before  $s_j$ .
- A sequence  $S$  is contained in another sequence  $S'$ —denoted by  $S \prec S'$ —if  $\forall$  item sets  $s \in S: \exists s' \in S'$  with  $s \subseteq s'$ .
- The set of transactions  $T$  is called *customer sequence* as it constitutes an ordered sequence of transactions referring to item sets, i.e., it constitutes an item set sequence itself.

The following example illustrates Definition 6.7.

*Example 6.6 (Example Sequences from Adverse Drug Scenario)* Example sequences would be

$$S_1 = <\{\text{Aspirin}\}, \{\text{BetaBlock}\}> \text{ and}$$

$$S_2 = <\{\text{Aspirin}, \text{Ibu}\}, \{\text{BetaBlock}, \text{AntiBiotics}\}>$$

where  $S_1$  is contained in  $S_2$  because

$$\{\text{Aspirin}\} \subset \{\text{Aspirin}, \text{Ibu}\} \text{ and } \{\text{BetaBlock}\} \subset \{\text{BetaBlock}, \text{AntiBiotics}\}.$$

The *analytical goal* of sequence mining is as follows: *Find the maximum sequences in the customer sequence with a user-defined minimum support.*

The *analysis task* splits in two subtasks, i.e., finding sequences with minimum support and out of these sequences finding the maximum ones. Sequences with minimum support are called large sequences. The support of a sequence  $S$  over a set of customers  $C$  is calculated by taking the number of customers who support  $S$  and divide it by the total number of customers. A customer  $c \in C$  supports  $S$  if  $S$  is contained in  $S_c$  where  $S_c$  denotes the sequence of customer  $c$ . Formally:

$$s(S, C) = \frac{|c \mid S \prec S_c|}{|C|}. \quad (6.18)$$

Finding these large sequences can be compared to finding large item sets as described in Sect. 6.5. Several large sequences might be found over  $C$ . According to the above problem formulation, we are only interested in the maximum sequences. Whether a sequence is a maximum sequence can be determined based on its length and whether it is contained in another large sequence. The length of sequence  $s$ , i.e.,  $\text{length}(s)$  is defined as the number of items within the sequential pattern [2]. A sequential pattern  $S$  is maximal if  $\nexists S'$  with  $S \prec S'$  and  $\text{length}(S) \leq \text{length}(S')$ .

*Example 6.7 (Sequence Mining for Adverse Drug Events)* Take the customer sequence data as provided in Example 6.5 and assume a minimal support of 0.4. Then, the following large sequences are found:

- $S_1 = \langle \{\text{Aspirin}\} \rangle$  with support 1 and length 1.
- $S_2 = \langle \{\text{Ibu}\} \rangle$  with support 0.5 and length 1.
- $S_3 = \langle \{\text{BetaBlock}\} \rangle$  with support 0.5 and length 1.
- $S_4 = \langle \{\text{AntiBiotics}\} \rangle$  with support 0.5 and length 1.
- $S_5 = \langle \{\text{Aspirin}\}, \{\text{BetaBlock}\} \rangle$  with support 0.5 and length 2.
- $S_6 = \langle \{\text{Aspirin}\}, \{\text{AntiBiotics}\} \rangle$  with support 0.5 and length 2.
- $S_7 = \langle \{\text{Aspirin}, \text{Ibu}\}, \{\text{AntiBiotics}\} \rangle$  with support 0.5 and length 3.

Sequences such as  $\langle \{\text{Aspirin}\} \rangle$  are not included in the result as they are no maximum sequences, but contained in larger sequences.

Let us conclude with some *algorithmic considerations*. Starting from a set of transactions  $T$ , the first step is to determine the necessary data structure, i.e., the data is organized along the customers as in the table given in Sect. 6.6. Then, large sequences are determined based on their support, and the fact that they exceed the minimal support. This phase is crucial due to its potential complexity, and several algorithms exist to implement the efficient finding of large sequences [16]. Within the transformation phase, we determine which of the large sequences are contained in which customer sequences. For this test, the sequences can be mapped onto their corresponding item sets (cf. Sect. 6.5). The transformation phase drops those customer sequences that do not contain any large sequences.

Then the item sets are retransformed into sequences. Finally, the maximum sequences are determined. Starting from the sequence of maximum length  $\max$ , all sequences of length  $k = \max, \dots, 1$  are visited, and all the sequences that are contained in them are dropped.

As mentioned before, the basic principle of sequence mining is the same since 1995. However, several algorithms were presented to make the different phases of sequence mining more efficient. A taxonomy and evaluation of existing algorithms is presented in [16]. Another overview together with implementations of different algorithms may be found on <http://www.philippe-fournier-viger.com/spmf/>.

## 6.7 Episode Mining

Similar to sequential mining, episode mining [17] aims at finding frequent event patterns. There are two basic differences between both techniques, i.e., (1) the input data and (2) the structure of the patterns to be mined.

### Sequence Mining vs. Episode Mining

1. *Input data:* Sequence mining grounds on a transactional set with associated time stamps. Episode mining assumes a continuous *stream* of time-stamped events.
2. *Pattern structure:* Sequence mining aims at finding maximum sequences of item sets. Episode mining aims at episodes where an episode is a “partially ordered collection of events occurring together” [17]. Episodes are mined on the basis of an event sequence but may offer more structure. Basically, we can distinguish between *serial* and *parallel* episodes.

In order to illustrate the above considerations, we alter the adverse drug event example:

*Example 6.8 (Sequence of Drug Prescription Events)* Now the prescription of drugs is stored within a stream of prescription events (A:Aspirin, I:Ibu, B:BetaBlock, M:Marcumar) where each event has a time stamp:

$$s = <(A,2), (M,3), (A,4), (B,5), (A,8), (M,9), (B,10), (I,12), (A,13), (A,15), (M,16), (B,18), (A,19)>.$$

A serial episode would express, for example, that A is always followed by M. An example for a parallel episode would be that A and B frequently occur together but in an arbitrary order. Serial and parallel patterns can also be mixed within an episode, for example, A and B occur together in an arbitrary order, and eventually they are always followed by M.

When compared to sequence mining, the challenge here is to find the “neighborhood” potential patterns that might occur within. In sequence mining, we know which item sets or sequences occur in which transaction and for which customer. For episode mining, an occurrence must be confined to a segment of the event stream. Otherwise, if an event A occurs, for example, in the beginning and at the end, and 10,000 events happen in between, it could also be considered as frequent. Thus, episode mining relies on the definition of *windows* of a certain size that subdivide the event stream. The occurrence of episodes can then be analyzed based on the windows. Fundamental definitions for episode mining are provided in Definition 6.8 based on [17]:

**Definition 6.8 (Notions in the Context of Episode Mining)** According to [17], episode mining necessitates the following notions:

- Let  $E$  be the event set of interest and  $V$  be a set of nodes.
- Event sequence  $s := <(e_1, t_1), \dots, (e_n, t_n)>$  where events  $e_i \in E$  are assigned time stamps  $t_i$  with  $t_i \leq t_{i+1}$ .  $t_1$  denotes the start time of  $s$  and  $t_n$  the end time, respectively.
- Episode  $\epsilon := (v, \leq, g)$  with  $v$  being a set of nodes,  $\leq$  is a partial order, and  $g$  is a mapping function with  $g : V \rightarrow E$ , i.e.,  $g$  maps each of the nodes to an event type.

Episodes can be represented as graphs with a set of nodes  $v$  and edges that represent the orders between the nodes. A serial pattern between the two nodes A and B will be reflected by a directed edge (A,B). For a parallel pattern between A and B, there will be no edge between A and B.

- A window is a segment of the event sequence  $s$ , i.e., window  $w := (s, t_s^w, t_e^w)$  with  $t_s^w < t_n$  and  $t_e^w > t_1$ .
- For window  $w := (s, t_s^w, t_e^w)$ , its width is defined as  $\text{width}(w) := t_e^w - t_s^w$ .
- Set  $\mathcal{W}(s, ws) := \{w \text{ over } s \mid \text{width}(w) = ws\}$ .

We want to find out how often a given episode  $\epsilon$  occurs for windows on sequence  $s$  of size  $ws$ . The frequency for this can be calculated as follows [17]:

$$f(\epsilon, s, ws) := \frac{|\{w \in \mathcal{W}(s, ws) \mid \epsilon \text{ occurs in } w\}|}{|\mathcal{W}(s, ws)|}. \quad (6.19)$$

Then we can define a minimal frequency threshold. All episodes that happen with a frequency  $f$  exceeding the minimal frequency threshold are denoted as frequent.

For illustration, consider the following example:

*Example 6.9 (Episode Mining for Drug Description (ctd))* Consider again event sequence  $s$  provided in Example 6.8. An example window would be  $w = (s, 3, 5)$  containing event occurrences A, M, A. Overall,  $s$  contains 9 windows of size 3. Describing them in terms of their event occurrences, the set of windows of size 3 turns out as

$$\mathcal{W}(s, 3) = \{(A, M, A), (M, A, B), (A, B, A), (B, A, M), (A, M, B), (M, B, I), (B, I, A), (I, A, A), (A, A, M)\}.$$

Assume that we are interested in episode  $\epsilon = (\{v_1, v_2\}, \leq, g)$  with  $g(v_1) = A$  and  $g(v_2) = B$ .  $\epsilon$  is a sequential episode as we are looking for patterns where event  $A$  precedes event  $M$ .

Assume that we want to check whether  $\epsilon$  is frequent in  $s$  with a minimum frequency of 0.4. The frequency of  $\epsilon$  turns out as  $f(\epsilon, s, ws) = \frac{4}{9} \approx 0.44$ . Thus,  $\epsilon$  can be considered as frequent.

Similar to association mining, the challenge is to find frequent episodes, i.e., if you do not have a certain episode in mind. The associated algorithms calculate candidate sets and exploit the fact that if an episode is frequent, all sub-episodes are frequent too [17]. Intuitively, episode  $\epsilon$  is a sub-episode of episode  $\delta$  if the corresponding graph of  $\epsilon$  is a subgraph of the graph corresponding to  $\delta$  (for a formal definition of sub-episode, see [17]).

## 6.8 Conclusion and Lessons Learned

Temporal analysis techniques based on events are useful for many applications. We used the analysis of adverse drug events in this chapter. Other prominent applications are basket analysis for association mining and Web usage for sequence

mining. In basket analysis, we are interested in finding rules on which good are frequently sold together. Based on such rules, customer-tailored offerings become possible.

Applying sequence mining to Web usage enables detection of “user navigational patterns on the world wide Web by extracting knowledge from Web logs” [16]. Again understanding the patterns of users navigating Web pages can help to place advertisements. Episode mining has been applied, for example, in the security domain by finding intrusion attacks [13].

As discussed in [60], sequence and episode mining show some relatedness with process mining as presented in Chap. 7. As we learned in Chap. 3, process mining techniques also work on events, more precisely, on event logs. Imagine now the task to derive an entire process model as the one depicted in Fig. 4.2 and compare this with sequences and episodes as exemplified in Sects. 6.6 and 6.7. At first sight, process models constitute by far more complex structures as sequences or episodes. In particular, sequences and episodes focus on the observable behavior rather than on the non-observable behavior [60]. Moreover, more complex patterns such as loops or alternative branchings that are common for processes cannot be derived and described in terms of sequences and episodes.

## 6.9 Recommended Reading

The following readings provide overviews on temporal data mining (cf. Laxman 2006; Mitsa 2010; Roddick 2002) and time series analysis techniques (cf. Hamilton 1994). A more specific literature on association rule mining, Hipp (2000) can be named as well as Mabroukeh (2010) on sequential pattern mining.

- Hamilton JD (1994) Time series analysis (2), Princeton University Press
- Hipp J, Güntzer U, Nakhaeizadeh G (2000) Algorithms for association rule mining—a general survey and comparison. ACM SIGKDD Explorations Newsletter 2(1):58–64
- Laxman S, Sastry PS (2006) A survey of temporal data mining, Sadhana 31(2):173–198
- Mabroukeh NR, Ezeife CI (2010) A taxonomy of sequential pattern mining algorithms. ACM Computing Surveys 43(1):3
- Mitsa T (2010) Temporal data mining, CRC Press
- Roddick JF, Spiliopoulou M (2002) A survey of temporal knowledge discovery paradigms and methods. IEEE Transactions on Knowledge and Data Engineering 14(4):750–767

## References

1. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Bocca JB, Jarke M, Zaniolo C (eds) VLDB'94: International conference on very large databases. Morgan Kaufmann, San Francisco, pp 487–499
2. Agrawal R, Srikant R (1995) Mining sequential patterns. In: Yu PS, Chen ALP (eds) ICDE'95: International conference on data engineering. IEEE, Los Alamitos, California, Washington, Tokyo, pp 3–14
3. Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. ACM SIGMOD Rec 22(2):207–216
4. Antunes CM, Oliveira AL (2001) Temporal data mining: an overview. In: KDD workshop on temporal data mining, pp 1–13
5. Baldi P, Frasconi P, Smyth P (2003) Modeling the internet and the web: probabilistic methods and algorithms. Wiley, New York
6. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
7. Broström G (2012) Event history analysis with R. CRC, Boca Raton
8. Everitt BS, Hothorn T (2006) A handbook of statistical analysis using R. Chapman & Hall/CRC, New York
9. Ferreira DR, Gillblad D (2009) Discovering process models from unlabelled event logs. In: Dayal U, Eder J, Koehler J, Reijers HA (eds) BPM'09: international conference on business process management. Lecture notes in computer science, vol 5701. Springer, Heidelberg, pp 143–158
10. Giorgino T (2009) Computing and visualizing dynamic time warping alignments in R: the dtw package. J Stat Softw 31(7):1–24
11. Hamilton JD (1994) Time series analysis (2). Princeton University Press, Princeton
12. Hipp J, Güntzer U, Nakhaeizadeh G (2000) Algorithms for association rule mining—a general survey and comparison. ACM SIGKDD Explor Newslett 2(1):58–64
13. Julisch K, Dacier M (2002) Mining intrusion detection alarms for actionable knowledge. In: ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 366–375
14. Killick R, Fearnhead P, Eckley IA (2012) Optimal detection of changepoints with a linear computational cost. JASA 107(500):1590–1598
15. Laxman S, Sastry PS (2006) A survey of temporal data mining. Sadhana 31(2):173–198
16. Mabroukeh NR, Ezeife CI (2010) A taxonomy of sequential pattern mining algorithms. ACM Comput Surv 43(1):3
17. Mannila H, Toivonen H, Verkamo IA (1997) Discovery of frequent episodes in event sequences. Data Min Knowl Discov 1(3):259–289
18. Mitsu T (2010) Temporal data mining. CRC, Boca Raton
19. Müller M (2007) Dynamic time warping. In: Müller M (ed) Information retrieval for music and motion, Chapter 4. Springer, New York, pp 69–84
20. Rebuge A, Ferreira DR (2012) Business process analysis in health care environments: a methodology based on process mining. Inf Syst 37(2):99–116
21. Roddick JF, Spiliopoulou M (2002) A survey of temporal knowledge discovery paradigms and methods. IEEE Trans Knowl Data Eng 14(4):750–767
22. Shmueli G, Patel NR, Bruce PC (2010) Data mining for business intelligence—concepts, techniques, and applications in Microsoft Office Excel with XLMiner. Wiley, New York
23. Silva EG, Teixeira AAC (2008) Surveying structural change: seminal contributions and a bibliometric account. Struct Chang Econ Dyn 19(4):273–300
24. van Dongen S (2000) Graph clustering by flow simulation. Ph.D. thesis, University of Utrecht

# Chapter 7

## Process Analysis

**Abstract** In this chapter, we present analysis techniques for business processes. This includes process analysis and simulation, process performance management, and process mining.

### 7.1 Introduction and Terminology

Process analysis techniques are used to answer process-related questions and to assess key performance indicators such as throughput times for process instances. In particular, process analysis techniques can be used to evaluate *qualitative* as well as *quantitative* analysis questions over process models and process instances.

*Qualitative* questions, in general, address the structure, behavior, and quality of business process models on the one side and qualitative execution aspects of process instances on the other side. According to [6], qualitative questions refer to the identification of unnecessary parts or value-added steps in the process. In addition, qualitative process analysis subsumes process *verification* [31], i.e., formulating and checking the soundness of process models and process instances. Note that the properties that are to be verified partly depend on the business process modeling language (cf. Chap. 2).

Basically, we distinguish between structural, behavioral, and semantic soundness in business processes [17, 38]. Which criteria must be fulfilled in order to guarantee soundness partly depends on the underlying process meta-model [23]. The structural soundness of a Petri net, for example, requires a bipartite structuring of the process graph. Moreover, certain properties of the process graph structure, e.g., connectedness or uniqueness of process start and end nodes, might be demanded.

*Behavioral soundness* implies—among other properties—that no transitions are dead, i.e., they cannot fire anymore, except for the final state where the end state is marked with a token [38].

In this book, we assume process models that are structurally and behaviorally sound. How to check semantic soundness will be discussed in this chapter in the context of business process compliance.

*Semantic soundness* can be approached from two angles: first of all, semantic correctness might imply the question of how well a process model reflects reality. This question can be answered by process *validation*. Validation techniques for

comparing process models and real-world process executions by means of *conformance checking* will be presented in Sect. 7.4.3.

Secondly, semantic soundness includes the verification of semantic properties over business processes that are usually captured by semantic process constraints. It is referred to by the term *business process compliance* [17].

BPM tools often include qualitative process analysis functionalities to check, for example, for business process flaws, such as media breaks<sup>1</sup> [3], or activities without any assigned actor within a process model. Commercial BPM tools, e.g., ARIS Platform,<sup>2</sup> offer various ways of documenting the qualitative properties of process models such as organizational handbooks. One aim of qualitative process analysis is to redesign the process model in a leaner or more efficient way. Redesign measures can include the reordering of process activities (e.g., parallelization) or centralization of resources. A collection of best practices in business process redesign can be found in [19].

Quantitative process analysis is complementary to qualitative process analysis as “results obtained from qualitative analysis are sometimes not detailed enough to provide a solid basis for decision making” [6]. If qualitative analysis yields, for example, that a certain part of the process model probably constitutes a bottleneck in the process execution, this result can be underpinned by quantitatively analyzing the process models by determining process performance measurements.

Figure 7.1 shows the life cycles of business processes (BP) and automated processes (AP), i.e., workflows. A business process design and automation project is often divided into two subprojects, i.e., firstly a project for acquiring, designing, and optimizing the business processes of interest and secondly a process automation project that designs the automated processes, implements and configures them, creates and starts the executed process instances, and diagnoses these running processes.

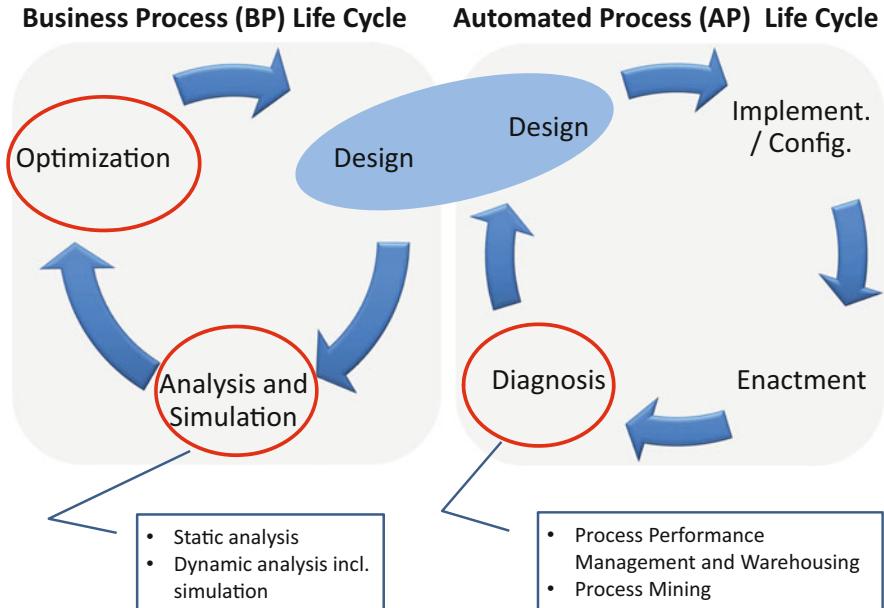
It is important to understand that the analysis techniques for BP operate on BP models, whereas analysis techniques for AP operate on real-world execution data of processes, e.g., on event logs. Note that for all phases of the BP life cycle, quantitative as well as qualitative questions might arise. Examples for a qualitative and quantitative question are summarized in Table 7.1.

The following sections present analysis techniques for the different phases of BP and AP life cycles. Section 7.2 introduces techniques for BP analysis and simulation as well as optimization. Section 7.3 illustrates techniques that are used for measuring the performance of running process instances and for analyzing process execution data within a process warehouse. Section 7.4 introduces process mining as a collection of techniques to analyze processes during runtime (*online*)

---

<sup>1</sup>Media breaks occur, for example, if computerized data is printed, transferred as paper-based form via some process steps, and later inserted into a computerized format again.

<sup>2</sup>[http://www.softwareag.com/corporate/products/arisp\\_platform/default.asp](http://www.softwareag.com/corporate/products/arisp_platform/default.asp).



**Fig. 7.1** Analysis phases within business process (BP) and automated process (AP) life cycle, cf. [9]

**Table 7.1** Examples for qualitative and quantitative questions during different phases of BP and AP life cycle

BP Analysis and Simulation	<p><i>Which activities are not assigned a role? (qualitative)</i></p> <p><i>How long is the average duration of an activity XY based on simulation? (quantitative)</i></p>
AP Diagnosis	<p><i>Given a set of process instance executions, how does the underlying process model look like? (qualitative)?</i></p> <p><i>How many running process instances currently exceed the deadline? (quantitative)?</i></p>

or *ex post (offline)*. While Sects. 7.2–7.4 refer to the analysis of process models and process instances, Sect. 7.5 draws on business process compliance and includes business rules and constraints as additional aspects.

## 7.2 Business Process Analysis and Simulation

Before automating business processes within *process-aware information systems* (PAIS), they are often modeled, analyzed, and optimized at a semantically higher level using specification languages such as BPMN or EPCs (cf. Chap. 2). The typical

goals of business process analysis are the reduction of errors and the optimization of key performance indicators such as process throughput time. Two example questions have been formulated in Table 7.1:

*Which activities are not assigned a role? (qualitative)*

*How long is the average duration of activity XY based on simulation? (quantitative)*

A first distinction for choosing an analysis technique is what input is used for the analysis. We can distinguish between analysis that is based on the business process model only (*static analysis*) and analysis that uses the process model and simulation data (*dynamic analysis*).

### 7.2.1 Static Analysis

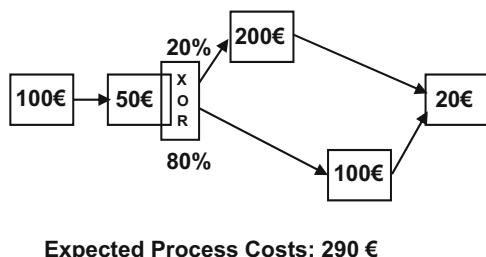
Static analysis refers to all techniques that are applied at process model level, i.e., without taking into consideration information on process executions in terms of simulation data or real process instance execution data. An example is process cost calculation as depicted in Fig. 7.2. The estimated costs for each process activity together with the probability assigned at the outgoing paths of the XOR branching can be aggregated to the expected overall process costs.

Tools such as ARIS Business Process Analysis Platform<sup>3</sup> enable the aggregation of numeric values (mainly time and costs) at process model level.

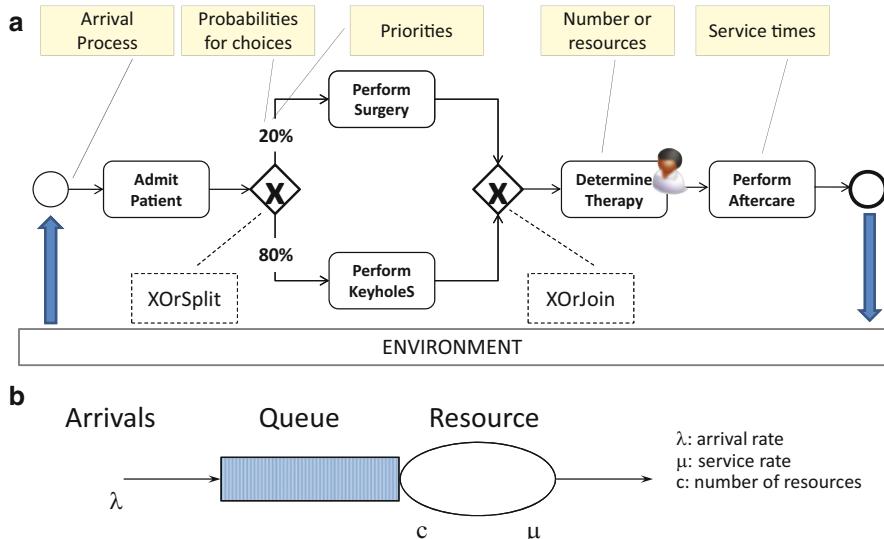
### 7.2.2 Dynamic Analysis and Simulation

Dynamic analysis aims at predicting the behavior of the process during execution and is based on *simulation* data. Simulation refers to the artificial creation and execution of process instances taking into consideration assigned parameters. The simulation of processes does not involve their execution, i.e., no actual data is written or read, no work lists are created, and no activity programs are invoked.

**Fig. 7.2** Process cost calculation (example)



<sup>3</sup><http://www.softwareag.com/corporate/products/aris/bpa/overview/default.asp>.



**Fig. 7.3** (a) Simulation setup and (b) modeling instance arrival based on waiting queues; figure based on [32]

The main goal of a process simulation is the quantitative evaluation of the process model. Typical input parameters for process simulation include:

#### Input Parameters for Business Process Simulation:

- The number of simulated processes
- Probabilities at alternative branchings
- Capacities of users and resources
- Strategies to resolve bottlenecks
- Costs, processing, and transport durations (some tools enable the definition of discrete values but also the definition based on probability distributions)

Figure 7.3 illustrates a typical process simulation setup [32] including the parameters as mentioned above. It is also of interest to consider the arrival process of process executions coming from the environment, produced, for instance, by some other process. In the treatment example, this arrival process could reflect patients being referred from the general practitioner to a specialist. The arrival process of the process instances can be simulated based on *waiting queues* [32], not only at the start point of a process but also at every other process activity [21].

Several BPM tools offer process simulation functionality, for example, Bonapart, ARIS Business Process Analysis Platform, and Signavio.

Typically, the simulation function realized in BPM tools enables the definition of several simulation parameters for process activities. Consider the treatment example displayed in Fig. 7.3, specifically, the process activity determine therapy.

Typically, based on the BPM tool, users would be assigned to this activity at first. For users, it is mostly possible to deposit unavailability times and processing strategies that are taken into consideration during the simulation. Connecting the process perspective with the organizational perspective allows to analyze the resource utilization and therefore might be helpful in detecting bottlenecks. Further on, in most cases, it is possible to define processing times for activities based on probability distributions such as normal distribution with a certain mean and standard deviation. This simulates that for many process activities, realistically, no fixed processing/execution time can be determined, but rather experience data with a deviation. At branching points, probabilities for selecting the different outgoing paths can be determined in a discrete way or based on probability distributions. Furthermore, usually at the starting points of the process, it can be set how many incoming processes will be generated. For example, this may be based on a fixed value within a given time frame or on a probability distribution.

During simulation, processing times are determined on the grounds of this probability distribution. Note that simulation is mostly accompanied by an *animation*, i.e., the visual reflection of the process execution (“blinking process activities”). A process simulation typically yields simulation protocols as well as different aggregations, i.e.:

#### **Expected Output from Business Process Simulation:**

- Process duration under given assumptions (e.g., a defined number of process executions)
- The number of possible process executions in a given time frame
- The workload of resources
- The localization of bottlenecks and flaws within the process model

Some BPM tools offer graphical support in order to visualize the simulation results. However, the interpretation of what the simulation results actually mean, e.g., where will the process bottlenecks occur during runtime, remains still a task for the (human) process analyst.

In addition to in-built simulation functionalities as offered by BPM tools, simulation scenarios can be set up by using process simulation environments such as colored Petri net (CPN) Tools<sup>4</sup> [32]. CPN tools provide a powerful environment for modeling and simulating processes. Due to a certain complexity in specifying the models and parameters, other simulation environments for specific analysis purposes have been developed recently. Examples include SecSy [1] for the simulation of processes with respect to security-relevant aspects or *DPA<sub>Sim</sub><sup>TimeSeries</sup>* [11] for the simulation of times series data in process applications.

---

<sup>4</sup><http://cpn-tools.org/>.

### 7.2.3 Optimization

The previous sections provided an overview on *how* business processes can be analyzed. The *what* has been addressed in an exemplary way. Literature mostly names four dimensions of a business process which can be analyzed and optimized [19, 24]: cost, time, quality, and flexibility.

Optimizing all four dimensions at the same time is hard or even impossible. Reducing, for example, the throughput time of the process execution might result in higher costs or a lower quality of the products [20].

Cost and time are quantitative dimensions. Hence, they can be analyzed using static and dynamic analysis including simulation. It is much harder to analyze quality and flexibility as they are qualitative dimensions. One possibility would be to find appropriate quantifications for quality and flexibility. Appropriate means that the quantification must yield meaningful results when applying static or dynamic analysis.

How can a business process be optimized? In [20], several best practices for business process optimizations are summarized that have been also evaluated with practitioners. Structural optimizations might include eliminating, combining, and reordering activities.

An activity might be eliminated, for example, if it has been executed only for historical reasons, e.g., archiving a certain paper form that is available in an online archive anyway. Combining activities is reasonable if their combined execution can be handled more efficiently than in a separated manner. Reordering activities refers to sequentializing parallel activities or parallelizing sequentially ordered activities. Reordering can also mean to reorder the process.

Moreover, there are optimizations that regard the organizational perspective of a business process such as empowering actors, going from specialists to generalists or vice versa, and the reduction or extension of organizational resources.

*Empowering actors* can mean to integrate them better into decisions. The reduction of organizational resources probably leads to a reduction of costs, their extension to an increase of costs. In turn, the extension of resources might result in decreased throughput times or increased quality and flexibility.

Finally, there are best practices that concern the overall process such as its integration with another process.

Overall, there is no automatism for optimizing business processes, i.e., there are no rules that say *if you detect this kind of flaw in the business process, you apply that kind of optimization*. Mostly, business process optimization or redesign is a “trial-and-error” procedure, i.e., applying a best practice and try to evaluate the success.

For some of the best practices, the success of applying them can be measured more easily than for others. For some of them, we can even analyze their success at business process model level, i.e., before implementing and executing the process in real life. Examples include structural optimizations and the reduction/extension

of organizational resources. For these best practices, static and dynamic analysis including simulation can be used for the original business process (AS-IS model) and the optimized one (TO-BE model). The results can be compared, and the success of the evaluation can be evaluated. Some BPM tools offer support for this variant management.

#### 7.2.4 Summary: Process Analysis and Simulation

In many practical BI applications, process models are developed based on a small set of available data, for example, interviews, workshops, or guidelines. In order to understand how these process models will “behave” during runtime, often, process simulation is used. In this section, we described how to set up simulation models and how to interpret the resulting data. Further, we try to raise awareness that process simulation and process execution are two different things. Simulation creates artificial data, whereas process execution creates real-world data. How to analyze the latter will be discussed in the following sections.

### 7.3 Process Performance Management and Warehousing

Quantitative process analysis techniques such as simulation and queuing are applied during the process design phase (cf. Fig. 7.1), i.e., they include process model information and create artificial process execution data. Hence, as stated in [31], a simulation mimics reality, but not vice versa.

#### 7.3.1 Performance Management

Process performance management provides means to analyze real-world process execution data in a quantitative way. More precisely, typical process performance analysis tasks are process monitoring, monitoring of key performance indicators defined on process executions (e.g., throughput time), and analysis of further aspects such as organizational structures.

A challenge in this context is the provision of process execution data in real time (cf. Chap. 3) in such a way that the above tasks can be executed in a satisfactory way. For process monitoring, most process execution engines (also referred to as workflow engines) provide monitoring components that visualize the current execution states of process instances, e.g., the *process monitor* of the AristaFlow

BPM Suite<sup>5</sup> or the monitoring component of the Cloud Process Execution Engine (CPEE).<sup>6</sup>

Some of the tools extend the monitoring function with the specification and monitoring of key performance indicators (e.g., throughput time) and functionalities. The latter include sending alerts if a certain KPI is exceeded or a defined deadline is going to be violated. This extended process monitoring component is often referred to as process cockpit or process dashboard. Tools for process performance management include ARIS Performance Manager (ARIS PPM) and IBM Business Monitor.

In IBM Business Modeler Advanced Version 7.0, for example, KPIs can be either chosen from a catalog of predefined KPIs or defined individually. For visualizations of KPIs, we refer to Chap. 4. The provided KPI catalog is categorized into different domains such as supply chain planning with associated KPIs. In addition, it is possible to define instance measures that can be monitored through a dashboard as well as by alerts. The latter inform users about certain situations during runtime, e.g., a task duration exceeds a certain threshold. Aggregated measures can be calculated based on instance measures. An example is the aggregated duration for a certain task over all instance executions.

The provision of process execution data in realtime remains a major challenge for the application of these tools, particularly, if the processes span multiple systems and application components. Hence, some tools demand for some kind of closed-world assumption, i.e., that the processes are executed within one environment, preferably the one provided by the tool itself, following a specific modeling language. Alternatively, the user is burdened with the task of linking process information from different sources and hence establishing an integrated process data basis.

### 7.3.2 *Process Warehousing*

As for data warehousing, process warehousing aims at the offline analysis of process execution data based on a table format. For an example, consider the data structure for process warehouse analysis as illustrated by Fig. 3.13 (cf. Sect. 3.4.1).

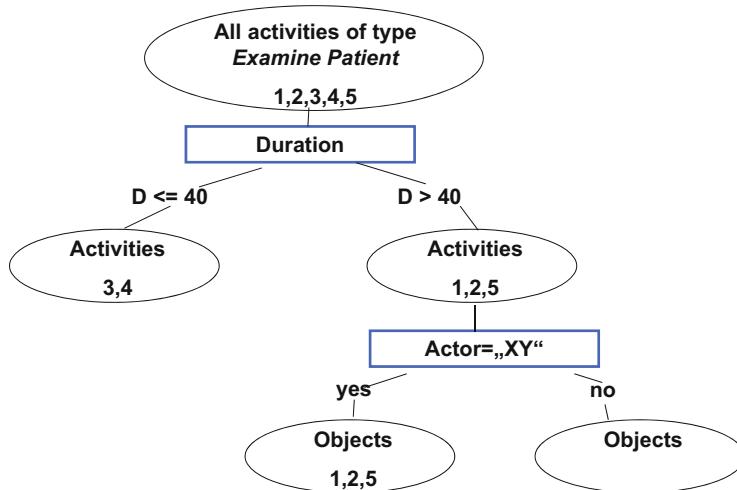
This multidimensional structure can be implemented in different ways by using, for instance, a star or snowflake schema as described in Sect. 3.4.1. Based on multidimensional structures, different analyses such as OLAP and data mining can be carried out:

- *Online analytical processing (OLAP)* techniques are described in detail in Sect. 3.4.1. Conceptually, there is no difference in applying OLAP techniques to data warehouse or process warehouse data.

---

<sup>5</sup>[www.aristaflow.com](http://www.aristaflow.com).

<sup>6</sup>[cpee.org](http://cpee.org).



**Fig. 7.4** Mining process warehouse data: classification by decision trees [11]

- *Data mining* techniques foster the extraction of implicit and not-yet-known hypotheses from data. Several data mining techniques have been explained in Chaps. 5 and 6. In the following, we want to show how these techniques can be specifically applied in the context of multidimensional process data.

Not much investigation exists on how to apply data mining techniques to process warehouse data. One example is the application of classification, more precisely, decision trees, that can be used for the analysis of exceptions in process executions such as overlong duration of activities [11].

#### EBMC<sup>2</sup> Use Case: Process Warehouse Analysis Applying Decision Trees

Consider the process warehouse setting of the EBMC<sup>2</sup> project as depicted in Fig. 3.9. Figure 7.4 shows a (artificial) decision tree that could have been generated based on this data. Analyzing the duration of the task `Examine Patient`, a distinction could be drawn between process activities of duration below and above a value of 40. For the activities of longer duration, i.e., activities 1, 2, and 5, it could be concluded that all of these activities have been executed by actor XY. Note that such results can be critical if the performance of certain persons is evaluated. Hence, they may even be contradictory with the works council. In this case, the analysis can rather be conducted on groups of person or with anonymized data. Details on the EBMC<sup>2</sup> project can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

Decision trees constitute one of the few techniques that have been proposed in connection with process warehouse data. Some approaches suggest the combined application of process mining and data mining [11, 31]. As an example, we will

explain the application of decision trees in connection with process mining analysis in Chap. 8.

### 7.3.3 Summary: Process Performance Management and Warehousing

Overall, the real-time monitoring and analysis of process execution data is of great importance in order to control running process instances. The main focus has been put on the specification and supervision of KPIs with associated warnings and alerts. Less attention has been turned to the analysis of real-time data. Apart from the definition of KPIs and specific techniques to analyze real-time data, the visualization of the running process instances and the KPIs plays an important role. Usually, the user monitors running process instances via a so-called process cockpit or dashboard where, for example, traffic light or steering wheel metaphors are used to convey the information to the user. As visualizations might reveal certain disadvantages for monitoring purposes, such as screen size or the complexity of the data [14], a combination with additional methods, e.g., sonifications, is currently investigated [13]. This might become particularly interesting in environments with huge process execution data (often stored and processed based on events). One example for such an environment is the manufacturing domain which has been investigated within the EU FP7 project ADVENTURE.<sup>7</sup>

Creating and maintaining process warehouse data can be an interesting approach for analyzing process data in an offline fashion. It can be of particular interest to integrate the process warehouse data with other warehouse data in the business context as proposed in [5]. A possible drawback of the process warehouse approach is the possibly “unnatural” multidimensional structuring of process data. Processes are living structures that are executed. Hence, a more execution-oriented modeling and analysis of the data in the form of logs is probably more helpful.

In summary, we can distinguish between the analysis of process execution data in an online or offline mode [33]. Further, we can distinguish between analyzing process data in table or log format.

## 7.4 Process Mining

In essence, process mining basically addresses two questions: *process discovery* and *conformance checking* [60]. Note that log repair is mentioned as the third question in this reference. However, we will abstain from this aspect in this book.

---

<sup>7</sup><http://www.fp7-adventure.eu/>.

Given the execution data of a set of process instances, the goal of process discovery is to derive the underlying process model, i.e., the process model that was used to create, initiate, and execute those instances reflected by the log. Doing so can be of high interest for many practical applications where processes are executed. This does not always happen explicitly through a process management or workflow system, but across multiple information systems such as database management systems, document management systems, or ERP systems. From an algorithmic perspective, process discovery techniques take process execution data as input and produce a process model as output. The process model can be expressed by any kind of process description language such as Petri nets (cf. Chap. 2). Process discovery can be estimated as unsupervised learning in case there is no reference process model to compare the results of the discovery.

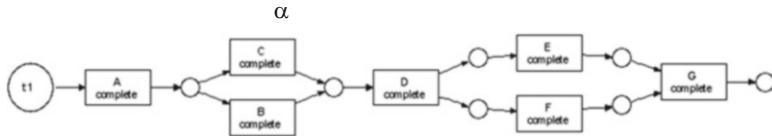
Conformance checking assumes the existence of some process model and checks whether associated log data follows or deviates from this process model. Hence, conformance checking can be subsumed as supervised learning. Typical analysis questions for conformance checking are “where do actors deviate from the process model” or “to what degree does reality reflect the prescribed process structure?”

### 7.4.1 Process Discovery

Process discovery is an analysis technique based on process execution logs, i.e., it analyzes process instances that have already been executed. Let us start with a short recap from Sect. 3.4.2: Process execution logs store information related to process execution, mostly in terms of events. As a minimum requirement to conduct process discovery techniques, events on process task execution (e.g., start or end events) must be contained within the log together with ordering information which can be either provided by time stamps or by the actual ordering in the log. In order to distinguish events of different process runs or instances, either one log does only contain information of one instance execution or the log entries (events) must be additionally equipped with (unique) identifiers for each instance. Events may be further augmented with information on, for example, the actors who performed a certain task. How this additional information can be exploited and analyzed is discussed in Sect. 8.2.

As process mining has significantly matured during the last years, a variety of process discovery techniques exists nowadays. In this section, we start with explaining the principles of the  $\alpha$ -algorithm [4] as it builds the basis for understanding discovery techniques that ground on frequency counting over execution logs. The result of applying the  $\alpha$ -algorithm to an event log is a Petri net that represents the underlying process model.

The *heuristic miner* [36] will be sketched as an advancement of the  $\alpha$ -algorithm. Finally, we discuss the basic idea of the *genetic miner* [5] that constitutes another

**a**Case1:  $\langle A, C, D, E, F, G \rangle$ Case4:  $\langle A, B, D, F, E, G \rangle$ Case2:  $\langle A, B, D, E, F, G \rangle$ Case5:  $\langle A, B, D, E, F, G \rangle$ Case3:  $\langle A, C, D, F, E, G \rangle$ **b**- Sequence:  $a \rightarrow_w b \Leftrightarrow a >_w b \wedge \neg(b >_w a)$ - Exclusive:  $a \#_w b \Leftrightarrow \neg(a >_w b \vee b >_w a)$ - Parallel:  $a \parallel_w b \Leftrightarrow a >_w b \wedge b >_w a$ **c**Sequential:  $A \rightarrow_w B, A \rightarrow_w C, B \rightarrow_w D, C \rightarrow_w D, D \rightarrow_w E, D \rightarrow_w F, E \rightarrow_w G, F \rightarrow_w G$ Exclusive:  $B \#_w C$ Parallel:  $E \parallel_w F$ **d**

**Fig. 7.5** Principles of the  $\alpha$ -algorithm and example (using ProM 5.2). (a) Process execution log, (b) analysis of order relations over process log, (c) result of analysis step, and (d) resulting Petri net (using  $\alpha$ -algorithm in ProM 5.2)

kind of technique, i.e., not based on frequency counting from the log, but based on evolutionary algorithms as optimization methods. For a detailed overview and discussion on existing process discovery techniques, we refer to [60].

Consider the log data provided in Fig. 7.5a, consisting of five cases (i.e., process instance executions), assuming that the occurrences of events connected to task executions in the log reflect their execution order. Note that for applying tools such as ProM,<sup>8</sup> this log has to be provided or transformed into a log format, e.g., MXML or XES (cf. Sect. 3.4.2).

The  $\alpha$ -algorithm starts with analyzing the order between activity entries within each single log. For Case1, for example, activity entry A has occurred before execution of B, and hence, A is considered a predecessor of B. This relation between two activities is denoted by  $A \rightarrow_w B$ .

<sup>8</sup><http://www.processmining.org/prom/start>.

After analyzing each log separately, the relations derived from each log are aggregated across all available logs. For each pair of activities, this aggregation results in one of the three basic relations depicted in Fig. 7.5b, i.e., sequential order, parallel order, or exclusive order (e.g., for each instance, either B or C, but never together). The results are summarized in Fig. 7.5c:

- An example for a sequential order is  $A \rightarrow_W B$  as A was observed before B for all logs.
- An example for activities ordered in parallel is  $E \parallel_W F$  because E happened before F and F happened before E for some logs.
- An example for activities ordered exclusively is  $B \#_W C$  as there is no log containing B and C.

The aggregated relations displayed in Fig. 7.5c are used to construct the resulting process model. In our example, the Petri net as depicted in Fig. 7.5d will result from the analysis (constructed using ProM 5.2<sup>9</sup>).

In summary, the  $\alpha$ -algorithm finds basic relations among activities and constructs the resulting process model.

#### HEP Use Case: Application of $\alpha$ -algorithm

Figure 7.6 illustrates a more complex log example from the HEP case study. 74 process instances (cases) included in the log result in 4,018 events. The mined Petri net model looks quite complex. Such complex or unstructured models are referred to as “spaghetti models,” whereas structured process models resulting from process discovery are denoted as “lasagna models” [60]. Details on the HEP project can be found on the homepage of the book:

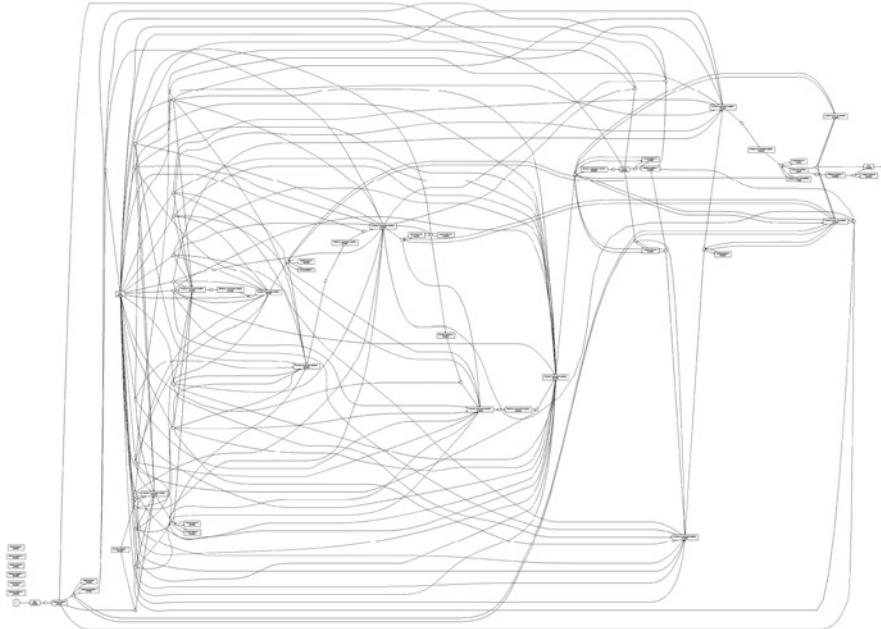
[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

Intuitively, understanding and working with spaghetti models can be cumbersome and error-prone. The reasons for the occurrence of spaghetti models are manifold. For example, such models might result from the characteristics of the underlying process structures and the possible behavior of process actors during process execution (*process-related problems*). According to [17], factors of influence for spaghetti models can be as follows: log data contributed by parallel branches, infrequent traces, and log data contributed by individually modified instances.

The heuristic miner introduced in the sequel is able to deal with infrequent behavior and hence—at least to some extent—with modified instances. Note that it cannot distinguish between behaviors that are part of the process model, but only rarely occurring behavior, and behavior that has been added by an instance change. The phenomena deviating from the prescribed process model during the real-life execution is referred to as *concept drift* [15]. How to exploit information on process deviations and changes is discussed in Sect. 7.4.2.

---

<sup>9</sup>We use ProM 5.2 to present the principles of the  $\alpha$ -algorithm, the heuristic miner, and the genetic miner. For later examples, we will use the more current version ProM 6.3.



**Fig. 7.6**  $\alpha$ -Algorithm applied to HEP log example (using ProM 5.2)

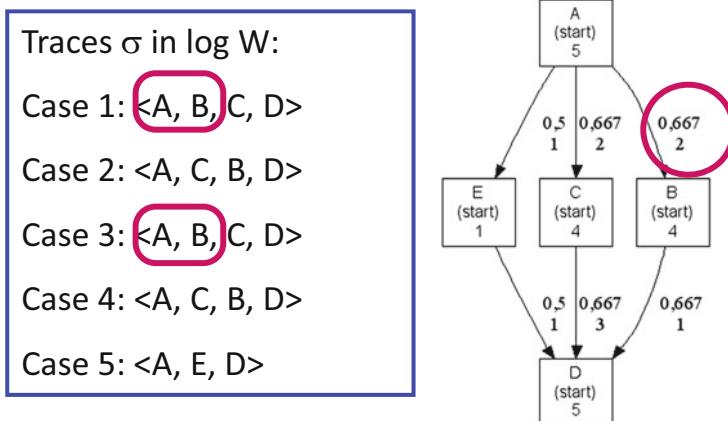
Spaghetti models might also occur due to low log data quality and noise such as incorrect or missing data [16] (cf. Sect. 3.5.2).

Process-related and data-related problems can be tackled from either the algorithmic side or by providing *preprocessing* [8] and *post-processing* [15] strategies for the data or resulting process models, respectively.

In order to be more robust against noise, several process mining algorithms have been developed. One example mentioned before is the heuristic miner that infers the ordering relations between tasks based on their frequencies within the log. At first, it is counted how often activity  $a$  directly follows activity  $b$  in event log  $W$  (denoted by  $a >_W b$ ). Then it is counted how often  $b$  is directly following  $a$  in event log  $W$ , i.e.,  $b >_W a$ . If the number  $|a >_W b|$  is much higher than the number  $|b >_W a|$ , it can be concluded that  $a$  also causally follows  $b$  [37]. The following formula taken from [37] reflects the above description:

$$a \Rightarrow_W b = \frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1}. \quad (7.1)$$

*Example 7.1 (Application of the Heuristic Miner)* Consider the log given in Fig. 7.7. The result of applying the heuristic miner shows that for activities A and B, the relation  $A >_W B$  can be found twice in the log. The opposite relation  $B >_W C$  is not present, resulting in a frequency of  $\frac{2}{2+1} = 0.67$ . Relation



**Fig. 7.7** Example log and result of applying heuristic miner (using ProM 5.2)

$B >_W C$ , for example, is observed twice within the logs as well as the opposite relation  $C >_W B$ . Hence, the probability of a causal relation of  $B >_W C$  turns out as  $\frac{0}{1}$ . Hence, B and C are evaluated as being parallel.

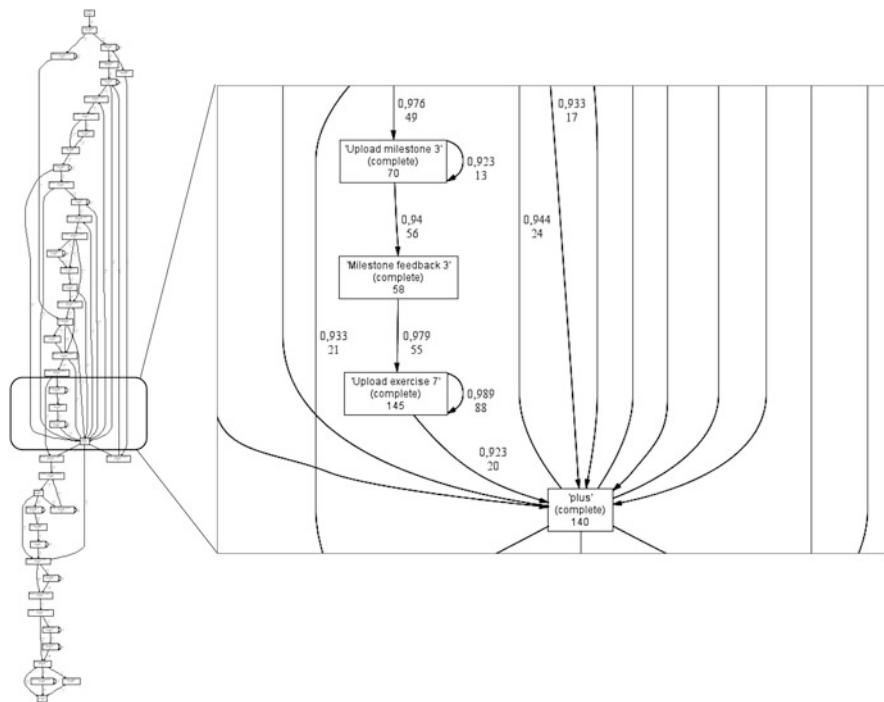
#### HEP Use Case: Applying Heuristic Miner

Recall the result of applying the  $\alpha$ -algorithm to the HEP event log data as depicted in Fig. 7.6. This time, we discover the process model based on the heuristic miner (cf. Fig. 7.8). It can be seen that the model is less complex, i.e., it contains less paths and the flow direction is visible. Roughly, we could call the model in Fig. 7.6 a spaghetti model and the model in Fig. 7.8 a lasagna model. The reduction of the number of paths is caused by cutting infrequently occurring paths. Overall, we can state that in this case, application of the heuristics miner yields a more compact and understandable process model than the application of the  $\alpha$  algorithm. Details on the HEP project can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

Another process mining technique that is resilient toward noise is the *genetic miner* [5] which follows the basic principle of evolutionary algorithms [10]. Such algorithms serve as heuristic optimization techniques that are inspired by natural evolution processes. The basic problem setting is as follows: given an initial solution and an optimization goal (e.g., minimize or maximize a certain value), how can the initial solution evolve towards fulfilling the optimization goal? Translated into a process discovery problem, starting from a set of individuals, i.e., process models, we try to optimize them towards the optimal model, i.e., that model that best reflects the given set of event logs.

Doing so, different challenges have to be addressed. We will explain them in the following and then illustrate the different steps.



**Fig. 7.8** Heuristic miner applied to HEP log example (using ProM 5.2)

### Steps of the Genetic Miner

1. *Encoding individuals:* A first important consideration is how to encode these individuals for optimization, i.e., finding an adequate encoding as genotype for the phenotype of the individuals. The genetic miner encodes the individuals, i.e., process models, as *causal matrices*. A causal matrix for a process model contains the input and output functions, i.e., the sets of activity set that enable or are being enabled by the execution of the activity, respectively.
2. *Creating initial solution:* If the encoding is chosen, an initial population has to be generated, i.e., individuals have to be selected. In general, either an initial population is at hand or created. For the genetic miner, the initial population is generated by taking into consideration all activities present in the logs and imposing random causal relationships on them (e.g., 50 % chance that a causal relation is established between two activities). Heuristic extensions exist that take into consideration frequencies of paths contained within the logs.
3. *Fitness and selection of individuals:* An initial population at hand, it is now important to have a means to measure the quality of the individuals. For this, in general, a *fitness function* is defined that calculates the quality of an individual with respect to the optimization goal and based on the chosen encoding.

The genetic miner employs a fitness function that punishes for imprecise and incomplete process models.

The terms imprecise and incomplete are tightly connected with evaluating the results of process discovery in general as well as with the question of conformance checking. A model is incomplete if not all traces of the log can be replayed on this model. It is imprecise if it allows for producing additional traces that are not present in the log. We will explain the associated terms and techniques in detail in Sects. 7.4.3 and 7.6.

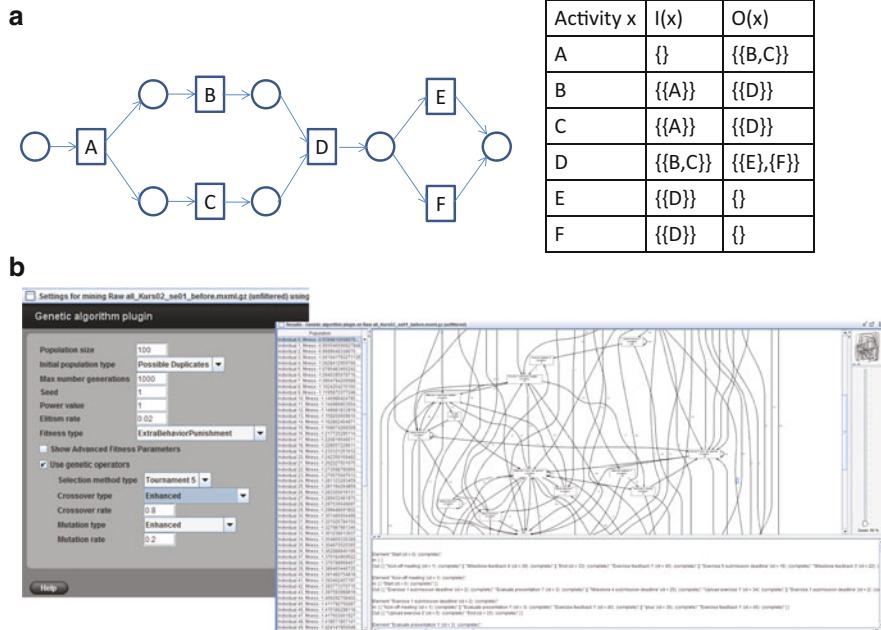
4. *Creating new offspring:* Based on a certain threshold and respecting the diversity of the solution (i.e., not always all the fittest individuals are selected in order to avoid local optima), those individuals are selected that create the next generation (offspring). In order to create offsprings, the two operations *mutation* and *crossover* can be used. Mutation works on one individual, whereas crossover combines the information of two individuals. For the genetic miner, mutation takes the causal matrix of an individual and adds new information to the causal matrix. This can be achieved by, for example, adding an activity from the underlying activity set to the input/output activity sets or by deleting an activity, respectively.
5. *Evaluating offspring:* Based on evaluating the offspring by using the fitness function, new parents for the next iteration are chosen, and so on, until a stop criterion is reached (e.g., the optimum—if known—is reached or a given threshold for the number of iterations is exceeded).

#### HEP Use Case: Application of Genetic Miner

In Fig. 7.9a, we can see a process model represented as a Petri net on the left side and its encoding as causal matrix on the right side. For activity A, for example, the set of activities that have been causally preceding A is empty. This means that A can start immediately. After completing A, two activities B and C follow causally. Hence, these two activities form the output set of A.

Figure 7.9b, c show the different settings for the genetic miner as discussed above as well as the result for mining the HEP log as already presented for the  $\alpha$ -algorithm (cf. Fig. 7.6) and the heuristic miner (cf. Fig. 7.8). It can be seen that after 1,000 iterations, the individuals expose a very low fitness. This is also reflected by the resulting spaghetti model. Hence, comparing the results of alpha-algorithm, heuristic miner, and genetic miner shows that the heuristic miner yields the most compact result. This can be explained by the fact that the heuristic miner removes infrequent traces. Thus, we can conclude that the underlying process shows several variations or even exceptional situations. Details on the HEP project can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)



**Fig. 7.9** Genetic miner basics and HEP use case (Net in c) (using ProM 5.2). **(a)** Process Model (Petri Net) and corresponding Causal Matrix, **(b)** settings for Genetic Miner (ProM 5.2), and **(c)** result for HEP example

As the genetic miner is capable of processing suboptimal solutions, i.e., process models, it is also capable to process noisy event logs.

#### 7.4.2 *Change Mining*

In addition to discovering process models from event data, it can also be of high interest for users to learn about previously applied process changes, particularly when they are about to apply a process change themselves [26, 35]. In the medical domain, for example, it could be helpful to review what other specialists have done in a similar situation. Hence, *change mining* [12] has been developed as means to mine and learn about process changes that have been applied to process instances previously.

Before describing change mining in more detail, the topic of process change will be introduced in a nutshell. Process change is a well-established research topic and relevant for almost any application domain. Basically, one can distinguish between design time and runtime process flexibility. Design time flexibility is referred to

by, for example, the underspecification of parts of the process models and late modeling/binding techniques. Runtime flexibility refers to the ability of adapting running process instances during their execution. For a discussion on process change, see [22, 25].

Now we come back to change mining: change mining is not based on execution logs, but on change logs [28]. Change logs store information about the kind of change operation that has been applied. They use parameters such as time or change originator. Examples for change operations are `INSERT(S, X, A, B)` and `DELETE(S, Y)`. In the first case, an activity X is inserted into schema S between activities A and B. In the second case, activity Y is deleted from schema S. A survey on process change operations and patterns may be found in [34].

Consider the change logs  $\tau_1$  and  $\tau_2$  (cf. execution logs in Sect. 3.4.2):

$$\begin{aligned}\tau_1 &= \langle \text{INSERT}(S, X, A, B), \text{DELETE}(S, Y), \text{INSERT}(Z, S, D, E) \rangle \\ \tau_2 &= \langle \text{DELETE}(S, Y), \text{INSERT}(S, X, A, B) \rangle\end{aligned}$$

Change mining basically applies the heuristic miner to change logs, resulting in *process change graphs*. In other words, instead of constructing the process execution graph, i.e., the process model, the change miner derives the process change graph, i.e., the model that includes all changes that have been applied at instance level in an aggregated manner.

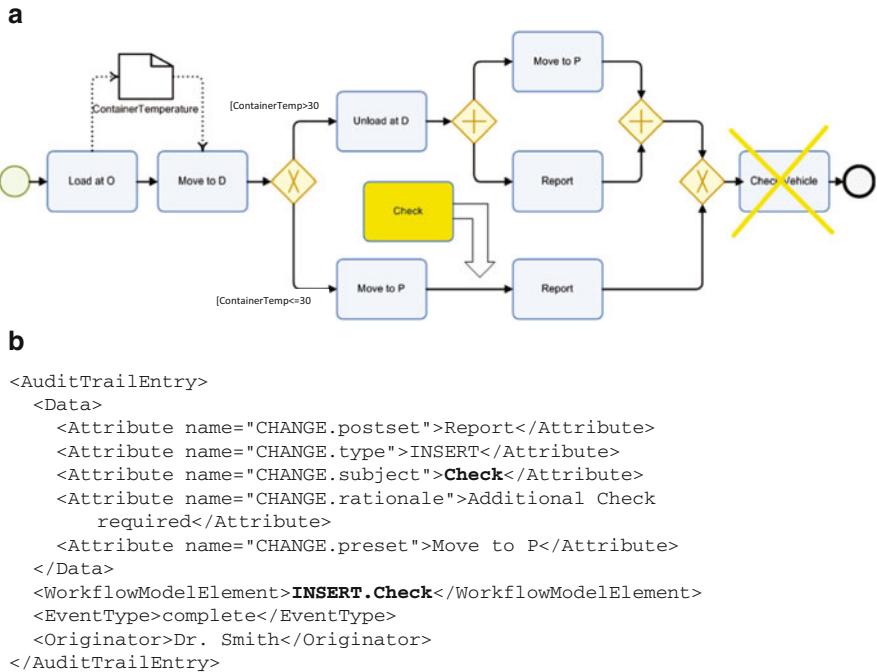
#### Logistics Use Case: Application of Change Mining

Figure 7.10a shows our example from the logistics domain on container transportation. Assume that several process instances have been executed based on this schema. Assume further that for several of these instances change operations have been applied, i.e., inserting the new activity `Check` between the activities `Move to P` and `Report` (`INSERT(S, Check, Move to P, Report)`) and deleting activity `Check Vehicle` (`DELETE(S, Check Vehicle)`). The information on the applied changes is collected for each instance and stored within associated change logs. Figure 7.10b depicts a change log entry for inserting activity `Check` for one of the instances.

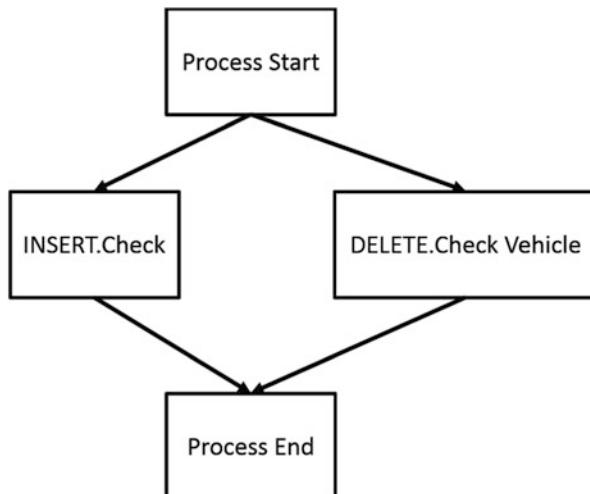
Figure 7.11 illustrates the result of applying the change miner algorithm to the change logs resulting in an event-driven process chain. Within this description language, it can be easily seen that both changes were applied separately but also together. An implementation of change mining is available in ProM 5.2. Details on the container transportation use case can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

In this simple example, the overview on applied changes and their relations can easily be kept. However, for more complex change settings as described in [12], the change mining results can provide a useful tool for users in order to analyze previously applied changes. This information can be taken as input for future change decisions.



**Fig. 7.10** Container transportation process with changes and corresponding change log. (a) Container process with change operations and (b) log data entry



**Fig. 7.11** Container transportation process: result of applying change mining

### 7.4.3 Conformance Checking

Process discovery works on a set of execution logs. Conformance checking requires both as input execution logs and a process model. As stated in [29, 60], the general goal of conformance checking is to identify commonalities and deviations between the real-world process (reflected by the logs) and the modeled process. Hence, conformance checking can be used for compliance checking and auditing (cf. Sect. 7.5). We can estimate conformance checking as supervised learning technique.

Assume the existence of some process model  $P$  represented by a Petri net and a set of event logs  $E$ . The degree to which  $P$  reflects  $E$  can be measured based on a conformance metrics that counts the number of tokens that remain in the process model (*remaining tokens*) and tokens that are missing for process execution along the model after having replayed the entire log data (*missing tokens*).<sup>10</sup>

Remaining tokens occur if the traces have been completely replayed and—except for the final state—at least one state is marked by a token. Missing tokens reflect how many tokens are missing for activating process activities that are in the log.

The formula that puts remaining and missing tokens into one number is as follows [29]:

$$f = 0.5 * \left( 1 - \frac{\sum_i n_i * m_i}{\sum_i n_i * c_i} \right) + 0.5 * \left( 1 - \frac{\sum_i n_i * r_i}{\sum_i n_i * p_i} \right) \quad (7.2)$$

where

- $n_i$  is the number of activities in the process model
- $m_i$  is the number of missing tokens
- $c_i$  is the number of consumed tokens
- $r_i$  is the number of remaining tokens
- $p_i$  is the number of produced tokens

The following abstract example illustrates how the conformance metrics works:

*Example 7.2 (Conformance Metrics Based on Abstract Petri Net Example)* Consider the example depicted in Fig. 7.12. Given a log of 8 traces and the depicted Petri net model  $P$ , conformance metric  $f$  [29] determines a weighted measure between missing and remaining tokens for all logs weighted by the number of logs, and the number of tokens that have been consumed and produced according to the logs on the net. These consumed and produced tokens reflect the conformance of the model to the logs, whereas remaining and missing tokens reflect deviations which are “punished” within the conformance metrics  $f$ .

Intuitively,  $P$  conforms with 4 of the logs, i.e.,  $\langle A, B, D \rangle$ , whereas the other 4 logs, i.e.,  $\langle A, C, D \rangle$ , cannot be completely replayed on  $P$ . The corresponding

---

<sup>10</sup>Note the similarity between the conformance metrics and the fitness function for genetic mining as presented in Sect. 7.4.1.

Petri Net model P:

$\langle A, B, D \rangle$	$\langle A, B, D \rangle: i = 1$	$\langle A, C, D \rangle: i = 2$
$\langle A, B, D \rangle$	Remaining tokens $r_i$	0
$\langle A, B, D \rangle$	Missing tokens $m_i$	1
$\langle A, B, D \rangle$	Produced tokens $p_i$	0
$\langle A, C, D \rangle$	Consumed tokens $c_i$	2
$\langle A, C, D \rangle$	Conformance:	$f = 0.5 * (1 - \frac{3*0+3*1}{3*3+3*2}) + 0.5 * (1 - \frac{3*0+3*1}{3*3+3*2}) = 0.8$

Fig. 7.12 Measuring conformance based on an artificial example

table shows the different numbers for the conformance analysis: for logs  $\langle A, B, D \rangle$ , one token will remain on P, i.e., on place  $p_2$ . The reason is that since  $B$  is not present in the log, it is not executed. Hence, it does not consume the token on  $p_2$ . In turn, one token will be missing, i.e., the one which will trigger  $D$ . For logs  $\langle A, C, D \rangle$ , two tokens are produced and consumed, respectively, by executing  $A$  and  $D$ . For the other 4 logs  $\langle A, B, D \rangle$ , there are neither remaining nor missing tokens and 3 tokens are consumed and produced. By putting all the numbers into the formula,  $f$  turns out to be 80 %.

Petri net model P (cf. Fig. 7.12) represents only a part of the real-world behavior reflected in the logs. Hence, its fitness, i.e., its ability to reflect the log, turns out to be 80 %.

#### 7.4.4 Summary: Process Mining

Process mining basically tackles two analysis questions: (a) based on a set of process execution logs, how does the underlying process model look like (process discovery) and (b) based on a set of process execution logs and a process model, how well does the model reflect the real-world behavior as reflected by the logs (conformance checking)? Process mining techniques have been proven to be relevant in many application domains. In [60], case studies for municipalities and the manufacturing domain are provided. In our EBMC<sup>2</sup> project, for example, the heuristic miner was applied to an initial set of the patient treatment processes. When discussing the results with domain experts, they observed several activities and process executions that aligned with the guideline. Though ten cases are too few to be representative, the experts stated that process discovery is useful as a screening tool [7].

Both algorithms and tool support for process mining have made significant progress during the last years. One remaining challenge is that data is often not available in a process-oriented structure and format. This demands for techniques

that enable the extraction, integration, and transformation of existing data into process logs. A relevant question for future research is how to integrate process mining techniques with further analysis techniques, e.g., data mining. This seems to be promising with respect to gaining more insights into the data and to be able to address more analysis questions. In Sect. 8, we will present existing combined techniques, for example, for mining organizational structures and discuss current limitations and open questions.

## 7.5 Business Process Compliance

Business process compliance has developed as one of the key concerns for process-oriented applications nowadays through many application domains such as finance, security and privacy, health care, service flows, and internal controls. The general question of business process compliance is to check and ensure that business processes and workflows obey to the relevant constraints, rules, guidelines, and controls imposed on the business processes. For simplicity reasons, we will refer to constraints in the following.

### 7.5.1 *Compliance Along the Process Life Cycle*

Business compliance constitutes a challenge throughout the entire business process life cycle [17] ranging from compliant-by-design business processes and design time compliance checks to runtime monitoring approaches.

At design time, different analysis questions arise (cf. Fig. 7.13). One question is whether relevant constraints are entirely considered within the process model or whether there is a co-existence between process models and constraints. In the first case, one can distinguish between imperative and declarative process models. For imperative process models, constraints can be either directly captured within the model design or specified by annotations.

Considering constraints within the models is particularly supported by declarative process modeling notations. Compared to the imperative notions used throughout this book, declarative process models consist of constraints which express themselves what can be done and what cannot be done instead of imposing a strict process execution. If process models are specified in a declarative manner, compliance constraints can be added. Then compliance checking means to identify consistency between the constraints.

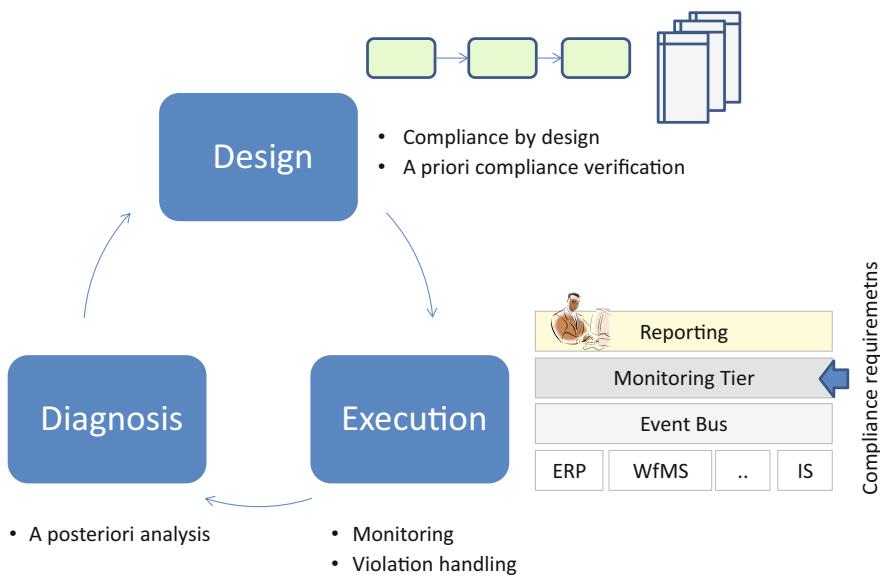


Fig. 7.13 Overview on compliance approaches along process life cycle (following [18])

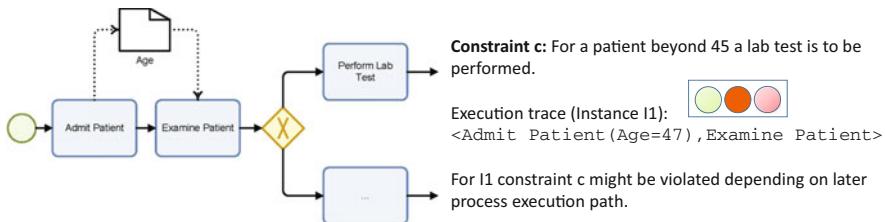


Fig. 7.14 Medical example compliance constraint

At process execution time, it often becomes necessary to monitor the adherence of running processes with imposed compliance constraints due to two aspects: (a) certain aspects of constraints that cannot be checked during design time, e.g., data or time, and (b) sometimes process models are not available. As indicated in Fig. 7.13, process execution information might be available as events associated with process activity executions and stemming from different underlying information systems. In this case, compliance constraints have to be verified on the fly. For an overview and comparison of existing approaches for compliance monitoring, see [18].

Special challenges arise not only in verifying compliance constraints but also in reporting back on violations. In particular, a true/false answer might not be sufficient in all cases, but reasons for compliance violations or even feedback for healing violations are required. The example in Fig. 7.14 shows a fragment of a medical treatment process and an associated constraint  $c$ . Based on the execution trace for a certain instance  $I1$ , we can conclude that  $c$  might be possibly violated in

case the “wrong” path of the XOR branching is chosen. Hence, it could be favorable to suggest the “right” path to the user as proactive handling of a possible violation.

A diagnosis can also comprise compliance-related analysis. In the security domain, for example, so-called a posteriori analysis techniques can be applied to detect security violations during process executions [2]. One set of techniques applied at a posteriori time are process mining approaches such as process discovery, process conformance, or LTL checking.

### 7.5.2 Summary: Compliance Checking

Business process compliance constitutes one of the most challenging questions for enterprises nowadays, and tremendous effort has been spent on providing system-based support for compliance integration and checking in process-aware information systems. The techniques have well matured. However, the challenge is to acquire, model, and formalize relevant compliance constraints over business process for later analysis. Another challenge is the interaction with users in case of compliance violations. In most cases, the pure indication of such a violation is not enough to enable the user to deal with the violation. Hence, advanced strategies for handling compliance violations in a user-friendly way are required.

## 7.6 Evaluation and Assessment

How can we evaluate and assess the results of process analysis?

### 7.6.1 Process Mining

The quality of a discovered process model after mining it from an event log depends on different dimensions. According to [30], these dimensions are *fitness*, *precision*, *generalization*, and *structure*. Fitness and precision have been already discussed along conformance checking (cf. Sect. 7.4.3): fitness refers to how much of the behavior described in the logs is reflected by the discovered model. The concern of precision is that the behavior as described by the logs should be reflected as exactly as possible within the model, i.e., no additional behavior should be producible. Both dimensions, fitness and precision, are captured within the conformance metrics by punishing for behavior that is not reflected and behavior that is additionally reflected.

The evaluation framework in [30] additionally introduces generalization as dimension. Generalization constitutes some kind of counterpart to precision. It aims at avoiding models that can only reflect the behavior of the logs, but no more behavior. Of course, generalization and precision are to be balanced, i.e., the right level of generalization must be determined.

Finally, structure evaluates the process model in terms of complexity. It is possible, for example, to have two process models that are execution equivalent, i.e., bear the same behavior, but are of different structure. One model might contain, for example, duplicate or silent activities<sup>11</sup> for structuring reasons. Then, a metric evaluating the structure would rate the quality of the more compact model higher than the one of more complex structure.

### 7.6.2 *Compliance Checking*

Evaluation in the context of process compliance checking might refer to two aspects, i.e., efficiency and quality of the feedback. Here, efficiency means how long compliance checks take when facing complex process models and a multitude of compliance constraints. Efficiency in the sense as discussed here is mostly relevant for automated compliance checks by a tool or system. Nowadays, compliance is often checked manually by, e.g., auditors. In such settings, it is more difficult to evaluate efficiency.

The quality of feedback is a crucial aspect as the strategy of how to deal with compliance violation depends on the expressiveness of this feedback. At least, compliance checking solutions should precisely pinpoint the source for the violation. Even better, they can offer possible solution strategies when the violation has already happened or proactively warn users in case of potential violations.

Overall, process analysis techniques are of high interest to companies as they provide means to optimize business processes, discover process models, compare different process models, and evaluate the compliance of business processes with a set of relevant constraints.

## 7.7 Conclusion and Lessons Learned

According to our experience and as it is shown by the case studies provided in this book, the interest in process analysis techniques spreads over all kinds of application domains such as health care, manufacturing, or logistics. The adoption of process analysis techniques would be fostered by providing systematic guidelines or methods of how to apply the different techniques. In addition, the interpretation

---

<sup>11</sup>Silent activities are not connected with any action and hence do not produce any log entry.

of the results often requires an advanced understanding of the techniques, but also the domain of interest. In times of big data being a hype, the trend of big process data will raise additional tasks such as the proper visualization of analysis results.

## 7.8 Recommended Reading

Introductions to the areas of business process management and process-aware information systems are provided in Weske (2007) and Dumas (2013). Dumas (2013), moreover, discusses process analysis techniques. van der Aalst (2011) provides introductory material as well as details on process mining and analysis techniques, applications, and related areas.

- Weske, M (2012) Business process management—concepts, languages, architectures. Springer, Heidelberg, 2nd edition
- Dumas M, La Rosa M, Mendling J, Reijers HA (2013) Fundamentals of business process management. Springer, Heidelberg
- van der Aalst WMP (2011) Process mining—discovery, conformance and enhancement of business processes. Springer, Heidelberg

## References

1. Accorsi R, Stocker T (2013) SecSy: synthesizing smart process event logs. In: Jung R, Reichert M (eds) EMISA'13: enterprise modelling and information systems architectures: proceedings of the 5th international workshop on enterprise modelling and information systems architectures workshop. Lecture notes in informatics, vol 222, pp 71–84
2. Accorsi R, Stocker T, Müller G (2013) On the exploitation of process mining for security audits: the process discovery case. In: Shin SY, Maldonado JC (eds) SAC'13: annual ACM symposium on applied computing. ACM, New York, pp 1462–1468
3. Becker J, Bergener P, Breuker D, Räckers M (2012) An empirical assessment of the usefulness of weakness patterns in business process redesign. In: ECIS 2012: 20th European conference on information systems
4. de Medeiros AKA, van der Aalst WMP, Weijters AJMM (2003) Workflow mining: current status and future directions. In: Meersman R, Zahir T, Schmidt DC (eds) OTM'03: On the move to meaningful internet systems 2003: CoopIS, DOA, and ODBASE. Lecture notes in computer science, vol 2888. Springer, Heidelberg, pp 389–406
5. de Medeiros AKA, Weijters AJMM, van der Aalst WMP (2007) Genetic process mining: an experimental evaluation. Data Min Knowl Discov 14(2):245–304
6. Dumas M, La Rosa M, Mendling J, Reijers HA (2013) Fundamentals of business process management. Springer, Heidelberg
7. Dunkl R, Binder M, Dorda W, Fröschl KA, Gall W, Grossmann W, Harmankaya K, Hronsky M, Rinderle-Ma S, Rinner C, Weber S (2012) On analyzing process compliance in skin cancer treatment: an experience report from the evidence-based medical compliance cluster (EBMC2). In: Ralyte J, Franch X, Brinkkemper S, Wrycza S (eds) CaISE'12: International conference on advanced information systems engineering. Lecture notes in computer science, vol 7328. Springer, Heidelberg, pp 398–413

8. Fahland D, van der Aalst WMP (2012) Simplifying discovered process models in a controlled manner. *Inf Syst* 38(4):585–605
9. Gabler Wirtschaftslexikon, Springer. <http://wirtschaftslexikon.gabler.de/Definition/business-process-reengineering.html>. Accessed 28 May 2014
10. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. *Mach Learn* 3(2):95–99
11. Grigori D, Casati F, Dayal U, Shan M-C (2001) Improving business process quality through exception understanding, prediction, and prevention. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT (eds) VLDB'01: International conference on very large daTa Bases. Morgan Kaufmann, San Francisco, pp 159–168
12. Günther C, Rinderle-Ma S, Reichert M, van der Aalst WMP, Recker J (2008) Using process mining to learn from process changes in evolutionary systems. *Int J Bus Process Integr Manag* 3(1):61–78
13. Hildebrandt T, Rinderle-Ma S (2013) Toward a sonification concept for business process monitoring. In: ICAD'13: International conference on auditory display
14. Hildebrandt T, Krügelstein S, Rinderle-Ma S (2012) Beyond visualization: on using sonification methods to make business processes more accessible to users. In: ICAD'12: International conference on auditory display
15. Jagadeesh Chandra Bose RP, van der Aalst WMP, Zliobaite I, Pechenizkiy M (2011) Handling concept drift in process mining. In Mouratidi H, Rolland C (eds) CaISE'11: International conference advanced information systems engineering. Lecture notes in computer science, vol 6741, pp 391–405. Springer, Heidelberg
16. Jagadeesh Chandra Bose RP, Mans RS, van der Aalst WMP (2013) Wanna improve process mining results? It's high time we consider data quality issues seriously. [BPMcenter.org](http://BPMcenter.org)
17. Ly LT, Rinderle-Ma S, Göser K, Dadam P (2012) On enabling integrated process compliance with semantic constraints in process management systems—requirements, challenges, solutions. *Inf Syst Front* 14(2):195–219
18. Ly LT, Maggi FM, Montali M, Rinderle-Ma S, van der Aalst WMP (2013) A framework for the systematic comparison and evaluation of compliance monitoring approaches. In: Gasevic D, Hatala M, Motahari Nezhad HR, Reichert M (eds) EDOC'13: International enterprise distributed object computing conference. IEEE, Los Alamitos, California, Washington, Tokyo, pp 7–16
19. Mansar SL, Reijers HA (2005) Best practices in business process redesign: validation of a redesign framework. *Comput Ind* 56(5):457–471
20. Mansar SL, Reijers HA (2007) Best practices in business process redesign: use and impact. *Bus Process Manag J* 13(2):193–213
21. Pflug J, Rinderle-Ma S (2013) Dynamic instance queuing in process-aware information systems. In: Shin SY, Carlos Maldonado, C (eds) SAC'13: annual ACM symposium on applied computing, enterprise engineering track. ACM, New York, pp 1426–1433
22. Reichert M, Weber B (2012) Enabling flexibility in process-aware information systems—challenges, methods, technologies. Springer, Heidelberg
23. Reichert M, Rinderle-Ma S, Dadam P (2009) Flexibility in process-aware information systems. In: Jensen K, van der Aalst WMP (eds) Transactions on Petri nets and other models of concurrency. Lecture notes in computer science, vol 5460, Springer, Heidelberg, pp 115–135
24. Reijers HA, Liman Mansar SL (2005) Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. *Omega* 33(4):283–306
25. Rinderle S, Reichert M, Dadam P (2004) Correctness criteria for dynamic changes in workflow systems—a survey. *Data Knowl Eng* 50(1):9–34
26. Rinderle S, Weber B, Reichert M, Wild W (2005) Integrating process learning and process evolution—a semantics based approach. In: van der Aalst WMP, Benatallah B, Casati F, Curbera F (eds) BPM'05: International conference on business process management. Lecture notes in computer science, vol 3649. Springer, Heidelberg, pp 252–267

27. Rinderle S, Bassil S, Reichert M (2006) A Framework for semantic recovery strategies in case of process activity failures. In: Manolopoulos Y, Filipe J, Constantopoulos P, Cordeiro J (eds) ICEIS'06: International conference on enterprise information systems, pp 136–143
28. Rinderle S, Reichert M, Jurisch M, Kreher U (2006) On representing, purging, and utilizing change logs in process management systems. In: Dustdar S, Fiadeiro SL, Sheth AP (eds) BPM'06: International conference on business process management. Lecture notes in computer science, vol 4102. Springer, Heidelberg, pp 241–256
29. Rozinat A, van der Aalst WMP (2008) Conformance checking of processes based on monitoring real behavior. *Inf Syst* 33(1):64–95
30. Rozinat A, de Medeiros AKA, Günther C, Weijters AJMM, van der Aalst WMP (2008) The need for a process mining evaluation framework in research and practice. In: ter Hofstede A, Benatallah B, Paik H-Y (eds) Business process management workshops. Lecture notes in computer science, vol 4928. Springer, Heidelberg, pp 84–89
31. van der Aalst WMP, Voorhoeve M (2014) Business process simulation. Lecture notes 2II75. [http://wwwis.win.tue.nl/~mvoorhoe/sim/ln2II75.pdf?origin=publication\\_detail](http://wwwis.win.tue.nl/~mvoorhoe/sim/ln2II75.pdf?origin=publication_detail). Accessed 24 June 2014
32. van der Aalst WMP, Nakatumba J, Rozinat A, Russell N (2010) Business process simulation. *Handb Bus Process Manag* 1:313–338
33. van der Aalst WMP, Pesic M, Song M (2010) Beyond process mining: from the past to present and future. In: Pernici B (ed) CaISE'10: International conference on advanced information systems engineering. Lecture notes in computer science, vol 6051. Springer, Heidelberg, pp 38–52
34. Weber B, Reichert M, Rinderle-Ma S (2008) Change patterns and change support features—enhancing flexibility in process-aware information systems. *Data Knowl Eng* 66(3):438–466
35. Weber B, Reichert M, Rinderle-Ma S, Wild W (2009) Providing integrated life cycle support in process-aware information systems. *Int J Coop Inf Syst* 18(1):115–165
36. Weijters AJMM, Ribeiro JTS (2011) Flexible heuristics miner (FHM). In: CIDM'11: IEEE symposium on computational intelligence and data mining. IEEE, Los Alamitos, California, Washington, Tokyo, pp 310–317
37. Weijters AJMM, van der Aalst WMP, de Medeiros AKA (2006) Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Technical Report WP 166
38. Weske M (2007) Business process management—concepts, languages, architectures, 2nd edn. Springer, Heidelberg

# Chapter 8

## Analysis of Multiple Business Perspectives

**Abstract** This chapter discusses analytical questions that arise at the interfaces between different BI perspectives, i.e., customer, production, and organization. In order to answer these questions, we introduce social network analysis as a new technique, adapt different analysis techniques introduced in the previous chapters to the analytical goals under consideration, and combine analysis techniques, for example, the combined application of cross-sectional analysis and process mining.

### 8.1 Introduction and Terminology

Let us first recall the different BI perspectives customer: production and organization as displayed in Fig. 8.1.

Chapters 5–7 introduce analytical techniques for cross-sectional, temporal, and process analysis. These techniques are mainly used to answer analytical questions regarding the customer and the production perspective. A possible example analysis question regarding the customer perspective answered by association analysis (cf. Sect. 6.5) reads as follows:

*Which products were bought in combination with buying a nail?*

An analytical question concerning the production perspective may be this:

*Do uploads take place after the corresponding milestone deadline?*

This question can be addressed based on process analysis methods such as process mining and compliance checking (cf. Sect. 7.5).

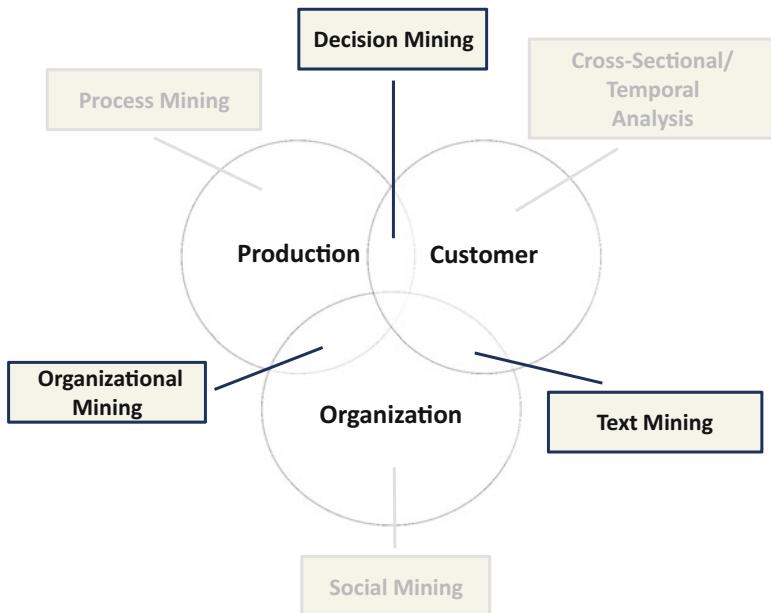
The organizational perspective and related questions have not been addressed in this book so far. We provide some insights on how organizational information is typically modeled and visualized in Chap. 4, Sect. 4.2.4, but analytical techniques dedicated to the organizational perspective have not been addressed yet.

In the case of the HEP use case (cf. Sect. 1.4.2), an analytical question regarding the organization could be:

*Which actors work on the same task in higher education processes?*

We can try to answer this question by using social network analysis (SNA). We will give an introduction into SNA in Sect. 8.2.1.

In addition to analytical questions that refer to a single perspective, questions that arise at the interfaces between the different perspectives can be of interest. An



**Fig. 8.1** Overview of analysis techniques, cf. Chap. 1.3

example question arising at the interface between the organization and production perspectives in the HEP use case is

*Which roles do participate in the higher education process?*

In order to answer this question, a combined application of process mining and social network analysis can be employed as discussed in Sect. 8.2.2.

A question arising at the interface between the production and customer perspectives in the logistics use case (cf. Sect. 1.4.3) is:

*For which containers did the vehicle return to the origin?*

Section 8.3 introduces decision mining as a combined application of cross-sectional analysis and process mining that can be used to answer the above question.

Finally, an example question arising at the interface between the customer and organization perspectives is:

*What is the opinion about the Oscar nominations of Jennifer Lawrence when compared to Sandra Bullock?<sup>1</sup>*

This question can be tackled by applying text mining as a combination of data mining and social mining (cf. Sect. 8.4). More precisely, we can apply *opinion mining* (Sect. 8.4.5) which employs text mining techniques.

Section 8.5 focuses on evaluation and assessment methods for combined analysis techniques.

---

<sup>1</sup>Compare the “social mentions” on the Oscar website <http://oscar.go.com/nominees>.

## 8.2 Social Network Analysis and Organizational Mining

This section presents methods to analyze the organizational perspective (cf. Fig. 8.1). At first, an introduction to social network analysis is provided, followed by the application in the context of business processes.

### 8.2.1 Social Network Analysis

Social network analysis (SNA) aims at describing and analyzing the relationships between *social entities* such as actors and their roles or organizations. In addition, SNA is interested in finding “patterns and implications of these relationships” (see [37]), for example, starting from a certain person and including friends of friends. Relationships (or relations for short) can be of different types, for example, *working together* or *is friend of*. Two actors who are connected by a relation in the social network form a so-called *dyad*; three actors form a so-called *triad*. SNA provides the basis for answering questions from different areas such as business, social sciences, or biology [37].

In order to enable a meaningful and focused analysis, we have to identify which social entities are of interest and which kind of relation between these entities should be used. Both decisions depend on the data. If the data basis becomes very large, “the boundedness of social relations and the possibility of drawing samples” from possible social entities is crucial (see [33]). Think, for example, of Twitter or Facebook data, which is huge. The relationships between the entities may be defined explicitly or have to be derived from other relations in the database. In the case of social networks, the data often directly reveal a social relation between the entities, for example, “*is friend of*.” Other data sources include more general relations between entities, such as “*are involved in the same event*,” and the relationship has to be defined according to the precise formulation of the analytical goal.

The main steps of SNA are summarized in the following analysis template.

**Template: Social Network Analysis**

- **Relevant Business and Data:** A database containing information about social entities together with relations between these entities.
- **Analytical Goals:**
  - Visualization of the relations between the entities
  - Describing the relationships by summary measures
  - Finding patterns in the relationships
- **Modeling Task:** Generate from the data, first of all, a *data matrix* which defines a graph with the social entities as nodes and the relations between the entities as edges. This graph defines the analytical model and is henceforward called *sociogram* [33].

- **Analysis Task:** Analyze the sociogram using different metrics which allow quantification of the relationship between the entities.
- **Evaluation and Reporting Task:** Visualize the sociogram and represent the descriptive measures. Using an explorative approach, possibly supported by tools, the visualization can be the basis for the interpretation of the analysis results. It can be useful to visualize the sociograms together with different metrics in different layouts. In particular at the presence of large social networks or sociograms, a visualization should convey the analysis results in an understandable and interpretable way.

## Modeling Task

The basis for SNA is the representation of the *social network* as sociogram. From the graph structures introduced in Sect. 2.3, the most important ones for the context are summarized in the following overview.

### Model Structures for Social Network Analysis

- *Undirected graphs:* As introduced in Sect. 2.3, an undirected graph  $G$  is defined as  $G = (V, E)$  with set of nodes  $V$  and set of undirected edges  $E$ .
- *Directed graphs:* Opposed to undirected edges, directed edges establish a relation that reflects a causal relation or a relation that is directed from one to another entity.
- *Weighted graphs:* It can be also useful to assign weights to the edges in the graph, i.e., a weight  $w(e)$  expressing some kind of quantitative measure for the relation.
- *Connected subgraphs:* Special connected subgraphs might be of interest. A subgraph consisting of two nodes (with or without relations between them) describes a dyad and a subgraph consisting of three nodes of interest, a triad.

In the case of undirected graphs, an edge  $(v_1, v_2)$  between the nodes  $v_1, v_2 \in V$  means that the relation between the entities associated with nodes  $v_1$  and  $v_2$  exists in both directions. For example, the entity associated with  $v_1$  works with the entity associated with  $v_2$  and vice versa. In the case of directed graphs, the interpretation of a directed edge  $(v_1, v_2)$  may be that the entity associated with  $v_1$  hands over work to the entity associated with  $v_2$ . Weighted graphs are useful for quantifying the strength of a relationship. For example, a directed edge  $e = (v_1, v_2)$  with assigned weight  $w(e) = 3$  could mean that the entity associated with  $v_1$  has handed over work to the entity associated with  $v_2$  for three times.

With respect to the representation of the graph, one can use either an adjacency matrix or an incidence matrix defined in the modeling task by extracting information from the existing data about the business. The main challenge in modeling is the selection of the entities and the definition of the social relations. Let us illustrate the modeling task by building the sociogram in connection with the higher education use case.

**HEP Use Case: Definition of a Sociogram**

In the teaching processes of the HEP project, different actors are involved, i.e., students, lecturers, tutors, and the system (learning platform). Assume that we are interested in the following question:

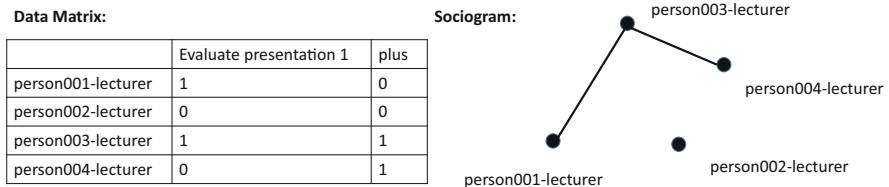
*Which actors work together in higher education processes?*

In order to answer the question, we decide that the nodes of the sociogram will reflect the actors. For establishing the relations in the sociogram recall that the HEP data is provided as process logs. From these logs, we can find out, for example, which actors worked on the same tasks. Consider the following log fragment consisting of entries of 4 activity executions. The Originator field contains the information which actor performed an activity denoted by the WorkflowModelElement, e.g., actor person001-lecturer performed activity Evaluate presentation 1.

```

<AuditTrailEntry>
  <WorkflowModelElement>Evaluate presentation 1</WorkflowModelElement>
  <EventType>complete</EventType>
  <Timestamp>2008-10-29T23:59:00.000+01:00</Timestamp>
  <Originator>person001-lecturer</Originator>
</AuditTrailEntry>
<AuditTrailEntry>
  <WorkflowModelElement>Evaluate presentation 1</WorkflowModelElement>
  <EventType>complete</EventType>
  <Timestamp>2008-10-28T23:59:59.000+01:00</Timestamp>
  <Originator>person003-lecturer</Originator>
</AuditTrailEntry>
<AuditTrailEntry>
  <WorkflowModelElement>plus</WorkflowModelElement>
  <EventType>complete</EventType>
  <Timestamp>2008-11-12T00:00:00.000+01:00</Timestamp>
  <Originator>person003-lecturer</Originator>
</AuditTrailEntry>
<AuditTrailEntry>
  <WorkflowModelElement>plus</WorkflowModelElement>
  <EventType>complete</EventType>
  <Timestamp>2008-11-05T00:00:00.000+01:00</Timestamp>
  <Originator>person004-lecturer</Originator>
</AuditTrailEntry>
```

Based on the log fragment, we can see that activity Evaluate presentation 1 is performed by actors person001-lecturer and person003-lecturer and activity plus by actors person003-lecturer and person004-lecturer. The corresponding incidence matrix and sociogram are shown in Fig. 8.2. The sociogram includes undirected edges between nodes person001-lecturer and person003-lecturer as well as person003-lecturer and



**Fig. 8.2** HEP Data: sociogram with three entities (actors) and relations “working on same task” for process activities evaluate presentation 1 and plus

person004-lecturer. Details on the HEP project can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

### Analysis Task

For the analysis of sociograms, different metrics can be calculated that reflect different social relations. We can, for example, determine whether a social network is *dense*, i.e., many interactions between the entities within the network take place. Other metrics refer to single entities or groups of entities instead of the entire network. For a single entity, it can be of interest whether it is a *central* entity within the network, i.e., many interactions occur with involvement of the entity. In this book, we cannot cover all measures on sociograms that are used for SNA. We will introduce intuitive measures that enable an understanding of the basic principles of SNA and its application in organizational mining (cf. Sect. 8.2.2). For an overview on further measures, we refer the interested reader to [37]. In the following, we present *local* and *global* measures, i.e., measures that refer to single entities (nodes) within the network (local) or to the entire social network (global).

#### Local Measures

One measure for the interactions of an entity, reflected by a node  $n$ , is the degree of  $n$ , i.e., the number of adjacent edges to  $n$  in an undirected graph and the number of edges starting at  $n$  (out-degree) or ending in  $n$  (in-degree) in a directed graph (see Sect. 2.3 for the definition of degree, in-degree, and out-degree). The degree could reflect, for example, how important or popular the associated entity is. The intuition behind is that a central entity is “well-connected” within the sociogram. The degree of node person003-lecturer is 2 based on the sociogram shown in Fig. 8.2.

The question is whether this absolute measure is meaningful, i.e., relative measures such as degree over number of all nodes or degree compared to degree of the other nodes might be more helpful. Think, for example, of a node having a degree of 2 in a sociogram of 4 nodes (as in Fig. 8.2) compared to a node having a degree of 2 in a sociogram of 10,000 nodes. Hence, for SNA, the measure of *degree centrality* can be calculated in an absolute and relative manner. Relative means to divide the number of adjacent nodes to a node of interest by the overall number of

nodes. The degree centrality of node `person003-lecturer` in Fig. 8.2 is 2 (absolute) and  $\frac{1}{2}$  (relative).

Degree centrality can be extended to counting paths<sup>2</sup> of a given length that originates from the node of interest (*k-path centrality*). In the example shown in Fig. 8.2, nodes `person001-lecturer` and `person002-lecturer` are connected by a path of length 2. Imagine an additional edge between `person003-lecturer` and `person002-lecturer`. For `person001-lecturer`, the degree centrality would still be 1, but the 2-path centrality would be 2 as `person001-lecturer` is connected to `person002-lecturer` and `person004-lecturer` via paths of length 2.

Another measure is the *closeness* of a node to the other nodes in the sociogram [33]. The closeness of a node of interest can be determined by counting in how many shortest paths<sup>3</sup> occur between two nodes. The length of the shortest path between two nodes is also referred to as the *distance* between them.

### Global Measures

*Density* is a property of the entire sociogram. It measures how many edges are present in the sociogram compared to the complete graph on the same set of nodes. Intuitively, a complete graph contains all possible edges between the existing nodes. The formula for calculating the density `dens` of an undirected graph  $G = (V; E)$  is

$$\text{dens}(G) := \frac{2 * |E|}{|V| * (|V| - 1)}. \quad (8.1)$$

Consider the sociogram depicted in Fig. 8.2. The density of this graph turns out as  $\frac{1}{3}$ .

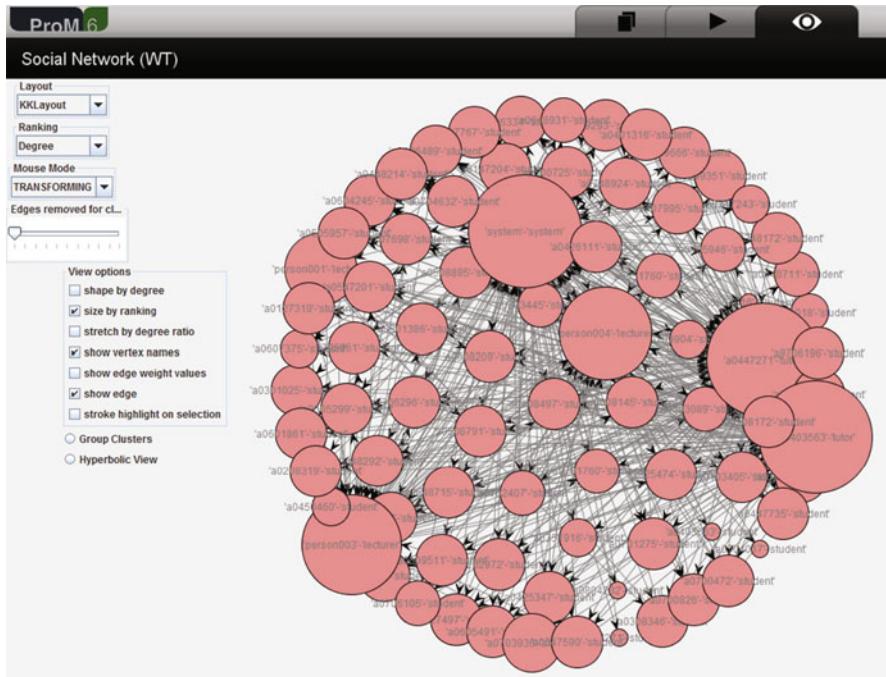
As a variant to density on the complete sociogram, *egocentric* measures of density can yield different insights. They are particularly interesting for analyzing the relations between the direct contacts of a given entity also called *ego* [33]. An interesting study about the implications of density is [38]. In summary, this study relates the density of personal networks to the family relations between members of the networks. The result is that the higher the density of the network, the higher is the percentage of people that are kin.

### Evaluation and Reporting Task

Visualizations of the sociogram and the calculated measures support the interpretation of the SNA results. They empower the user to analyze the SNA results in an explorative manner, possibly supported by tools [4]. There are several techniques for visualizing sociograms, for example, circular layout, Kamada–Kawai layout

<sup>2</sup>A path between two nodes  $v, w \in V$  consists of a set of (directed) edges connecting  $v$  and  $w$ . The length of a path is the number of edges it consists of, cf. Sect. 2.3.

<sup>3</sup>The shortest path between two nodes is the one with minimal length. Several algorithms exist to calculate the shortest paths between two or all nodes in a (directed) graph, for example, Dijkstra or Floyd–Warshall algorithm respectively, [16].



**Fig. 8.3** HEP: sociogram using KKLlayout, measure degree, and vertex size by ranking (using ProM 6.2)

(KKLayout) [17], or Fruchterman-Reingold layout [13]. Tools, such as Pajek<sup>4</sup> and ProM offer different options for laying out the sociograms. In this section, we do not dig deeply into visualization techniques, but will present different visualizations of the HEP data set in the following. Figure 8.3 shows the sociogram for the full HEP data set. It is produced by using ProM 6.2. We chose the KKLlayout for the sociogram depicted in Fig. 8.3. It uses the degree of vertices as measures to determine their size.

### 8.2.2 *Organizational Aspect in Business Processes*

After introducing some SNA fundamentals, we explain in this section basics on organizational modeling regarding business processes.

In general, business processes do not only consist of process activities and their order relations but also incorporate aspects such as the data flow between

---

<sup>4</sup><http://pajek.imfm.si/doku.php?id=download>.

process activities, the assignment of actors based on organizational structures, and invoked services (application components). All these aspects are of particular importance for the automation and execution of business processes within a Process-Aware Information Systems (PAIS). At runtime, process activities are executed by human actors or invoke internal or external services that might exchange data. Typically, PAIS log which user has worked on which task. Consider, for example, the following MXML fragment for which the `Originator` tag shows that actor Luke performed the activity registration.

In the following, we present a set of techniques that refer to mining organizational structures from event logs (given that the `Originator` field is provided).

```
<AuditTrailEntry>
  <WorkflowModelElement>registration</WorkflowModelElement>
  <EventType>start</EventType>
  <Timestamp>2011-01-01T01:00:00.000+01:00</Timestamp>
  <Originator>Luke</Originator>
</AuditTrailEntry>
```

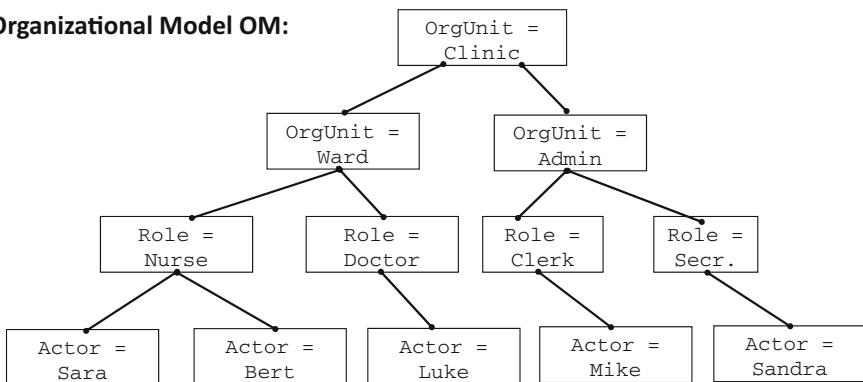
In general, process activities are not directly assigned to persons. Instead, organizational entities, such as *role* or *organizational unit*, are used as an abstraction layer. These organizational entities are captured and modeled by organizational models, often, based on role-based access control (RBAC) models [36]. Roles as well as organizational units can be related to actors. In addition, roles and organizational units can be also related in a hierarchical manner, i.e., expressing that one role is a subrole or junior role of another role. There exist several extensions to basic RBAC models, for example, capturing entities such as abilities or teams.

Figure 8.4 displays an example for an organizational model in a clinic. On top, organizational unit `Clinic` consists of organizational units `Ward` and `Admin`. Two roles are assigned to organizational unit `Ward`, i.e., `Nurse` and `Doctor`. Two actors have the role `Nurse`, i.e., `Sara` and `Bert`.

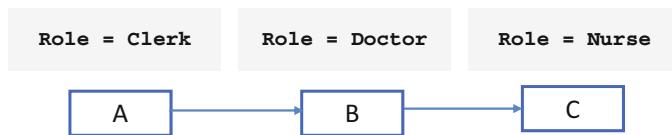
At the bottom of Fig. 8.4, process model PM refers to organizational model OM, i.e., the tasks in PM will be assigned to actors captured by OM. This is achieved by formulating a so-called actor assignment for a task. For task C, for example, we would like to express that only actors having the role `Nurse` can work on C. This is reflected by actor assignment `Role = Nurse`. At the process model level, this means that `Sara` and `Bert` can work on C.

Imagine now that process instances are created and executed based on process model PM. Then, during runtime, the actor assignments are resolved to the set of actors qualifying for the assignment. In the case of task C, `Sara` and `Bert` qualify. Hence, C is offered to both actors in their work list. Typically, the actor who first selects the task to work on is finally assigned to it. In turn, this task is then removed from the work list of all other actors.

### Organizational Model OM:



### Process Model PM:



**Fig. 8.4** Medical example: organizational model and process model with actor assignments

### 8.2.3 *Organizational Mining Techniques for Business Processes*

In this section, we describe how SNA techniques can be applied in combination with process mining techniques in order to address analysis questions at the interface between organization and production. The following paragraphs illustrate a selection of organizational mining techniques for business processes based on [34] and the associated implementation in ProM 5.2.<sup>5</sup> We opt for the version 5.2 of the ProM framework as it provides the implementations of the organizational and role hierarchy miner.

#### Organizational Mining

Organizational mining tries to derive the working behavior behind a process from event logs. The approach creates profiles for each actor in counting how many times an actor has performed a certain task. Based on the profiles, several distance metrics can be calculated addressing questions such as “who is doing similar tasks” or “who is working with whom” [34]. In turn, this information can be used to establish a suggestion for the organizational structure.

<sup>5</sup><http://promtools.org/prom5/>.

*Example 8.1 (Organizational Mining in Medical Example)* Consider, for example, that activities A, B, and C of the process depicted in Fig. 8.4 were executed for ten instances by the three actors Mike, Luke, and Sara. The corresponding log for one instance would look like the following:

```
...
<AuditTrailEntry>
  <WorkflowModelElement>A</WorkflowModelElement>
  <EventType>start</EventType>
  <Timestamp>...</Timestamp>
  <Originator>Mike</Originator>
</AuditTrailEntry> <AuditTrailEntry>
  <WorkflowModelElement>B</WorkflowModelElement>
  <EventType>start</EventType>
  <Timestamp>...</Timestamp>
  <Originator>Luke</Originator>
</AuditTrailEntry> <AuditTrailEntry>
  <WorkflowModelElement>C</WorkflowModelElement>
  <EventType>start</EventType>
  <Timestamp>...</Timestamp>
  <Originator>Sara</Originator>
</AuditTrailEntry>
```

For all ten executed instances, the actor profiles turn out as:

Mike (10, 0, 0), Luke (0, 10, 0), and Sara (0, 0, 10),

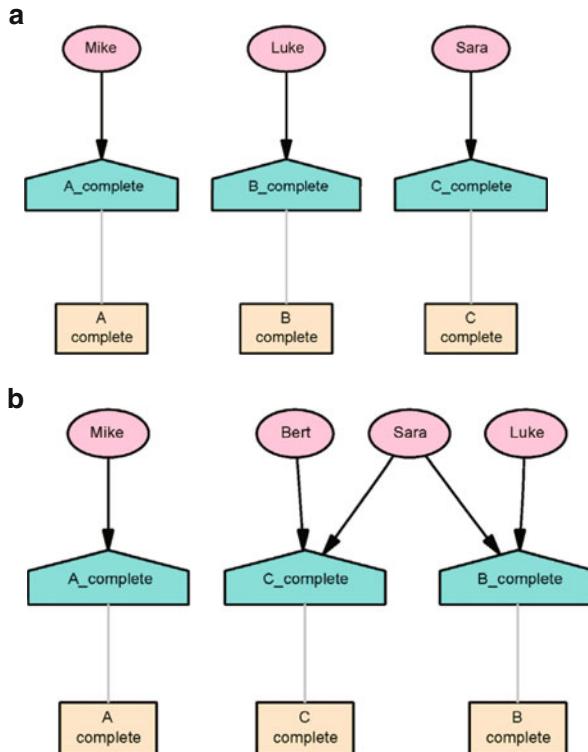
meaning that Mike has worked on A, Luke on B, and Sara on C for all ten instances. Note that this means that for some reason, Bert has never executed task C even though he qualifies for it.

For subsequent analysis, these profiles can be transformed into the following representation:

Mike:  $10 \times (1,0,0)$ , Luke:  $10 \times (0,1,0)$ , and Sara:  $10 \times (0,0,1)$ ,

The second representation builds the basis for cluster analysis (cf. Sect. 5.4), i.e., we interpret each execution as a case and want to find clusters of persons having similar work profiles. Usually, one uses hierarchical clustering or  $k$ -means clustering based on distances between the profiles, e.g., Hamming distance, Euclidean distance, or distances based on Pearson's correlation coefficient. These cluster solution defines profiles, and the distance between the clusters can be interpreted as the distance between the working profiles.

Obviously, a hierarchical cluster analysis of the working profiles in the above example would lead to a solution with three clusters and the profiles have maximal distance, i.e., no one is *working together*. These clusters are afterwards identified as three organizational units by the organizational miner and visualized by a bipartite graph (cf. Sect. 2.3.2). The result is depicted in Fig. 8.5a.



**Fig. 8.5** Results of organizational mining on the medical example (using the *organizational miner* in ProM 5.2). **(a)** Organizational Miner, Profiles Mike(10,0,0), Luke(0,10,0), Sara(0,0,10) and **(b)** Organizational Miner, Profiles Mike(10,0,0), Bert(0,0,5), Luke(0,5,0), Sara(0,5,5)

*Example 8.2 (Organizational Mining in Medical Example with Modifications)*  
 Assume now a modification of Example 8.1 with profiles:

Mike(10,0,0), Luke(0,5,0), Bert(0,0,5), and Sara(0,5,5).

Applying *organizational miner* results again in three clusters depicted in Fig. 8.5b. As Luke and Sara worked on task B and Bert and Sara worked on task C, Sara has now been assigned to two clusters. One assignment is in agreement with the role model and the other one in disagreement. This result can be interpreted in three ways. At first sight, this could mean that Sara has both roles Nurse and Doctor that is not reflecting the original model. As a second interpretation, task B could have been assigned to both roles Nurse and Doctor by a corresponding actor assignment:

B  $\leftarrow$  Role='Doctor' AND Role='Nurse'

This is not reflected by the original actor assignment either. As a third interpretation, during process execution, actors deviated from the original assignments due to, for example, substitution or emergency cases [30]. Such workarounds are neither

reflected by the organizational nor the process model (including actor assignments), but can be detected by techniques such as organizational mining.

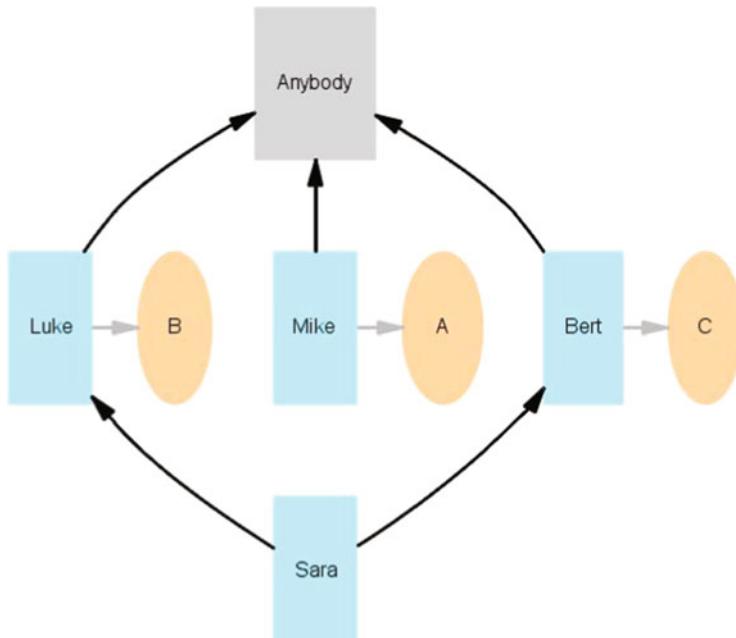
### Role Hierarchy Mining

In practical settings, the role structure within the organizational model is not “flat,” i.e., roles might be ordered in some hierarchical relation to each other. More precisely, a “more powerful” role can be defined as a *senior* role to a “less powerful” *junior* role [32]. More precisely, the senior role inherits all permissions of its junior roles and might have additional permissions.

Role hierarchy mining aims at finding these senior-junior role relations based on logs and the contained originator information. As for organizational mining, the user profiles build the basis for determining the role hierarchy.

*Example 8.3 (Role Hierarchy Mining in Medical Example)* Consider the user profiles provided in Example 8.2. Using the *role hierarchy mining*, plug-in of ProM 5.2 results in the role hierarchy depicted in Fig. 8.6. As Sara has worked on the same tasks as Luke and Bert, it can be assumed that Sara has a senior role related to the roles of Luke and Bert.

Another organizational mining technique is *staff assignment mining* [23], i.e., deriving the actor assignments connecting process tasks to organizational units. Note that this technique does not only require a log as input but also an existing



**Fig. 8.6** Results of role hierarchy mining on medical example (using ProM 5.2)

organizational model. The algorithm itself uses decision trees to separate the set of actors (from the log) along the organizational entities (from the organizational model). Hence, the approach constitutes an example for combined application of analysis techniques, i.e., process mining and data mining. In other words, staff assignment mining applies the classification of actors along the organizational entities.

### Social Network Mining

Beyond deriving organizational structures, it can also be of interest how the actors participating in the process execution interact with each other. Interaction can mean that they directly work together, hand over work to each other, or work on similar tasks. Such analyses can be performed by using techniques from social network mining (cf. Sect. 8.2.1) on the process execution logs. These logs yield two kinds of input information, i.e., the user profiles and the underlying process structure. The process structure is important for determining the relations between the actors. If, for example, two actors are only working on tasks that are ordered in parallel, they do not work together.

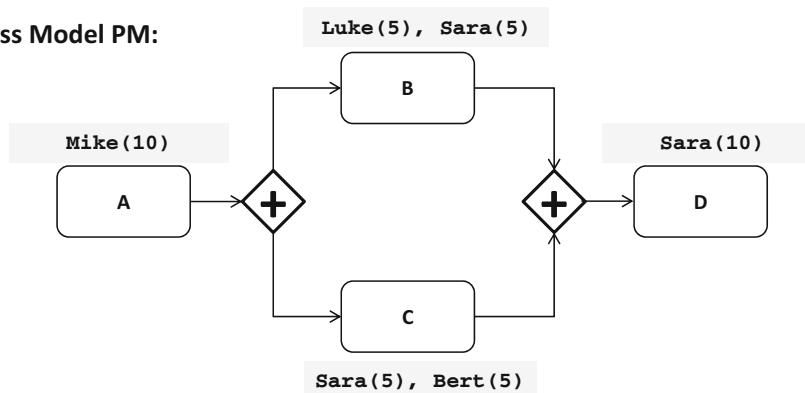
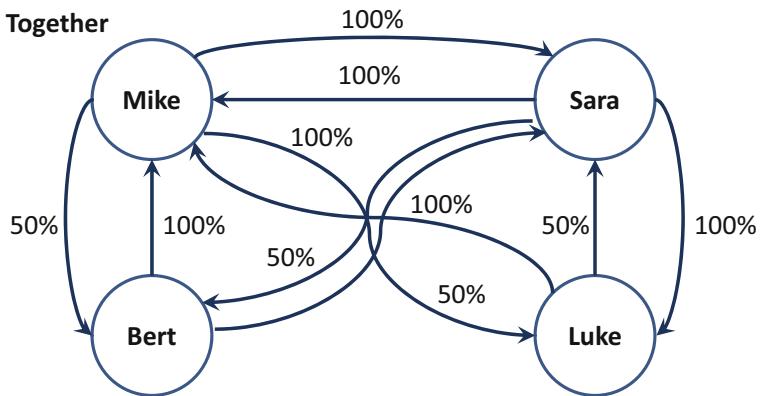
*Example 8.4 (Social Network Mining on the Medical Example)* Consider the process displayed in Fig. 8.7. In this process, the tasks B and C are ordered in parallel. Sara works on both B and C for five times each, Luke works on B for five times, and Bert works on C for five times. A and D are performed by Mike and Sara ten times each. Using the Social Network Miner of ProM 5.2. the result for analysis *working together* is also shown in Fig. 8.7. We can conclude that based on the profiles and the process structure, Mike works together with, for example, Luke in 50 % of the cases, whereas Mike works with Sara in 100 % of the cases. It can be also seen that Bert and Luke do not work together directly as there is no edge connecting the nodes reflecting Bert and Luke.

The resulting visualization of the social network based on the relation *working together* is shown on the bottom of Fig. 8.7. As expected, the relationship between Mike and Sara states that Sara takes over 100 % of the work from Mike and vice versa. There is no relationship between Bert and Luke. Bert takes over 100 % of his work from Mike, and Mike hands over 50 % of his work to Bert. The other 50 % is handed over to Luke.

So far, we have illustrated existing techniques for organizational mining in business processes as an example for combined analysis. The examples were rather simple and abstract. Hence, we provide some results of using the social network miner and the role hierarchy miner to the higher education process data in the following.

HEP Use Case: Visualization of a Sociogram

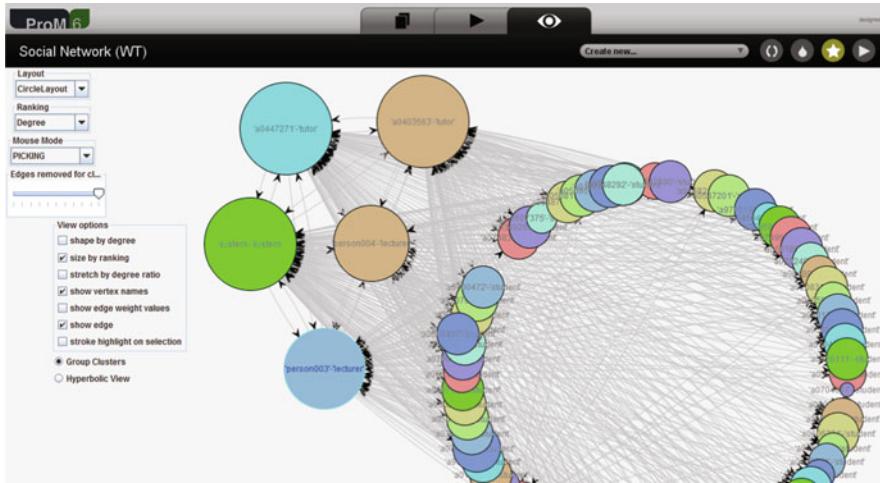
Figure 8.8 depicts the result of using the *social network miner*, *working together* (ProM 6.3) to the higher education data. Statistics show that the process was executed 74 times, producing 4,018 events and involving 80 originators. The application of social network mining to

**Process Model PM:****Working Together**

**Fig. 8.7** Process model with actors and the number of task executions; result of the application of social network mining using the relationship *working together*

the log produces the result displayed in Fig. 8.8. Obviously, teaching staff including actors 'a0403563'-'tutor', 'a0447271'-'tutor', 'person003'-'lecturer', and 'person004'-'lecturer' are *working together*, i.e., communicates often with the system, i.e., the online learning platform. Note that some of the edges have been removed as their weight does not exceed a certain threshold. Details on the HEP project can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)



**Fig. 8.8** HEP: Social network mining (using ProM 6.3)

### 8.2.4 Summary: Social Network Analysis and Organizational Mining

Social network analysis provides means to analyze BI perspective organization. The analysis model is a sociogram that consists of entities reflected as nodes and relations between them reflected by edges. These relations can be analyzed based on different metrics, either for the entire sociogram (e.g., density) or for single nodes (e.g., centrality).

Organizational mining aims at addressing analysis questions at the interface between BI perspective organization and production by an application of social network analysis techniques on process logs. An analysis of the organizational perspective can be very valuable for enterprises in order to, for example, detect anomalies in user behavior. Recently, the mining of role-based access control (RBAC) models over event logs for anomaly detection, policy maintenance, and policy specification errors has gained interest, not only in the business process community but also in, for example, security and RBAC applications [18, 26].

## 8.3 Decision Point Analysis

Business processes unfold aspects beyond the control flow that can be subject to analysis and mining such as organizational structures or process data. How to analyze and mine organizational structures from process logs have been explained in Sect. 8.2.2, aiming at analysis questions at the interface between organization and

production. In this section, we show how analysis questions at the interface between BI perspectives customer and production can be tackled by the combined analysis of process instance data (cross-sectional analysis) and process mining.

The related technique is called *decision mining* or *decision point analysis* (DPA) introduced by Rozinat and van der Aalst [31]. DPA aims at deriving the decision rules that are connected with decision points in a business process. The essence is to find rules based on the data values that have triggered the different decisions at process runtime along the paths chosen and stored in the event log.

DPA is illustrated by means of the logistics use case:

**Logistics Use Case: Mining a Process Model**

Consider the container transportation process as illustrated by Fig. 8.9. Obviously, this process contains a decision point, i.e., after execution process activity Move to D, it is decided whether the vehicle unloads at D or moves to P. We can also see that the decision depends on the value of data element ContainerTemperature. Based on the process model, we created 25 synthetic logs simulating different values for ContainerTemperature. The following log fragment displays how the values for data element ContainerTemperature are stored within the log. In this case, container temperature was logged with a value of 55.

```
...
<AuditTrailEntry>
    <WorkflowModelElement>Move to D</WorkflowModelElement>
    <EventType>completed</EventType>
    <Timestamp>2013-02-27T10:29:06.373+01:00</Timestamp>
    <Originator>unknown</Originator>
    <Data>
        <Attribute name="ContainerTemperature">55</Attribute>
    </Data>
</AuditTrailEntry>
```

We applied the  $\alpha$ -algorithm (see Sect. 7.4) using ProM 5.2 to the logs. One can see that the resulting Petri net (cf. Fig. 8.10) reflects the decision point of the original model by the place between the transitions reflecting process activities Move to D and Move to P and Unload at D.

The question for DPA is to find the rule behind this decision, for example, “for a container temperature above 30 °C, return to P; otherwise, continue and unload at D.” Details on the logistics use case can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

DPA works as follows: first of all, the structure of the underlying process is determined by applying process mining. If the process structure contains a decision point, this means that the set of all process execution paths can be discriminated along this decision. Decisions in business processes are mainly based on values of

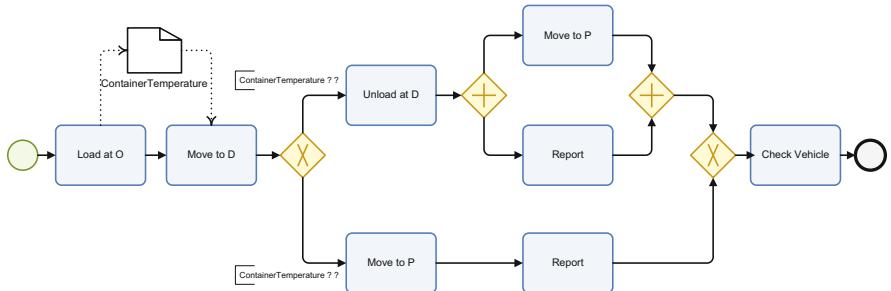


Fig. 8.9 Container transportation use case (based on [27])

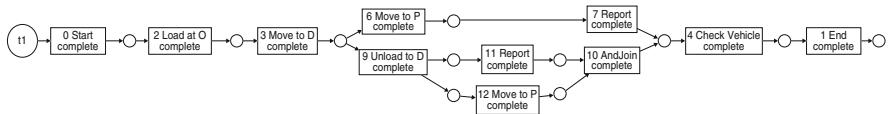


Fig. 8.10 Result of applying the  $\alpha$ -algorithm to the container transportation example (using ProM 5.2)

process data. The analysis question is to determine how the paths can be classified based on the data.

For doing so, DPA employs decision trees (introduced in Sect. 5.3.3). More precisely, it is determined for which data values the distinction into the different execution paths becomes significant.

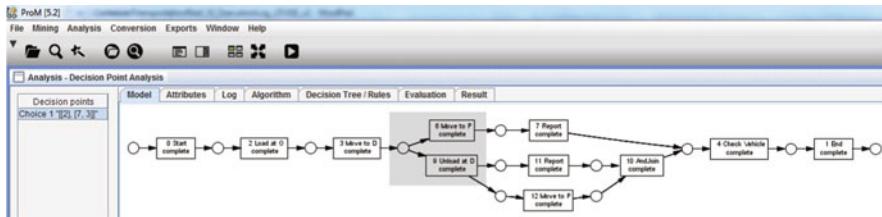
#### Logistics Use Case: Decision Point Analysis

As the first step of DPA, we applied the  $\alpha$ -algorithm to the 25 container transportation logs resulting in the Petri net model shown in Fig. 8.10. This model contains a decision point which we want to explain. Hence, in a second step, we employ the DPA plug-in (ProM 5.2). In Fig. 8.11, we can see that the DPA plug-in determines the decision point (shaded) within the Petri net model. DPA invokes the Weka

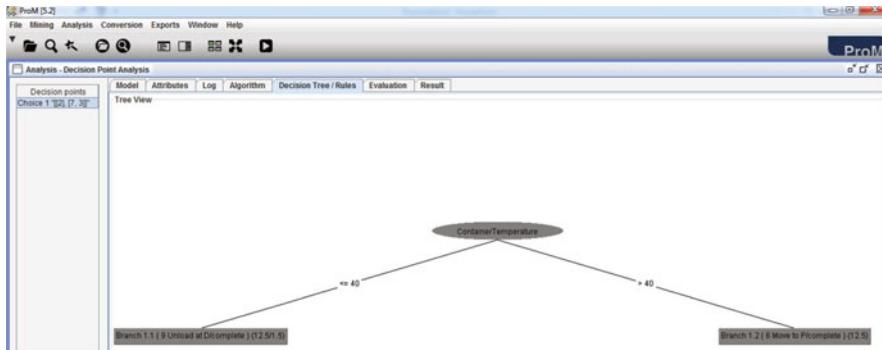
Data mining software in order to run decision tree analysis on the process data connected with the decision point. Figure 8.12 shows the corresponding decision tree. It reflects a significant distinction of execution traces at a container temperature value below or greater or equal to 40. Finally, in Fig. 8.13, we see the integration of the decision rule into the underlying process model. In detail, for a container temperature below 40, the vehicle moves to P. Otherwise (container temperature  $\geq 40$ ), the vehicle unloads at D. Details on the logistics use case can be found on the homepage of the book:

[www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com)

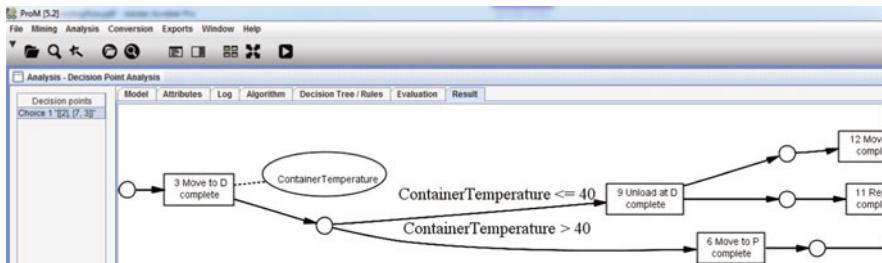
DPA offers a promising approach for combining different analysis methods, i.e., cross-sectional analysis and process mining, in a staged approach in order to address additional analysis questions. As a repetition, in the container transportation



**Fig. 8.11** Finding a decision point with DPA (using ProM 5.2)



**Fig. 8.12** Determining the decision tree (using ProM 5.2)



**Fig. 8.13** Determining the decision rule (using ProM 5.2)

example, the container can be perceived as the customer of the process. By applying process mining, it is revealed how the container transportation works (production perspective). Applying additional cross-sectional analysis shows how the customer, i.e., the container, determines the choice of the process execution paths.

DPA can be extended in different ways. For example, de Leoni et al. [9] suggest different techniques to derive more complex decision rules containing more than one variable and possibly connected by arithmetic operators.

In [11], DPA has been also extended by considering time series process data, i.e., data that is not written once a process activity completes, but during process execution in a continuous way. An example is the continuous measurement of

container temperature during the time a vehicle is moving from its origin to its destination. The basic idea is to apply process mining and cross-sectional analysis in an iterative way. The results of the cross-sectional analysis within one iteration are captured within a complex variable. The complex variable can then be used as input for DPA. If DPA results in a significant discrimination, the complex decision rule can be unfolded from the complex variable again.

## 8.4 Text Mining

The Internet offers a plethora of data and applying text mining for extracting useful information from such data for business decisions has become more and more important in the last 15 years. Text mining combines ideas of different scientific areas which have worked with textual information for a long time. The most important ones are databases and information retrieval, computational linguistics, artificial intelligence, machine learning, statistics, and data mining. We will focus in this section on a text mining approach using mainly analytical techniques from data mining and statistics. Section 8.4.1 introduces a data structure frequently used in this approach towards text mining and identifies the analytical goals treated in detail in this section. Data preparation and modeling for text data is discussed in Sect. 8.4.2. Section 8.4.3 considers analysis techniques for descriptive goals and Sect. 8.4.4 analysis techniques for clustering, classification, and understanding of text data. In Sect. 8.4.5, we briefly discuss other approaches and applications, in particular opinion mining.

### 8.4.1 *Introduction and Terminology*

The basic entity for our considerations are text documents. These text documents may be of different origins and can occur in various formats. Typical sources for documents are reports, abstracts, blogs, tweets, emails, journal articles, or notes stored in a database. Corresponding to the variety of sources, the documents are stored in miscellaneous formats, for example, PDF, MS-Word, HTML, or XML. Text mining software offers readers for the different formats and converts the document into a simple text format together with some metadata. These document-specific metadata cover entries which are useful for search and retrieval of text documents and getting basic information, e.g., the author and title of the document, the content of the document, the creation date, the access rights, and how to access the documents. A standard representation of such information is a structure proposed by the Dublin Core Metadata Initiative<sup>6</sup> (DCMI). This standard has its

---

<sup>6</sup><http://dublincore.org/about-us/>.

origin in the library science and is based on 15 tags which allow a unified description of Web resources, books, CDs, and objects like artwork. Such metadata describe documents by well-structured data and allow application of the analysis techniques presented in the previous chapters. For example, one can use basic descriptive analysis for the metadata, one can retrieve documents in the corpus according to specific properties of metadata, or one can use methods of process analysis for exploring threads in a discussion.

In the following, we will not pursue metadata approaches in detail but focus on analytical goals which look at the entire text in the documents. Moreover, in most cases, we are not interested in one single document but in analyzing a collection of documents, called a *corpus*. The following overview box lists the goals treated in this section in some detail. Besides these goals, there are a number of other goals in text mining which will be briefly reviewed in Sect. 8.4.5.

### Analytical Goals in Text Mining

- *Descriptive goals:* Corpus description of the contents of the documents based on word frequencies.
- *Segmentation and understanding goals:* Find clusters of documents which are similar with respect to content and identify the topics in these clusters.
- *Classification goals:* Learning of a classification rule from already classified documents which allows the classification of a new document in one of the classes.

The model we use for the analysis is the document term matrix, which will be introduced in Sect. 8.4.2. The main emphasis in this section is on the necessary data preparation activities, taking into account the peculiarities of text data. Section 8.4.3 considers techniques for the descriptive analysis and Sect. 8.4.4 considers methods for document understanding. The demonstration of these methods is done by using a corpus defined by earlier versions of the 33 subsections of Chaps. 1 and 2 of this book.

In order to achieve the analytical goals, we follow the analysis steps described in Chap. 1, which are summarized in the text mining template.

#### Template: Text Mining for a Corpus

- **Relevant Business and Data:** A text corpus defined by a collection of text documents
- **Analytical Goals:**
  - Description of the documents in the corpus
  - Clustering the documents in the corpus
  - Finding topics of the corpus
  - Classification of documents based on rules derived from a training corpus
- **Modeling Task:** Definition of a document term matrix

- **Analysis Task:**

- *Description of Corpus:* Determination of type-token relation and association measures; visualization of the content in the corpus using word clouds and correlation plots
- *Clustering documents:* Use cluster analysis methods for clustering the documents
- *Topic Models:* Define a number of topics and find the probability of assignment of the documents to the topics
- *Classification:* Learn classification rules for assignment of new documents
- **Evaluation and Reporting Task:** Represent the results of the analysis by word clouds, by correlation plots and by characterization of the topics with terms.

#### **8.4.2 Data Preparation and Modeling**

In text mining, the basic unit for describing documents and corpora are terms. Basically, we can understand a term as a string derived from the words in a document. However, in most cases, we do not use the words as they appear in a text document but apply some transformations. These transformations are to some extent language specific, and we assume that the documents are written in English. The following transformations are offered by text mining software packages and are usually applied for the standardization of the documents.

##### **Transformations for Standardization of Documents**

- *Removal operations:* In general, numbers and punctuation do not contribute to the content of the text and are removed. Additionally, special characters like “/” or “@” are removed. In case of email documents, email address and signatures are usually removed.
- *White space and lowercase letters:* After the removal operations, it may be useful to remove extra white space produced by the removal operations. Furthermore, all words are transformed to lowercase letters.
- *Stop words:* Words that do not contribute to the intended meaning of the text are removed. Typical examples are articles, preposition, or auxiliary verbs.
- *Stemming:* words in a text are composed from a stem and affixes which mainly have syntactic meaning. Typical examples are endings like “ed” or “s.” The standard procedure is to remove these affixes.

*Example 8.5 (Example for Transformations)* For demonstration, let us consider Sect. 1.3.5 of this book. For the description of the document in the corpus, we use the following local metadata:

```
Metadata:  
author      : Rinderle-Ma, Grossmann  
datetimestamp: 2014-09-28 08:09:19  
description : brief task description  
heading     : 1.3.5 Evaluation and Reporting Task  
id          : 11  
language    : en  
origin      : Fundamentals of Business Intelligence  
              V1.0
```

The text described by the metadata reads as follows:

### 1.3.5 Evaluation and Reporting Task

The evaluation and reporting task looks at the analysis results from a global business perspective and positions the results of the analysis in the context of the business. Its main goals are the interpretation of the results in reference to domain knowledge and coming to a decision of how to proceed further. Usually, the evaluation task employs reporting techniques which are similar to data description and visualization techniques. Depending on the intended audience of the report, different types of reporting can be distinguished. We will sketch some ideas in Chapter 4.

Using the transformations described above, the text of the document is transformed into the subsequent:

```
[1] "evaluation reporting task"  
[2] "  
[3] "evaluation reporting task looks analysis results  
  global busi"  
[4] "perspective positions results analysis context  
  busi"  
[5] "main goals interpretation results reference  
  domain knowledg"  
[6] "coming decision proceed usually evaluation  
  task"  
[7] "employs reporting techniques similar data  
  description visu"  
[8] "techniques depending intended audience report  
  different types "  
[9] "reporting can distinguished will sketch ideas  
  chapter "
```

The stemmer used for stemming is a snowball version of the Porter stemmer<sup>7</sup> which is available in the text mining environment of R.

For further analysis, a number of decisions have to be made which may require further transformations. Let us mention the following ones:

1. *Consideration of synonyms:* Sometimes it may be useful to replace synonyms, i.e., words with the same or a similar meaning, by one term. For example, we could replace in context of BI the word “client” by the word “customer.”
2. *Consideration of homonyms:* By homonyms, we understand words with the same spelling and pronunciation but with different meaning. An example in BI applications may be the term “article” which may refer to something a customer orders or to a paragraph in a business contract.
3. *Definition of specific stop words:* Depending on the intended analysis, some words may not contribute to the content. Typical candidates may be adjectives, adverbs, or written numerals.
4. *Usage of abbreviations:* In some cases, it may be useful to replace two or more words occurring together by a commonly used abbreviation. In the context of the book, an example is the abbreviation “BPMN” for “business process model and notation.” On the contrary, it may be useful to expand abbreviations, for example, BI to “business intelligence.”

After preprocessing, the next step is *tokenization*, i.e., breaking the documents into a list of tokens. Generally speaking, tokens are defined by *n*-grams, i.e., a sequence of *n* contiguous words in a text. The most simple and frequently used cases are 1-grams, i.e., the tokens correspond to words in the text, or bigrams defined by two contiguous words. Bigrams are of interest in the case of composed terms like “business intelligence.” For special purposes it may be of interest to define tokens by mixtures of different *n*-grams, for example, words and bigrams. The different types of tokens are called *terms*.

Each term has a frequency of occurrence in a document which defines the *type-token relation*. In the case of tokens and terms defined by words, a model for the type-token relation is *Zipf's law* which states that the frequency of a term in a document is inverse proportional to its rank in the frequency table. If a document contains altogether *T* words of *V* different types and the rank of a type *w* in the list of types ordered by a decreasing frequency is  $\text{rank}(w) = k$ , then the probability of the rank is defined by a power relation, i.e.,

$$f(k) = \frac{1/k^\alpha}{\sum_{n=1}^V (1/n^\alpha)}, \quad k = 1, 2, \dots, V. \quad (8.2)$$

Here,  $\alpha$  is a document-specific parameter. A more theoretical exposition of this law can be found in [15]. In close relation to Zipf's law is Heaps' law, sometimes called

---

<sup>7</sup><http://snowball.tartarus.org/algorithms/porter/stemmer.html>.

Herdans' law. Heap's law defines a relation between the number of types  $V$  and the number of words  $T$  in a document by a power relation. A visual check for both laws can be done by double logarithmic plots. In the case of Zipf's law, the logarithms of the frequencies of the types are plotted against the logarithms of the ranks. In the case of Heaps' law, the logarithms of the number of types  $V$  is plotted against the logarithm of the text length  $T$ .

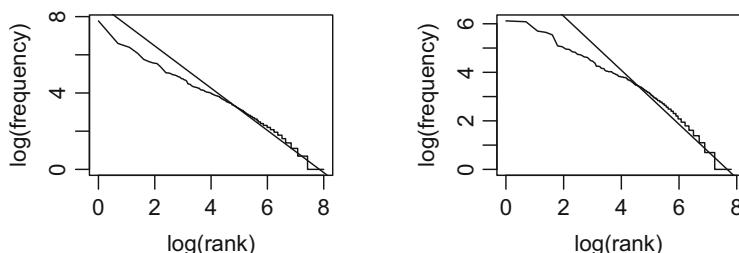
Given  $M$  documents  $D = \{d_1, d_2, \dots, d_M\}$  containing all together  $V$  different terms (words)  $W = \{w_1, w_2, \dots, w_V\}$ , we denote the term frequency of a term  $w_j$  in document  $d_i$  by  $t_{ij}$  and the  $M \times V$  document term matrix by

$$\text{DTM} = [t_{ij}], \quad i = 1, \dots, M, \quad j = 1, \dots, V \quad (8.3)$$

The rows of the DTM define the type-token relation for each document and the column sums of the DTM define the type-token relation for the entire corpus. Besides the DTM, also the transposed matrix called *term document matrix* (TDM) is used. The TDM is also known as *bag of words*.

*Example 8.6 (Document Term Matrix)* Let us demonstrate the concepts developed up to now using the corpus containing the 33 documents defined by the sections of Chaps. 1 and 2. After the removal of punctuation, numbers, and stemming of the words, we obtained by tokenization a DTM with 2,012 terms. The most frequent term occurring 2,381 times was "the," followed by "and" and "for" with frequencies 730 and 598. After removal of stop words, there were only 1,962 terms. The most frequent terms after removal were "model" and "busi" (which is the stem for "business") and "data" with frequencies 490, 475, and 439. The visual inspection of Zipf's law is shown in Fig. 8.14 for the corpus with and without removal of stop words. This figure shows that the removal of the stop words reduces the fit to Zipf's law.

The application of a bigram tokenizer resulted in a DTM with 11,377 terms, but 87 % of the bigrams occurred only once in the corpus. Not surprisingly, the most frequent bigrams were "business intelligence," "business process," and "key performance" (indicator).



**Fig. 8.14** Zipf's laws without removal of stop words (*left*) and with removal of stop words (*right*) for the example corpus (R package `tm`)

As the example shows, the DTM usually encompasses a large number of terms, and an important issue is the reduction to those terms that give important information about the documents. Such a reduction has to take into account two different reasons for frequent occurrence of terms. On the one hand, a term  $w_i$  occurs frequently in a document because it is important, i.e., gives more information about the content than a term occurring only once or twice. On the other hand, frequent terms occur only due to language usage and are not helpful for describing the contents of the documents. Typical examples are verbs or adverbs.

Depending on the analysis goal, different strategies can be applied for reducing the number of terms in the DTM. One basic strategy is defining a lower and an upper threshold for the term frequencies and remove all words which are not inside these thresholds. Another method is defining only a lower threshold and remove those frequent terms afterwards which are not important for the contents by enlarging the stop word list.

A more theoretically motivated method is using the *term frequency-inverse document frequency* (TF-IDF), in particular, if one is interested in finding terms which separate documents. Besides the term frequency  $t_{ij}$  of a term  $w_j$  in document  $d_i$ , i.e., the entries in the DTM, TF-IDF uses the document frequency  $DF_{ij}$  for the term defined by the number of documents which contain the term. Using these two quantities, TF-IDF is defined as

$$\text{TF-IDF}_{ij} = t_{ij} * \log(\text{IDF}_{ij}) = t_{ij} * \log(|D|/\text{DF}_{ij}). \quad (8.4)$$

Here,  $|D|$  denotes the number of documents. This basic definition is often modified. First of all, the definition of the term frequency may be modified. For example, one can standardize the term frequencies by the length of the documents in order to reduce the importance of large documents. Another frequently used modification is to take  $\log(|D|/(1 + \text{DF}_{ij}))$  which allows the application to terms which occur in none of the documents in the corpus.

TF-IDF can be understood as a weight for frequencies in the DTM which increases the importance of terms occurring in only few documents. A more theoretical analysis of TF-IDF and comparison with other weighting schemes may be found in [2]. For further aspects, we refer to Chapter 4 in [1].

The following example shows how one can apply TF-IDF for selection of terms using a summary statistic for the values of the TF-IDF for a corpus.

*Example 8.7 (TF-IDF for Document Term Matrices)* In the DTM for the 33 documents of Chaps. 1 and 2, there were altogether 1,962 terms after removal of stop words and stemming. An inspection of the term frequencies showed that in the list of terms with term frequencies above 20, there were altogether 186 terms. The list of words showed many words like “also,” “like,” “often,” or “will,” have no direct connection to the contents defined by BI. Hence, we computed the TF-IDF with term frequencies standardized by the size of the documents measured by the number of words in the documents. The mean was used as summary measure for all documents. Looking at the summary statistics of the average TF-IDF over the documents, we

selected terms with a value larger than the median. The reduced DTM contained only 1,007 terms. An inspection of the terms with a frequency above 10 showed that according to our understanding, all the 132 terms had a connection to the contents of the chapters and differentiated between the documents. On the other hand, the most frequent terms like “business,” “model,” or “data” were not in the list due to the occurrence in many documents.

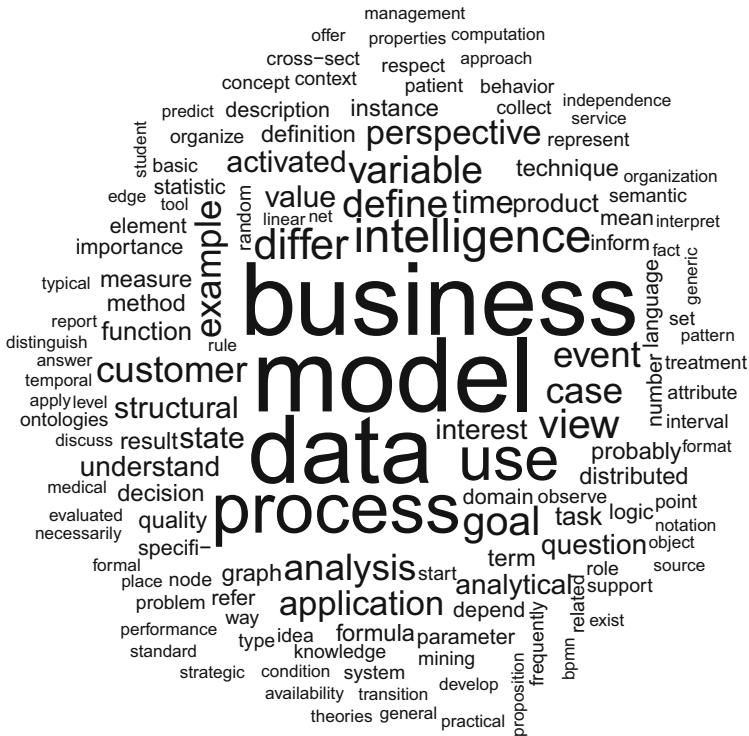
Besides the data preparation techniques treated in this section, there exist also a number of other techniques which modify the DTM. A frequently proposed method is dimensionality reduction similar to the idea of principal components introduced in Sect. 4.4.3. The matrix  $(DTM) \cdot (DTM)^T$  is factorized according to its eigenvalues and only those dimensions of the matrix are retained which correspond to the largest eigenvalues. We will not pursue this approach and refer the interested reader to monographs on text mining, for example, [5].

### 8.4.3 Descriptive Analysis for the Document Term Matrix

Based on a document term matrix, different kinds of descriptive analysis can be done. The most popular is probably the visualization of the terms using a word cloud. A word cloud represents terms in the DTM with size according to the frequency in the corpus or in a document. The word cloud allows quick perception of the most frequent terms in the corpus. Besides size, colors can also be used for representing the terms. The order of the terms can be done according to the frequency with most frequent words in the center. Another option is to use a random position for the words. A further parameter allows the representation of a proportion of terms rotated by  $90^\circ$ . With respect to the overall layout, rectangular or circled shapes can be used. Usually one includes only terms with frequency above a certain threshold which is an additional parameter in the design of the word cloud. In the case of DTMs produced with stem terms, expansion of the stem terms to words improves readability.

*Example 8.8 (Word Cloud for a Document Term Matrix)* Taking the basic DTM with 1,962 terms from the 33 documents as described in Example 8.6, we decided to reduce the DTM to words above a frequency threshold of 20. The stem terms were expanded to the words with the highest frequency. However, some manual correction was necessary after applying the expansion function of R. Furthermore, a number of words had a frequency above the threshold 20 and were removed as additional stop words. Overall, 138 terms remained for the cloud. The cloud in Fig. 8.15 shows all the terms in a circled layout and with fixed order. The most frequent terms are shown in the center of the cloud.

For a comparison of the frequencies of the terms in the DTM over the documents a *comparison cloud* can be produced by the R package *wordcloud* ([28]). If the DTM contains  $M$  documents, the deviation of the relative frequencies of a term  $w_i$



**Fig. 8.15** Word cloud for frequent terms in Chaps. 1 and 2 (R package wordcloud)

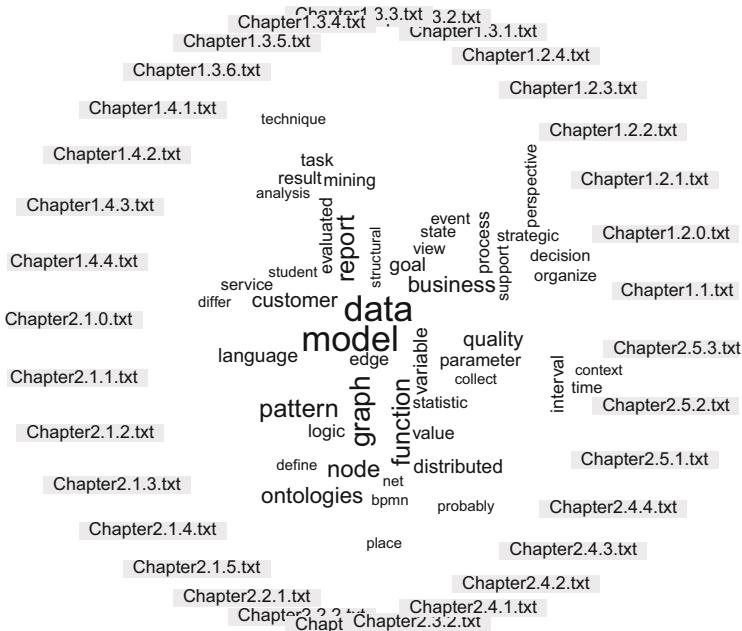
in a document  $d_j$  from the mean relative frequency of the document is measured by

$$d_{ij} = (p_{ij} - p_j), \quad p_{ij} = t_{ij}/\sum_j t_{ij}, \quad p_j = \sum_i p_{ij}/M. \quad (8.5)$$

The comparison cloud shows the different documents in an outer circle. The angular position of the terms is defined by the document in the outer circle for which  $d_{ij}$  attains its maximum. The size of the term is chosen according to the value of the maximum deviation. Additionally, one can use colors for identification of the documents.

*Example 8.9 (Comparison Cloud for a Document Term Matrix)* Figure 8.16 shows the comparison cloud for the 33 documents defined by the sections of Chaps. 1 and 2. As one can see, there is a good coincidence with the contents of the sections and the position of the terms.

Besides word clouds, another descriptive technique for a DTM is the calculation of the association between the terms in the different documents. Association can be made precise in different ways, for example, as a correlation between the term



**Fig. 8.16** Comparison cloud for frequent terms in Chaps. 1 and 2 (R package `wordcloud`)

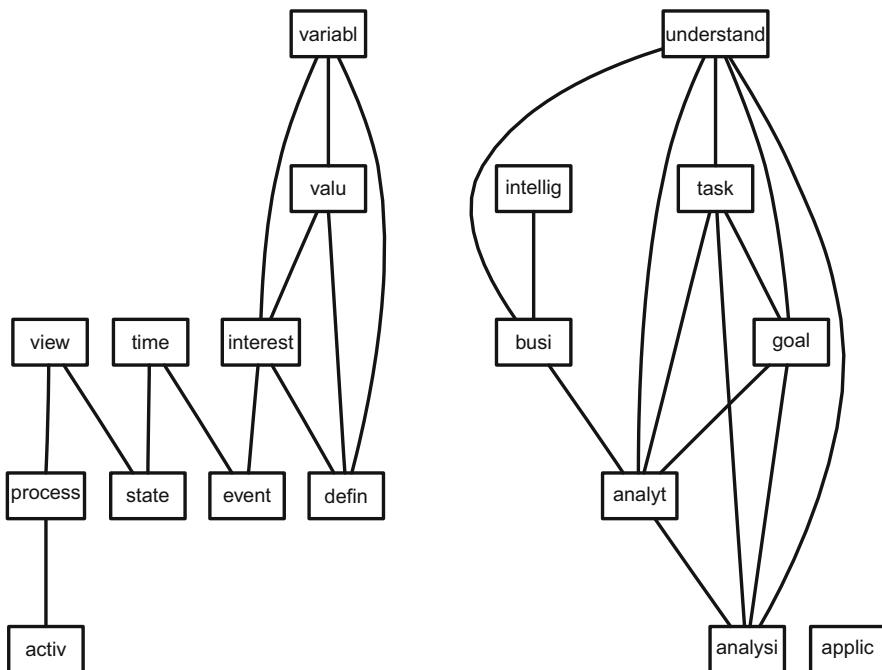
frequency vectors of the different documents or as an association in the sense of association analysis in Chap. 6. In the following example, we will show an application of correlation analysis.

*Example 8.10 (Association Analysis for a Document Term Matrix)* Starting with the DTM of Example 8.6 we looked for terms for which the correlation with the term “business” is larger than 0.6. For example, the correlation between “business” and “intelligence” is 0.76 and between “business” and “understanding” 0.72, respectively.

Next, we restricted the DTM to the 32 terms with a frequency above the threshold 70. For these terms, the correlation matrix was calculated and a threshold for a meaningful correlation between two terms was defined by 0.6. Figure 8.17 shows the relationships between the terms with correlations above the threshold. Here, we used only the stems of the words.

#### 8.4.4 Analysis Techniques for a Corpus

Based on the example for a corpus defined by the sections of Chaps. 1 and 2 of the book, we show in this section how the methods for clustering introduced in Chap. 5 can be applied for a corpus. For understanding the structure behind the documents



**Fig. 8.17** Associations between frequent terms in Chaps. 1 and 2 (R package `tm`)

in a corpus, we introduce topic models. Finally, we discuss some peculiarities in connection with the classification of documents.

### Cluster Analysis for Text Data

Cluster analysis for text documents is one of the most frequently used analysis techniques in text mining. Frequently, clustering techniques are used based on the distances between documents. An important decision for the analyst is what kind of preprocessing should be done for the DTM representing the corpus. As discussed in Sect. 8.4.2, it may be useful to use a selection based on TF-IDF instead of the entire DTM.

For finding the similarity or distance between the documents in a corpus, the cosine function is the first choice. If  $d_i$  and  $d_j$  are two documents in the corpus with term frequency vectors  $\mathbf{v}_i$  and  $\mathbf{v}_j$  defined by the rows of the DTM, the similarity is defined by

$$\text{sim}(d_i, d_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j'}{\|\mathbf{v}_i\| \cdot \|\mathbf{v}_j\|}. \quad (8.6)$$

As usual “ $\cdot$ ” defines the inner product of vectors and  $\|\cdot\|$  denotes the norm of the vector.

After the definition of the distance, one can use hierarchical cluster methods or  $k$ -means clustering. Sometimes, partitioning around medoids (PAM, Sect. 5.4.3) is proposed. A nice property of PAM is that the center of the clusters are elements of the data. There are two main disadvantages of PAM. The first one is that the computation is rather slow in the case of large data, and the second one is that the representation of the clusters by the medoid may miss important terms in the case of a sparse DTM.

*Example 8.11 (Clustering a DTM)* Using the same DTM as in the case of demonstration of the word cloud in Example 8.8, we used hierarchical clustering with the Ward method. The solutions at different levels can be explained quite well. In the case of two clusters, which show a high level of separation, one cluster with 14 documents can be interpreted as a cluster with documents treating general aspects of modeling, logical and algebraic structures, and analytical structures. The second cluster with 19 elements mainly contains the documents from Chap. 1 dealing with data and general aspects of BI. A grouping into three clusters splits the analytical and modeling cluster into one cluster containing documents about general aspects of modeling and one cluster with documents about analysis techniques. Further splits from the cluster containing the 19 documents about data and general aspects of modeling separate two small clusters at almost the same level. The two new clusters can be labeled by the terms “data cluster” (the three documents of Sects. 1.3.1, 2.5.1, and 2.5.3) and “general considerations cluster” (Sects. 1.1 and 1.2.2 and the introduction of Chap. 2).

Besides these basic approaches, many other techniques have been proposed. For example, instead of the cosine function, one can use string kernels for definition of the distances (cf. [12] for an application using R). Another technique for clustering documents is *co-clustering*, which aims at simultaneous clustering of the documents and the terms. Clustering of the documents and the terms is done in such a way that there is an optimal fit between the two cluster solutions. One method for the definition of this fit is the interpretation of the DTM as a bipartite graph, where the documents and the terms define the two sets of nodes and the entries in the DTM define the edges. The algorithm introduced in [10] defines a partition in the two node sets which minimizes the edges between different partitions in the two clusters.

## Topic Models

The idea behind topic maps is similar to model-based clustering in Sect. 5.4.4. We assume that a document  $d_i$  in the corpus  $D = \{d_1, d_2, \dots, d_M\}$  is composed from different topics  $t_1, t_2, \dots, t_K$  with probabilities for topic  $t_k$  given by  $p(t_k|d_i)$ . For each topic  $t_j$ , the occurrence of a word  $w_\ell$  from a set of words  $W = \{w_1, w_2, \dots, w_N\}$  follows a topic-specific distribution defining the probability  $p(w_\ell|t_k)$ . Using Bayes’ theorem, we can represent the probability of the word in the document by

$$p(w_\ell|d_i) = \sum_{k=1}^K p(w_\ell|t_k) p(t_k|d_i). \quad (8.7)$$

Based on this model, one can understand the occurrence of words in a document as the result of the following process (cf. [6]):

### Data Generation in Topic Models

1. Choose randomly for a document a distribution over the topics.
2. For each word in the document.
  - a. Choose randomly a topic according to the distribution of the topics in the document.
  - b. Choose randomly a word according to the distribution of the words in the topic.

The estimation task is now to find for a given DTM the number of topics, the distribution of the topics for each document, and the distribution of words within the topics.

Given the number of topics, two solutions are frequently used. The first one is known as *probabilistic semantic index (PLSI)*. This method uses the model specification defined in Eq. (8.7) and estimates the topic probabilities and the word probabilities by maximum likelihood estimation in an iterative way. An outline of the algorithm can be found in [1, Chapter 4].

The second method nowadays most frequently used is *latent Dirichlet allocation (LDA)* which reformulates the problem in a Bayesian framework. The priors for the topics and the words are defined by Dirichlet distributions (cf. Sect. 6.4.1 for the definition of the Dirichlet distribution). The estimation of the parameters is done by a special version of the EM algorithm (cf., Sect. 5.4.4). The results of this algorithm are posterior distributions for the topics in each document and for the terms within the topics. A detailed description of the algorithmic solution can be found, for example, in [14]. For the estimation of the number of topics, one can use the method of cross-validation. An alternative is proposed in [35] and implemented in the R package `maptpx`.

*Example 8.12 (Topic Models for the Sample Corpus)* Using the DTM of the corpus of the sections of Chaps. 1 and 2, we first reduced the DTM according to TF-IDF as described in Example 8.7 and fitted topic models from two to five. The model with two topics corresponds, not surprisingly, to Chaps. 1 and 2. A model with five topics showed that for most of the sections, there is one dominant topic. For example, topic 4 dominates in Sect. 1.2.4 (Goals of BI), 1.3.2 (Business and Data Understanding Task), 2.3.1 and 2.3.2 (Graph Structures), and 2.4.4 (Modeling Methods Using Analytical Structures). Correspondingly the five most important terms for this topic are “graph,” “goal,” “node,” “variable,” and “transition.” An example for a document which is a mixture of two topics is Sect. 2.1.3 about model building, which is mainly a mixture of topic 2 (0.634) and topic 3(0.364). A detailed description of the results can be found on the homepage of the book.

## Classification of Documents

A straightforward solution for the classification of documents is the application of the algorithms of Chap. 5 to the DTM. Practically, all different methods for classification have been used in document classification. For an overview, see [1], for example.

However, many times the DTM of the training data is a sparse matrix, and it is advisable to reduce the number of terms. Besides the general methods for term reduction discussed in Sect. 8.4.2, techniques that use the information about the class membership in the training data have been proposed. The basic idea is to find terms with high discriminatory power and apply the classification algorithms to the modified DTM. A well-known example is spam filtering for emails. In this case, it is useful to focus on terms which occur frequently in spam mails, for example, emails promising high winnings. Identification of such terms can be based on different methods. For example, one can use measures of the information gain for the selection of terms which have high discriminatory power in the training data. This method is similar to the selection criteria for variables in decision trees (cf. Sect. 5.3.3) and is described in detail in [1, Chapter 6]. Other methods for term selection are frequently specific to the problem. Methods for the selection of terms in connection with opinion mining will be discussed in Sect. 8.4.5.

As soon as the terms of interest have been identified, one can use either a DTM with this terms or the incidence matrix corresponding to the DTM, i.e., we consider only the occurrence of the term.

### 8.4.5 Further Aspects of Text Mining

Up to now, we have considered analytical goals in text mining at the level of a corpus based on the DTM. Besides these goals, a number of other goals can be formulated which are not confined to a corpus of documents but refer to different types of text data like words, sentences, single documents, or linked documents (cf. [25] for a systematic overview of text mining goals). Achieving these goals requires additional analytical techniques; however, for evaluation purposes, the reference to a corpus is often necessary. In the following, we will briefly discuss some frequently used techniques and then show how different techniques can be applied in opinion mining.

#### Analysis at the Word Level

Natural languages have a high expressive power and allow the denotation of similar concepts with different words, called *synonyms*. Moreover, nouns representing a concept are usually embedded into a hierarchy of other concepts. Terms representing a broader concept are called *hyponyms*, and terms representing a narrower concept are called *hyponyms*. Other possible relations between concepts is the “part of” relation. In the case of verbs, it is also possible to express a special way of doing an activity (*troponyms*). In the case of adjectives, one can define *antonyms*, which are

adjectives with opposite meaning. Additionally, one can define relations between different types of words (e.g., nouns and verbs).

For usage of such information in text mining, a database is needed which keeps lexical information and semantic information. For English words, WordNet<sup>8</sup> offers such a database which is freely available and can be downloaded from the website. R offers access to this database in the library `wordnet`. Similar projects exist for other languages but are in general not free of charge.

*Example 8.13 (Relation Between Terms Using WordNet)* Looking for synonyms in WordNet for the word “business” provides altogether 16 words which are grouped into 9 senses. The most frequent sense is given by the synonyms “concern,” “business concern,” “business organization,” and “business organisation.” Not so frequently used are senses of business with synonyms “occupation,” “job,” “line of work,” or “line.” As senses without synonyms, the interpretation of business as a volume of commercial activity (“business is good today”) or concern (“mind your own business”) is listed. The different senses together with the synonyms are called the *synset* of the word.

A hierarchy of hypernyms for business in the first sense is defined by the terms “enterprise,” “organization,” “social group,” “group,” “abstraction,” “abstract entity,” and “entity.” Hyponyms are according to WordNet terms like “agency,” “brokerage,” “carrier,” “chain,” “firm,” “franchise,” and others. For the sense of the term “business” as a “commercial enterprise,” a part of relation is defined between the terms “business” and “market place.”

If one looks for the synset of the word “model,” one can find sysnsets for the noun “model,” as well as the verb “model,.” In both cases, different senses are possible. In this case, the relation between the noun and the verb is also indicated.

Looking for the term “busy,” the word is identified either as verb or as adjective. In case of the adjective, the most important sense is related to active. Correspondingly, the antonym is given by the word “idle.”

### Analysis at the Sentence Level

Text mining at the sentence level is mainly concerned with understanding the structure of a sentence, i.e., identification of the position and the role of the words in a sentence. This identification is known as *part-of-speech tagging (POS tagging)*. Part-of-speech tagging identifies the role of the words in a sentence using tags as standard descriptors. A frequently used set of tags are the Penn Treebank tags.<sup>9</sup> Finding tags in a sentence has to resolve disambiguation, and different machine learning algorithms have been proposed. There are many open-source taggers for different languages available; for example, the text mining package of R offers a tagger of the Apache OpenNLP library.<sup>10</sup>

---

<sup>8</sup><http://wordnet.princeton.edu/>.

<sup>9</sup><http://www.cis.upenn.edu/~treebank/>.

<sup>10</sup><https://opennlp.apache.org/>.

*Example 8.14 (Part-of-Speech Tagging)* Applying part-of-speech tagging to the sentence:

*The evaluation and reporting task looks at the  
analysis results from a global business perspective.*

leads to the following structure

```
{ (TOP
  (S
    (NP (DT The) (NN evaluation) (CC and)
          (NN reporting) (NN task) )
    (VP (VBZ looks)
    (PP (IN at) (NP (DT the) (NN analysis)
          (NNS results)))
    (PP (IN from) (NP (DT a) (JJ global)
          (NN business) (NN perspective)))) }
```

The sentence starts with the tag S and contains a noun phrase NP, a verb phrase VP, and two prepositional phrases PP. Within the phrases, the words are identified by tags. For example, NN stands for noun singular and NNS for nouns in plural, VBZ is a verb, 3rd person singular present, IN indicates prepositions, DT determiners, JJ adjectives, and CC coordinating conjunction.

### Keyword Extraction

The extraction of keywords is of utmost importance in indexing Web documents, and many different techniques have been proposed. Most analytical techniques for the automatic extraction of keywords from documents combine text mining methods described in this section. The precise formulation of the analytical task depends on the available information. We will outline here an approach which is used, for example, in the open-source tools *keyword extraction algorithm* (KEA)<sup>11</sup> and the further development *multipurpose automatic topic indexing* (MAUI).<sup>12</sup> Another tool using his approach is RAKE which is described in detail in [5].

Usually, the first step in keyword extraction is the selection of candidate terms based on statistical measures for the terms in the document. These candidates are found by tokenization using  $n$ -grams, in most cases up to three words, and the most frequent terms are used. If there is a thesaurus of controlled keywords available, these candidates can be matched against the vocabulary in the thesaurus. Based on these candidates, a number of features for the document can be calculated. If one is interested in finding keywords for documents in a corpus, a basic feature is the TF-IDF (cf. Sect. 8.4.2). Other features of interest can be based on part-of-speech tagging (POS) described above, on the occurrence of the word in the text (usually a word at the beginning of the text is more likely to be a keyword than a word at the end of a text), or on the semantic relatedness of a term to words in a thesaurus. Using

---

<sup>11</sup><http://www.nzdl.org/Kea/>.

<sup>12</sup><https://code.google.com/p/maui-indexer/>.

these features, a scoring of the candidates for the keywords is done. In the case of training data these scores can be calculated using a supervised learning approach. If no training data are available, heuristic methods can be applied. Afterwards, the keywords are defined by the terms with the highest scores.

### Opinion Mining and Sentiment Analysis

Understanding how customers perceive products or services is a core topic in marketing. The traditional methods used for learning about the opinion of customers is market and opinion research. However, nowadays, one can find customer opinions about products and services at e-business portals, opinions about politicians and political decisions in blogs, tweets and fora, or customer reviews of movies or theater performances at specific portals. Hence, it is not surprising that there is great interest in understanding and interpreting these customer opinions on the Internet using text mining methods.

One can find a lot of research activities and applications in this area under the headings *opinion mining*, *sentiment analysis*, or *subjectivity analysis*. According to [29], subjectivity analysis is the oldest and broadest term subsuming the other two. Subjectivity is defined in this paper as *linguistic expressions of a persons' private states like opinions, evaluations, or emotions which cannot be completely verified by objective criteria*. This lack of objectivity discerns opinionated information from factual information. In combination with the linguistic diversity in the expression of sentiments and opinions, the analysis of subjectivity in text documents creates a number of challenging problems for text mining.

Consequently, many different approaches have been proposed and a rich set of techniques is available based on data mining methods described in the previous sections, methods from artificial intelligence, computational linguistics, or psychology. For an excellent summary of the problems, together with an extensive bibliography we refer to [29]. In the following we will describe some basic tasks shown in the overview box. These tasks are elaborated in more detail in [21] and start with a problem formulation for sentiment analysis which resembles some concepts of market and opinion research and points to specificities of opinion mining.

#### Tasks in Opinion Mining

- Finding in a document all opinionated sentences
- Identification of the objects and the features about which an opinion is expressed
- Classification of the expressed opinion

The basic unit for opinion mining is a document in which an opinion holder expresses her/his opinion about certain features of an object of interest. This opinion is given at a certain time and has an orientation, in the simplest case positive, negative, or neutral. Note the similarity of the set-up with a traditional survey: The document plays a role similar to a questionnaire, the opinion holder corresponds to a surveyed person, the features correlate to the questions about the product, and the

orientation of the opinion replaces the answers to the questions. However, making this similarity operational for analysis by machines causes a number of challenging problems.

### *Finding Opinionated Sentences*

Opinionated sentences occur in two guises: either a sentence expresses an opinion explicitly, for example, a sentence like “The performance of Sandra Bullock in this movie was exceptional” or in comparative form, for example, a sentence of the form “In this movie the performance of Sandra Bullock was much better than in her previous film.”

As the first example shows, explicit opinions are usually expressed by adjectives, e.g., “good,” “beautiful,” “poor,” or “awful.” Another way for expressing opinions is using certain verbs as “like” or “hate,” adverbs and nouns, for example, “remarkable” and “junk,” or opinion phrases like “deserves attention.” Opinions expressed in a comparative way many a time use adjectives in comparative or superlative form. Using part-of-speech tagging allows the identification of such phrases.

### *Identification of Objects and Features*

Frequently, identification of the objects and the features about which the opinion is expressed in a document starts with the assumption that there is only one object of interest in the entire document. Such an assumption is justified in the case of review documents about products or documents expressing the opinion about persons. The identification of the object is often done according to the context of the creation of a document, e.g., a product review on a business portal. Other methods for the identification of the objects are using keyword matching of documents using the name of a person or the name of a brand.

The identification of the features of interest is more complicated because one has to take into account the linguistic diversity for naming the features. For example, for the evaluation of a movie, one can use features such as script, director, music, and the performance of the actors. For each feature, different terms can be used. Take as an example the performance of an actress which can be described using phrases like “she acts,” “she plays,” or “she represents.” For the resolution of this diversity, one can use semantically oriented databases such as WordNet.

### *Opinion Classification*

The opinion expressed in a sentence is called the *polarity* of the sentence. Finding the polarity of a short sentence with only one object of interest can be rather simple by using the polarity of words in the sentence. This polarity of words can be accessed by comparison with a dictionary with polarity tags. Such dictionaries exists, for example, SentiWordNet which is an extension of WordNet including the polarity of words (cf. [3]).

Using a list of positive and negative terms for the English language,<sup>13</sup> one can assign polarity in the case of direct expression of the polarity simply by matching the words with such a list. Such an application has got rather popular in the last years for analyzing tweets. A tutorial for explaining this approach using R for the evaluation of airlines can be found in [7].

The evaluation of complex sentences needs a more complex representation of words and sentences. On the one hand, the syntax of the sentences has to be analyzed, for example, negation has to be identified. Another aspect is finding the overall opinion in the case of differing opinions about different features of the object of interest. The opinions about features can be summarized in scores or can be represented individually using techniques like radar plots introduced in Sect. 4.5.1. Another aspect which hampers opinion classification is the identification of ironic statements.

An interesting project in this area is *SenticNet* described in detail in [8]. The approach was developed in the spirit of artificial intelligence and aims at modeling common sense in sentences by looking more explicitly at psychological aspects in expressions and the recognition of emotions. To achieve this goal, a *bag of concepts* model is used instead of the bag of words model described in Sect. 8.4.2. Using this bag of concepts, a number of aspects are distilled which correspond to the features of the object which is opinionated by the sentence. These aspects are evaluated with respect to the polarity in the sentence and mapped into an affective space, and an overall opinion score is calculated. A prerequisite for application of SenticNet is the assumption that the sentence is an opinionated sentence and not a factual statement.

*Example 8.15 (SenticNet Example)* Suppose in a movie review occurs the following summary:

*The script is stupid but the performance of the actors is excellent*

Using SenticNet,<sup>14</sup> the following concepts are derived:

“script,” “stupid\_script,” “performance,” “performance\_of\_actor,” “actor,” and “excellent\_performance.”

The aspects extracted are:

“performance,” “actors,” and “script.”

The overall sentence polarity is evaluated with *0.5/null*.

If one replaces the word “stupid” by “illogical,” the polarity of the sentence

*The script is illogical but the performance of the actors is excellent*  
changes to *positive*.

As in other applications of text mining, the development of knowledge which allows the classification of opinions is based on the analysis of existing text corpora. Similar to the case of keyword extraction, the first step in the analysis is the definition of features of the document which are used afterwards in techniques for

---

<sup>13</sup>A list of positive and negative words can be found in <http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon>.

<sup>14</sup><http://sentic.net/demo/>.

supervised and unsupervised learning for text documents. The tuning of the methods often depends on the application domain. Details can be found in [22].

In the evaluation of results of opinion mining, two effects have to be taken into account. The first one is spam opinion, i.e., persons post a faked opinion about a product. For example, on tourist portals, an accommodation may post faked positive evaluations or on a platform about politicians, negative statements may be posted about a candidate. Identifying such deceptive opinions has become a special kind of text mining in the last years. In [20, 27] one can find applications of supervised learning for the identification of a deceptive mail. A second problem is a phenomenon similar to *publication bias* in *meta-analysis*. Meta-analysis aims at an improvement of research results by combining the findings of different publications, particularly in medical research. Publication bias occurs due to the fact that many a time insignificant research results are not published. A similar effect can occur in published opinions because we do not know what motivates a person to post an opinion on the Internet. For example, persons satisfied with a service will not post their opinion, whereas people with negative experience may be interested to post their opinion.

#### 8.4.6 Summary: Text Mining

Text mining combines ideas from different scientific disciplines and has got more and more importance in the last 15 years. This section focused on a text mining approach based on the document term matrix for a corpus. The definition of the document term matrix needs a number of data preparation steps like removal of special characters, removal of stop words, or stemming. Next, one has to decide which method is used for tokenization and how the different terms are weighted. An important method for weighting is the calculation of the so-called *term frequency-inverse document frequency* matrix.

Descriptive analysis of the document term matrix allows the representation of the documents as word clouds and the comparison of documents can be done by a comparison cloud. For understanding the structure of a corpus, one can apply various cluster algorithms. A more theoretically oriented method for analyzing the corpus structure are topic models.

Finally, the basics of text mining at the word level and at the sentence level were defined, and the application of mixed methods for keyword extraction and for opinion mining and sentiment analysis was demonstrated.

## 8.5 Conclusion and Lessons Learned

In the following, we comment on existing methods to evaluate and assess the analysis methods presented in this chapter.

For social network analysis, the basic question is what should the results be compared to in order to evaluate them. If the task is to classify in a social network the type of the relation between actors using a number of attributes, the quality of the classification can be measured using metrics such as precision and recall (an introduction to these metrics can be found in Sect. 5.3.1). This was applied in a study for extracting the relations between researchers from Web data [24]. This paper also investigates questions of finding networks with similar affiliation of researchers and compares the results with networks obtained from questionnaires.

Mining organizational models from an event log is often done in the presence of a so-called *prescriptive* organizational model. Prescriptive organizational models set out how the organizational structure of the company looks like and how the access of actors to process activities is managed. However, as mentioned in several studies and works, e.g., [18, 19, 23, 26], actors might deviate from the prescriptive model due to various reasons such as delegation or unavailability. Thus, the results from organizational mining will deviate from the prescriptive models. Of course, one could apply metrics, such as precision and recall, to compare the results of organizational mining and prescriptive models. However, the conclusion would not be very meaningful. In this case, it is more interesting to find differences in order to pinpoint and analyze deviations. This can be achieved by applying *delta analysis* between the prescriptive and actual organizational model [18].

To the best of our knowledge, there is no evaluation or assessment method that has been proposed for decision point analysis (DPA). DPA combines two analysis methods, i.e., process mining and cross-sectional analysis (more precisely, decision trees). Evaluation techniques exist for both methods. Discovered process models can be evaluated based on conformance checking (see Sect. 7.6) and related quality metrics fitness, precision, generalization, and structure.

In the case of text mining, we have shown how one can apply the methods of supervised an unsupervised learning to text data. The main challenge in text mining is data preparation and the definition of a model which transforms unstructured text into a structured model. The focus in this section was on models based on the document term matrix DTM. The DTM is basic for analyses of a corpus of text documents. Depending on the analytical goal, a number of additional features of texts have to be considered. These features are often derived by methods of computational linguistics. An important resource for many text mining applications are databases, such as WordNet, and algorithms for natural language processing such as part-of-speech (POS) tagging. These tools are of utmost importance in the case of opinion mining and sentiment analysis.

With respect to the evaluation of text mining results, the evaluation criteria are basically those defined in Chap. 5 for supervised and unsupervised learning. In the case of opinion mining, an additional benchmark is defined by the human perception of expressed opinions. This perception is usually not as unique as for factual information (cf. [29]). An additional problem in opinion mining is the relevance and the validity of the results. Even if the problems of analysis of opinion by machines are solved successfully, the interpretation of the results may be rather tricky. Contrary to traditional market research, secondary data are frequently used

and it is by no means evident that the data correctly represent the target group. Finally, the effects of spam opinion and the bias of published opinions have to be taken into account.

## 8.6 Recommended Reading

The reference work for organizational mining is Song (2008). For text mining, we recommend from a more practical point of view Berry (2010). A more theoretical exposition of text mining can be found in Aggarwal (2012). The topic of opinion mining is treated from a more practical point of view in Pang (2008) and Liu (2010).

- Aggarwal CC, Zhai C (2012) Mining text data, Springer
- Berry MW, Kogan J (2010) Text mining: applications and theory. Wiley Online Library
- Liu B, Zhang L (2012) A survey of opinion mining and sentiment analysis. In: Aggarwal CC, Zhai C (eds) Mining Text Data, Springer, pp. 415–463
- Pang B, Lee L (2008) Opinion mining and sentiment analysis. Foundations and trends in information retrieval 2(1–2):1–135
- Song M, van der Aalst WMP (2008) Towards comprehensive support for organizational mining. Decision Support Systems 46(1):300–317

## References

1. Aggarwal CC, Zhai C (2012) Mining text data. Springer, New York
2. Aizawa A (2003) An information-theoretic perspective of tfidf measures. Inf Process Manag 39:45–65
3. Baccianella S, Esuli A, Sebastiani F (2010) SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. LREC 10:2200–2204
4. Batagelj V, Mrvar A (2004) Pajek—analysis and visualization of large networks. Springer, New York
5. Berry MW, Kogan J (2010) Text mining: applications and theory. Wiley Online Library, Chichester
6. Blei DM (2012) Introduction to probabilistic topic models. Commun ACM 55(4):77–84
7. Breen J (2011) R by example: mining Twitter for consumer attitudes towards airlines. <https://jeffreybreen.wordpress.com/2011/07/04/twitter-text-mining-r-slides/>
8. Cambria E, Hussein A (2012) Sentic computing. Springer, Dordrecht
9. de Leoni M, Dumas M, García-Bañuelos L (2013) Discovering branching conditions from business process execution logs. In: Cortellessa V, Varró, D (eds) FASE’13: 16th int’l conference on fundamental approaches to software engineering. Lecture notes in computer science, vol 7793. Springer, Heidelberg, pp 114–129
10. Dhillon IS (2001) Co-clustering documents and words using bipartite spectral graph partitioning. In: Lee D, Schkolnick M, Provost FJ, Srikant R (eds) ACM SIGKDD’01: International conference on knowledge discovery and data mining. ACM, New York, pp 269–274

11. Dunkl R, Rinderle-Ma S, Grossmann W, Fröschl KA (2014) Decision point analysis of time series data in process-aware information systems. In Nurcan S, Pimenidis E, Pastor O, Vassiliou Y (eds) CaISE forum: joint proceedings of the CAiSE 2014 forum and CAiSE 2014 doctoral consortium, CEUR workshop proceedings 1164, CEUR-WS.org, pp 33–40
12. Feinerer I, Hornik K, Meyer D (2008) Text mining infrastructure in R. *J Stat Softw* 25(5):1–54
13. Fruchterman TMJ, Reingold EM (1991) Graph drawing by force-directed placement. *Software* 21(11):1129–1164
14. Grün B, Hornik K (2011) topicmodels: an R package for fitting topic models. *J Stat Softw* 40(13):1–30
15. Johnson NL, Kotz S, Kemp AW (1992) Univariate discrete distributions, 2nd edn. Wiley, New York
16. Jungnickel D (1994) Graphen, Netzwerke und Algorithmen, 3rd edn. BI-Wissenschaftsverlag (in German)
17. Kamada T, Kawai S (1989) An algorithm for drawing general undirected graphs. *Inf Process Lett* 31(1):7–15
18. Leitner M, Rinderle-Ma S (2014) Anomaly detection and visualization in generative RBAC models. In: Osborn SL, Tripunitara MV, Molloy I (eds) SACMAT'14: ACM symposium on access control models and technologies. ACM, New York, pp 41–52
19. Leitner M, Baumgrass A, Schefer-Wenzl S, Rinderle-Ma S, Strembeck M (2013) A case study on the suitability of process mining to produce current-state RBAC models. In: La Rosa M, Soffer P (eds) Business process management workshops. Lecture notes in business information processing, vol 132. Springer, Heidelberg, pp 719–724
20. Li J, Ott M, Cardie C, Hovy E (2014) Towards a general rule for identifying deceptive opinion spam. In: ACL'14: annual meeting of the association for computational linguistics. The Association for Computer Linguistics, Stroudsburg, Pa, pp 1566–1576
21. Liu B (2010) Sentiment analysis and subjectivity. In: Indurkhya N, Damerau FJ (eds) Handbook of natural language processing. Chapman&Hall/CRC, Boca Raton, pp 627–666
22. Liu B (2012) Sentiment analysis and opinion mining (synthesis lectures on human language technologies). Morgan & Claypool Publishers, San Rafael
23. Ly LT, Rinderle S, Dadam P, Reichert M (2005) Mining staff assignment rules from event-based data. In: Bussler C, Haller A (eds) Business process management workshops. Lecture notes in computer science, vol 3812. Springer, Heidelberg, pp 177–190
24. Matsuo Y, Mori J, Hamasaki M, Nishimura T, Takeda H, Hasida K, Ishizuka M (2007) POLYPHONET: an advanced social network extraction system from the web. *Web Semantics* 5(4):262–278
25. Miner G, Delen D, Elder J, Fast A, Hill T, and Nisbet R (2012) Practical text mining and statistical analysis for non-structured text data applications. Elsevier, Waltham, Ma
26. Molloy I, Park Y, Chari S (2012) Generative models for access control policies: applications to role mining over logs with attribution. In: Atluri V, Vaidya J, Kern A, Kantarcioğlu M (eds) SACMAT'12: ACM symposium on access control models and technologies. ACM, New York, pp 45–56
27. Ott M, Cardie C, Hancock JT (2013) Negative deceptive oponion spam proceedings of NAACL-HLT 2013, pp 497–501
28. Package ‘wordcloud’ <http://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf>. Accessed 12 December 2014
29. Pang B, Lee L (2008) Opinion mining and sentiment analysis. *Found Trends Inf Retr* 2(1–2):1–135
30. Rinderle S, Reichert M (2007) A formal framework for adaptive access control models. *J Data Semant IX*:82–112
31. Rozinat A, van der Aalst WMP (2006) Decision mining in ProM. In: Dustdar S, Fiadeiro JL, Sheth AP (eds) BPM'06: International conference on business process management. Lecture notes in computer science, vol 4102. Springer, Heidelberg, pp 420–425
32. Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. *Computer* 29(2):38–47

33. Scott J (2013) Social network analysis. SAGE, London
34. Song M, van der Aalst WMP (2008) Towards comprehensive support for organizational mining. *Decis Support Syst* 46(1):300–317
35. Taddy MA (2012) On estimation and selection of topic models (2012). In: Proceedings of 15th international conference on artificial intelligence and statistics
36. Wainer J, Barthelmess P, Kumar A (2003) W-RBAC—a workflow security model incorporating controlled overriding of constraints. *Int J Coop Inf Syst* 12(4):455–485
37. Wasserman S (1004) Social network analysis: methods and applications, vol 8. Cambridge University Press, Cambridge
38. Wellman B (1004) Are personal communities local? A Dumptarian reconsideration. *Soc Netw* 18(4):347–354

# Chapter 9

## Summary

Summarizing the contents of a book is a matter of personal preferences. One way to obtain more objectivity is to use formal criteria for the identification of the most important findings. In Chap. 8, we introduced text mining, in particular, analysis methods based on the term *document matrix* for the detection of structure in text data. Hence, we thought that it is self-evident to use the text mining approach as a starting point for a summary. Specifically, we used the following procedure in order to acquire an overview on the contents in the different chapters:

- Definition of a corpus containing the eight chapters of the book.
- Cleaning the documents in the standard way by removal of stop words, punctuation, and numbers.
- Definition of two document term matrices: one with words and the other one with words and bigrams. For these two matrices, some stemming was done, mainly to clean plurals. Furthermore, some additional stop words were removed, mainly words in context of the examples in Chap. 8.
- For both matrices, term frequency-inverse document frequencies (*TF-IDF*) were calculated.
- Definition of comparison clouds each based on 60 terms.
- Calculation of topic maps of order 2–8 for the term document matrices with stem terms.

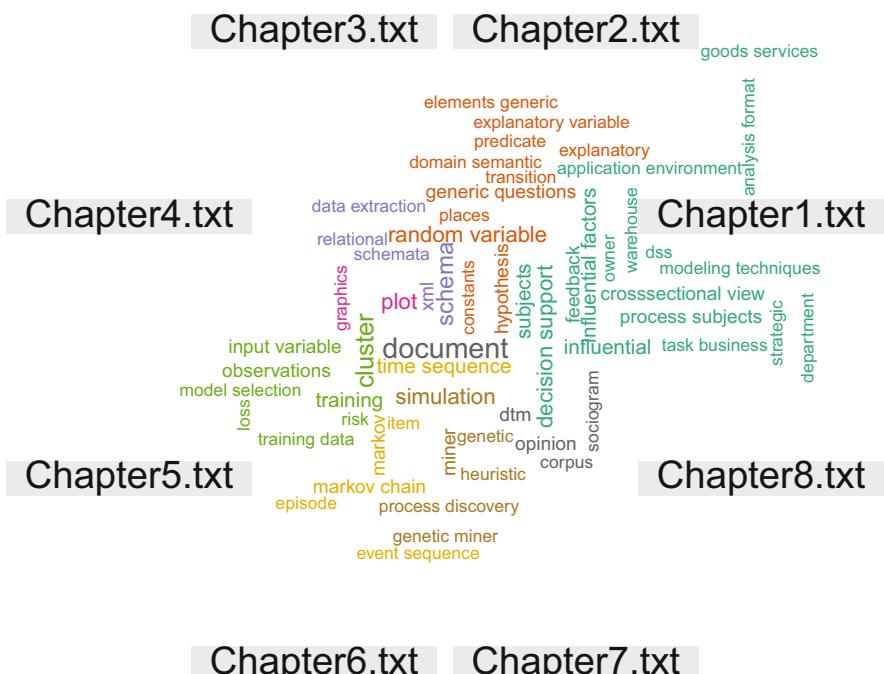
Probably the simplest way to obtain an overview is considering term frequencies. Looking at the ten terms with the highest frequencies, we decided to consider the term list *can*, *different*, *one*, *section*, and *using* as language-specific stop words. After dropping these terms, a sentence based on the ten most frequent words could summarize the book as follows (in brackets we show the frequency of the terms):

*Based on models (996) and mining (255), the book deals with goal (275)-oriented analysis (517) of data (1337), captured as variables (426), describing cases (344) of a business (693) process (892) in time (339).*

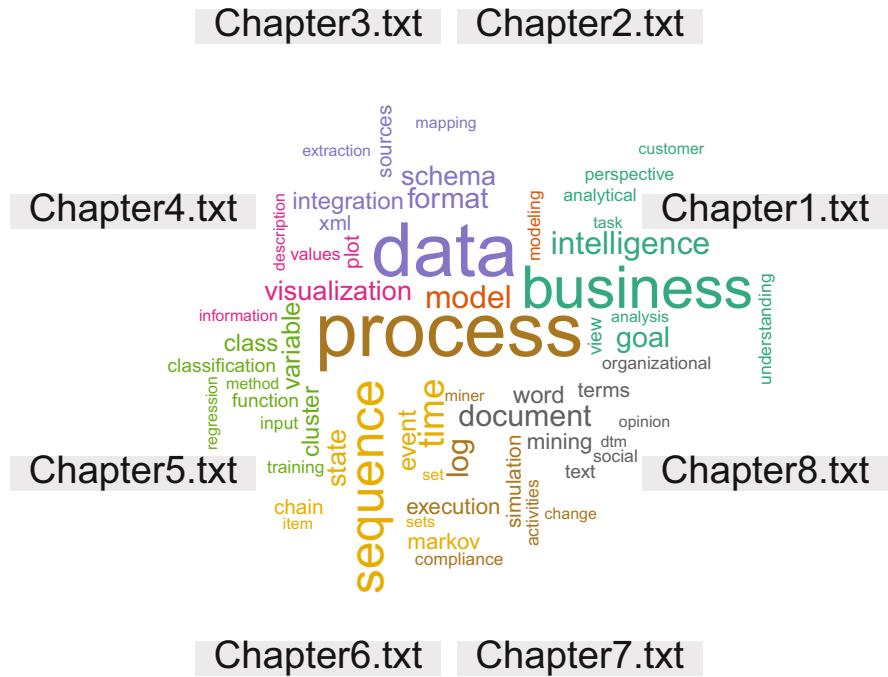
For a more detailed investigation of the organization of the book, let us first of all look at topics defined by the eight chapters. Based on the idea of topic models in Sect. 8.4.4, one can understand the different chapters as a mixture of different topics. For example, using a model with four topics would lead to the following assignment of chapters to topics: Chaps. 1 and 2 are assigned to one topic; Chap. 5 is assigned to a second topic; Chaps. 2, 4, and 6 are assigned to a third topic; and Chaps. 3 and 8 to a fourth topic. Using a model with eight topics results in a topic structure which is rather close to the organization of the chapters. This is in correspondence to our intention, and for a detailed summary, we will use the structure defined by the chapters.

Summarizing the contents of the chapters we will be done by writing a short description around the most likely terms in the eight topics according to the topic model. The visualization of the terms shown by the two comparison clouds in Figs. 9.1 and 9.2. In Fig. 9.1, we show the comparison cloud for the most frequent bigrams based on a matrix obtained by the term frequency-inverse document frequencies (*TF-IDF*) selection. Figure 9.2 shows the comparison based on the document term matrix *DTM* for single words without a selection.

The introduction in Chap. 1 traces the roots of BI from decision support systems, more oriented towards operations research, over the developments of data warehouses up to today's understanding of BI as a rather broad discipline



**Fig. 9.1** Comparison cloud based on the *TF-IDF* (R package *wordcloud*)



**Fig. 9.2** Comparison cloud based on the *DTM* (R package *wordcloud*)

for analyzing any kind of business activities. For the development of a unified umbrella for the diverse BI actions, we use a process-oriented definition of the term business applicable in many different domains. One can look at business activities from different perspectives, in particular the production perspective, the customer perspective, and the organizational perspective are identified. In connection with the perspective, it is often important to identify the roles of actors within the business process. Especially important actors in BI are the process subjects which generate process instances. The main input for all BI activities are data about the instances of business processes. These data are generated according to a specific view on the business process. Three views called event view, state view, and cross-sectional view are identified.

Using data as input, BI procedures start from a certain goal frequently formulated as key performance indicator (KPI). Such key performance indicators have to be seen in connection with the strategic use of BI inside the business. These strategies range from BI for achieving short-term targets with no connection to the management strategy over BI as a feedback for the overall management strategy up to using BI as a strategic resource for management decisions. For hitting the target, it is of utmost importance to identify analytical goals which allow a formal specification of the target in dependence of influential factors. The identification

of the analytical goals and definition of the application environment are the main tasks of business and data understanding which is central in the first phase of any BI project.

Based on business and data understanding, the following tasks of a BI project are organized in an appropriate analysis format. This book uses the iMine analysis format based on ideas of CRISP for data mining and L\* for process mining. Different use cases are used as examples showing the application of the ideas throughout the book.

Finding answers for analytical goals is usually based on models. Hence, modeling is an essential task in BI, and Chap. 2 discusses basic ideas about models and modeling. Contrary to many other scientific disciplines where models of phenomena are based on idealization, analogy, or definition of an equation for the relation between different variables, BI uses frequently a modeling approach characterized by the term *models of data*. This means that we want to learn the relation between a quantity of interest, i.e., in most cases a KPI, and the explanatory variables, i.e., candidates for influential factors, from the empirical data. For many BI activities, the term mining illustrates this modeling approach. Besides this approach, another access to modeling called *models of theories* is used, in particular in case of database models. It resembles the understanding of the term model in predicate logic which defines the model as an interpretation of a formal theory.

The formulation of a model requires a modeling language with a specific syntax, semantics, and notation. Moreover, such model languages allow the definition of a number of generic model elements and provide algorithms and procedures for answering questions about the properties of a model. We call the language together with some generic model elements and the algorithms a model structure. The art of modeling consists in the application of model structures to an application domain in such a way that the analytical goals can be answered in terms of the model. Besides modeling in the sense described above, patterns play an important role in many BI applications. In BI, the term pattern is used for describing local structures, for example, the co-occurrence of two events, or an interesting value of a variable outside the standard range.

Three important model structures are introduced in Chap. 2. The first one are models using logical structures. Prominent examples are ontologies and frames for setting the background for understanding the data in the domain. The second one are models based on the structures defined by graph theory. Such models are widely used, particularly in the case of business process modeling, for example, BPMN or Petri nets (indicated by then terms transition and places). The third modeling structure are analytical structures. This group encompasses a large number of different models ranging from calculus and its application in optimization over probabilistic structures, for example, Markov chains, up to statistical structures which are basic for the application of probability theory to empirical observation. Some core elements of statistical structures are the concepts of a random variable and testing statistical hypotheses.

An important issue for the interpretation of models is some basic understanding of the process of data generation, methods of measurement, data quality, and an understanding of the interpretation of time in connection with the data from business processes. These topics are briefly treated in Chap. 2. (These topics are missing in all formal summaries.)

Chapter 3 addresses the challenges with respect to the data provisioning task. Data provisioning is one of the most crucial and at the same time most complex tasks for any BI project. The data provisioning task can be divided into several subtasks, i.e., data collection and description as well as data extraction and data integration. The latter specifically comprises the selection of a target integration format.

Data collection and description is usually the first step where the decision falls which data sources are to be integrated into the analysis. Several criteria can be used to decide on the usage of a data source, e.g., accessibility (organizational and technical). The description of the data and the data sources is important for documentation, in particular for the overall data provisioning process, and for data understanding. Chapter 3 provides a template for documentation as well as practical experience from the medical domain.

The extraction of data refers to taking data from the sources and transferring it to some “analytical area,” e.g., the staging area in a data warehouse. A central approach in this context is extraction-transformation-load (ETL). This also includes transforming data derived from the sources in the sense of cleaning before the analysis. Loading refers to the transport of large data sets from one source to the other. Chapter 3 describes details of the ETL process and illustrates them by the implementation in Pentaho based on the health-care use case.

Special challenges stem from the advent of big data. Chapter 3 approaches the topic based on its challenges volume, variety, and velocity. Volume is addressed by new ways of storing data such as NoSQL and graph databases. Variety refers to the possible heterogeneity of data collected from different sources. Velocity subsumes how to extract and integrate streaming data, i.e., data that is continuously updated. In Chap. 3, different techniques for dealing with big data challenges are introduced. Please also refer to the discussion of tools supporting big data extraction, integration, and analysis in the Appendix.

The variety of data sources, big data or not, is the main driver for the step of data integration, i.e., the union and consolidation of data from different sources. In this chapter, approaches for schema mapping and matching are presented. Moreover, data integration and quality are important topics in this context. Integration is also interconnected with the choice of the integration format. The chapter discusses two formats, i.e., multidimensional table and log formats, and shows how the formats can be realized and which formats are meaningful for which analysis questions and data sources. For multidimensional structures, reporting and online analytical processing (OLAP) operations are discussed. Chapter 3 also addresses alternative ways for data integration such as linked data and mashups.

Overall, Chap. 3 indicates that the data provisioning process should be targeted at the analysis questions and can contain several cycles, i.e., depending on insights that are gained during the process one might need to return to earlier stages.

Any successful BI application essentially depends on data description and data visualization. This starts already in the first phase of business and data understanding and ends in the deployment phase of the analysis results. Chapter 4 takes a broad look on data description and visualization and introduces techniques for the visualization of the business process organized according to the information needs defined by information about business processes, information about instances of business processes, and information necessary for reporting.

In a first step, the chapter presents graph-based modeling and layout approaches for process model and process instance information. This is complemented with an evaluation of selected process modeling tools with respect to layout functionality. Specific challenges in the context of visualizing further aspects of business processes such as organizational or change information are discussed in the sequel.

For the description and visualization of process instances needed in data assessment and presentation of analysis results we use mainly techniques well known from statistical graphics. After a short summary about the principles of statistical graph grammars and basics about interactive and dynamic graphics, a number of basic visualization techniques are considered. As usual, the organization of the presentation is based on basic methods for qualitative information and graphics for quantitative information. An important aspect is the presentation of multidimensional data with an inherent data structure defined by the problem, i.e., nested variables. Furthermore, the method of dimensionality reduction using the method of principal components is considered. A number of methods are sketched aiming at the visualization of temporal data and dynamic visualization.

In case of methods for reporting, some methods for reporting metadata, in particular, reporting missing values and data quality, and reporting in connection with balanced scorecard are mentioned. The chapter concludes with interactive and dynamic graphics like motion charts, which are very popular nowadays under the heading *infographics*.

Without doubt, the analytical goals supervised and unsupervised learning are the core topics in business analytics and the most frequently used analysis techniques in BI. Chapter 5 is devoted to an introduction into these methods addressing cross-sectional data. Due to the fact that both methods are typical examples of techniques for models of data, it is important to develop a sound methodology for selecting a model out of a number of candidate models. In supervised learning, this selection is based on the concept of loss and risk which is well known from decision theory. The difference between training error and generalization error is carved out, and the splitting of the data into training, test, and validation is emphasized.

After these methodological introduction, the chapter treats methods for regression and classification as the two basic supervised learning problems in BI. In both cases, we present the basic ideas of the models, the different algorithmic solutions, and numerical as well as graphical methods for the evaluation of a solution. In case of regression, we focus on linear regression, neural networks, and nonparametric regression. As methods for classification, naive Bayes, logistic regression, tree-based methods, nearest neighbor classification, support vector machines, and boosting are explained. A short demonstration of all methods for

the data from the use cases can be found in the text. A detailed demonstration of the methods can be found on the homepage of the book.

The final section of the chapter deals with cluster analysis as the major application of unsupervised learning. The differences in the evaluation of solutions between classification and clustering are outlined. From the numerous methods for classification, we present hierarchical clustering, partitioning methods, and model-based clustering. As in the case of supervised learning, ideas are discussed of how to evaluate the results of the analysis with numerical and graphical techniques. Again, the results for the examples are sketched based on data for the use cases. Details can be found on the homepage.

Chapter 6 deals with analysis methods for temporal data. The temporal structure defines data as either event sequences in the event view or time sequences in the state view. Many a time, these data are not used in its original form but are transformed for modeling and analysis. Different models can be defined corresponding to the data structure. Besides statistical and probabilistic models, patterns of sequences of events are of interest. These models allow a more detailed formulation of the general analytical goals description and prediction.

In detail, the following models and analysis methods are explored in Chap. 6:

- In case sequences of states, an important probabilistic model are Markov chains allowing the formulation of a number of analytical goals. In particular, the estimation problem for the parameters of the chain is considered, and applications in the case of Web mining are briefly outlined.
- For prediction problems in connection with time, we consider descriptive analysis and regression models for time-to-event analysis.
- For classification and clustering, we consider time warping and response feature analysis as important transformations which allow the application of the methods developed in Chap. 5 for cross-sectional data to time sequences.

In case of event sequences, the analysis of pattern plays an important role. Such patterns typically describe associations between the advents of one or more events, i.e., if one event occurred, one or more other events also occurred. Chapter 6 introduces association analysis as analysis technique to detect such patterns in event sequences. Association analysis is then extended towards sequence mining where not only the occurrence of events in patterns is analyzed but also their order. Finally, Chap. 6 features episode mining which operates on event streams and enables the detection of more complex patterns (episodes) which might even include parallel occurrences of events.

Chapter 7 starts with an introduction and classification of terminology used in the context of process analysis. At first, techniques for the analysis at the process model level are presented. They can be subdivided into static and dynamic analysis techniques. Static techniques evaluate information that is available within the process model and enable, for example, the detection of media breaks or expected process costs. Dynamic analysis techniques are based on process simulation, i.e., the artificial creation and execution of a number of process instances. Simulations typically result in protocols. These protocols can then be analyzed in a cumulative

way, i.e., over several instances or for single process instances. Dynamic analysis is predominantly quantitative and focuses on time and costs. Based on the results of static and dynamic analysis process, optimizations might be suggested such as reordering tasks or (re)allocating resources.

The analysis of processes is continued with a discussion of process performance management and process warehousing. Process performance management refers to the definition, supervision, and evaluation of key performance indicators specific to process execution, for example, monitoring the throughput time of process instances and alerting the user in case of exceeding deadlines. Process warehousing is an approach to combine process analysis with multidimensional data structures that are typical for data warehousing applications. Based on the resulting “process cubes,” one can employ OLAP and data mining techniques.

Process performance analysis spans the bridge from analysis techniques at the process model level and process performance management during runtime to process mining which typically comprises techniques for the *post mortem* analysis of processes. Process mining techniques operate on process logs. Chapter 7 presents techniques for process discovery, change mining, and conformance checking. The discovery techniques range from explaining the basic approach based on the  $\alpha$ -algorithm over the frequency-based heuristic miner to the genetic miner which is based on evolutionary algorithms.

Chapter 7 also discusses business process compliance. This area raises many practically relevant questions such as “is my process compliant with a relevant regulation?” Compliance can be an important task throughout the process life cycle, i.e., it has to be checked for design and runtime or even *post mortem*.

The chapter closes with a discussion on evaluation methods for process analysis techniques, particularly for process mining and compliance checking. The evaluation is based on different metrics defined on top of process models that enable the assessment of the quality of the mining result.

Chapter 8 is devoted to the analysis of models and methods which cover multiple business perspectives. Specifically, this chapter presents techniques that can be applied to questions that arise at the intersections between different BI perspectives. A first example is the intersection between the perspectives production and organization where questions arise such as “who was delegating the patient treatment to Sara during skin cancer aftercare?” The techniques that can be applied to answer this question are subsumed by organizational mining. In order to prepare the discussion for organizational mining, Chap. 8 starts off with the introduction of social network analysis techniques. These techniques are interesting per se, specifically in the context of analyzing, for example, social networks such as Facebook or Twitter, but they also provide, together with process mining, the ingredients for organizational mining. Chapter 8 explains basic terminology and metrics that are typically applied in social network analysis. Based on this, organizational mining techniques such as role hierarchy and social network mining are introduced.

The next interface to be addressed in Chap. 8 is the one between the perspectives production and customer by introducing decision point analysis. Based on mining processes (production perspectives), the rules behind decision points within the

processes (e.g., taking different treatments based on the patient age) are mined using decision trees (cross-sectional analysis). The basic approach is introduced and extensions towards more complex decision rules based on time series data are presented.

The final section of Chap. 8 deals with text mining which has accumulated importance in the last 15 years. Text mining can be applied in order to answer questions that refer to the interface between perspectives organization and customer. In the area of text mining, the transformation of semistructured text into formal structures allowing the application of algorithmic analysis methods is of utmost importance. From the numerous approaches in text mining, summarized often under the heading *text analytics*, we describe in detail the methods based on the document term matrix. This representation of the text allows the application of the methods of supervised and unsupervised learning of Chap. 5 to text data. At first sight, the document term matrix seems to be a rather simple model, but it allows a number of powerful analysis methods. Finally, we show how the basic statistical methods can be augmented by other approaches leading to methods for automatic keyword extraction and to opinion mining which became highly popularity with the advent of social platforms.

All chapters employ the four use cases presented in the introduction on patient treatment, higher education, logistics, and customer relationship management. From these use cases, data and selected exercises for all chapters are provided on the book's homepage [www.businessintelligence-fundamentals.com](http://www.businessintelligence-fundamentals.com).

In addition, all chapters refer to tools that support the different BI tasks and techniques. The focus was put on open source tools in order to enable BI projects to be free of costs, e.g., in lectures. An overview and evaluation of the open source tools are provided in the Appendix.

# Appendix A

## Survey on Business Intelligence Tools

In the following, tools and systems are provided and described that support the BI tasks *data modeling and understanding* (cf. Chap. 2) and *ETL support* (cf. Chap. 3); *big data and cloud* (cf. Chap. 3); *visualization, visual mining, and reporting* (cf. Chap. 4); *data mining* (cf. Chaps. 5 and 6); *process analysis* (cf. Chaps. 7 and 8); and *text mining* (cf. Chap. 8).

Each tool description will be based on the following schema (cf. Table A.1). The schema features general criteria which can be more detailed for special purposes, e.g., mobile access or dashboard creation. An abundance of commercial and noncommercial tools tailored for specific purposes can be found on the Web. Every decision depends to some extent on personal preferences.

*Tool collection:* The tool descriptions shall cover all tools used throughout the book, complemented with further tools. Please note that the focus is on open source tools.

### A.1 Data Modeling and ETL Support

The open-source tools mentioned in Chaps. 2 and 3 are COMA 3.0 and Protègè for data modeling and schema integration as well as Pentaho Spoon and Talend for supporting the entire ETL process.

In Table A.2, Protègè and COMA 3.0 are described which provide support for the data modeling and schema integration. Protègè, for example, enables the definition of ontologies for data modeling that can also be used in order to resolve, for example, ambiguities for later schema integration. Plug-in PROMPT for Protègè additionally enables the mapping between ontologies. COMA 3.0 supports schema matching. Similar functionality is offered by, for example, Altova Mapforce [21]. Here, two schemas can be loaded and matched manually. The respective

**Table A.1** Description schema for selected BI tools

<i>Availability</i>
Link, url
Existing documentation such as white papers
Licensing
Existing evaluations
<i>Technical criteria</i>
Operating system
Supported data formats
Extensibility
<i>User interfaces</i>
Evaluation
<i>Functionality</i>
Algorithms, techniques, visualizations
Data export/import, interfaces
Data preprocessing
Interactivity
Community, e.g., forum, blog

transformation queries or statements can be produced automatically in different ways, e.g., XQuery of Java. Altova MapForce is not open source, but commercial.

Note that both tools, Protègè and COMA 3.0, feature a Web-based version, i.e., WebProtege and COMA+ Web Edition 0.5.

The following Table A.3 provides an overview for Pentaho and Talend. Note that Pentaho is a full-fledged BI suite, i.e., consists of the *Business Analytics Platform*, *Kettle* as ETL tool, the *Report Designer*, and the *Marketplace*, complemented by further functionality provided by, for example, the *Aggregation Designer* [12].

Many open sources and commercial tools for supporting tasks of the extraction-transformation-load (ETL) process are available (for an overview, see, e.g., [10, 11]). To name a few, other open-source solutions are CloverETL [17] and Jitterbit [18].

## A.2 Big Data

Tools on big data follow two directions, i.e., (a) supporting the analysis of big data and (b) supporting the extraction and integration of big data. Moreover, it is important to distinguish between tools and techniques. MapReduce, for example, is more a technique than a tool. It supports, for example, (a) by fragmenting an analysis job into smaller jobs for which their results are aggregated afterwards. Different tools offer MapReduce implementations, for example, Apache Hadoop [23] (open source). Also as mentioned in Sect. A.1, Pentaho offers support for big

**Table A.2** Tools supporting schema integration: Protégè and COMA 3.0

	Protégè	COMA 3.0
<i>Availability</i>		
Link, url	[19]	[20]
Existing documentation	[22]	[2]
Licensing	MPL	AGPL
Existing evaluations	Several scientific papers, e.g., [3]	Several scientific papers, e.g., [1]
<i>Technical criteria</i>		
Operating system	Protégè Desktop: Linux, Windows, Mac OSX; WebProtege: Web-based	COMA 3.0: Windows, Linux; Coma+ Web Edition 0.5 web-based;
Supported data formats	OWL 2	SQL
Extensibility	Development of plugins based on OSGi	Java-based API
<i>User interfaces</i>		
Evaluation	Graphical GUI	Graphical GUI
<i>Functionality</i>		
Algorithms, techniques, visualizations	Reasoning can be connected	Support of different matching strategies
Data export/import, interfaces	Data export/import: RDF, XML, OWL	XSD, OWL
Data preprocessing		
Interactivity		Mapping candidates can be adapted by users
Community, e.g., forum, blog	Protégè and COMA 3.0 are both supported by a variety of documentations and fora	

data analytics via so-called Hadoop Shims [12]. Table A.4 compares two tools for big data analytics, i.e., Pentaho and H2O.

For the extraction and integration of big data (b), challenges volume, variety, and velocity are vital, as well. Let us first comment on challenge volume. As discussed for big data analytics, volume has been tackled among other approaches by suggesting NoSQL databases such as key value stores or graph databases. Chapter 3 mentions sonesDB which is a graph database. As sonesDB is nice for illustration, but seems to be no longer supported, in this section, we will introduce OrientDB as graph database. For challenge variety, Chap. 3 discussed BaseX as XML database. Finally, tackling velocity, Apache Storm supports the analysis of streaming data, i.e., data that is continuously injected into the analytical database and hence addresses the big data challenge velocity. Table A.5 summarized and compares the representatives mentioned before, i.e., OrientDB, BaseX, and Apache Storm.

**Table A.3** ETL tools: Pentaho and Talend

	Pentaho	Talend
<i>Availability</i>		
Link, url	[12]	[13]
Existing documentation such as white papers	[14]	[15]
Licensing	GPLv2, LGPL, Apache, depending on the version	Basic: open source, extended functionality: commercial
Existing evaluations	[6]	Four software tests (in German) [16]
<i>Technical criteria</i>		
Operating system	Linux, Windows, Mac OSX	Linux, Windows, Mac OSX
Supported data formats	Variety, e.g., XML, SQL, text, csv	Variety of data formats, e.g., XML, SQL, text, csv, and standards, e.g., BPMN
Limitations		
Extensibility	Java-based API; Pentaho Marketplace stimulates testing and exchange of developed plug ins	Several extensions possible based on, e.g., java-based API, Web services
<i>User interfaces</i>		
Evaluation	ETL processes can be designed and traced in a graphical way	
<i>Functionality</i>		
Algorithms, techniques, visualizations	With Pentaho business analytics platform and report designer, various analyses and reports/visualizations can be created; in particular, aggregation designer supports OLAP analysis	Reporting and dashboards are not supported in the open source version
Data export/import, interfaces	Various interfaces to many of the existing tools/systems, e.g., databases, excel, XML	Talend specifically powers the connection to NoSQL sources such as Hive, MongoDB, Apache
Data preprocessing	Pentaho report designer enables reporting on data through the integration process, hence fosters data understanding; data can be preprocessed through Kettle	In the open source solutions, Talend open studio for data quality offers data profiling as well as graphical charts on the data in order to foster data understanding
Interactivity	Strongly supported throughout all phases of the BI process	
Community, e.g., forum, blog	Pentaho and Talend are both supported by a variety of documentations and fora	

**Table A.4** Big data analytics tools: Pentaho and H2O

	Pentaho	H2O
<i>Availability</i>		
Link, url	[12]	[24]
Existing documentation such as white papers	[14]	[27]
Licensing	GPLv2, LGPL, Apache 2.0, depending on the version	Basic: Apache 2.0
Existing evaluations	[6]	Benchmarks [28]
<i>Technical criteria</i>		
Operating system	Linux, Windows, Mac OSX	Java-based platform, Web interface
Supported data formats	Variety, e.g., XML, SQL, text, csv	Local sources, Hadoop, EC2, multiple nodes
Extensibility	Java-based API; Pentaho Marketplace stimulates testing and exchange of developed plug ins	APIs to R and JSON
<i>User interfaces</i>		
Evaluation	Graphical UI	Graphical UI
<i>Functionality</i>		
Algorithms, techniques, visualizations	With Pentaho business analytics platform and report designer, various analyses and reports/visualizations can be created; in particular, aggregation designer supports OLAP analysis	Variety of analysis algorithms and techniques, e.g., regression, classification, neural networks
Data export/import, interfaces	Hadoop distributions via abstraction layer (shim); provision of predefined shims, but not for open source distribution	Import of csv, SQL
Data preprocessing	Reporting and transformation functions on different Hadoop clusters, e.g., Hive;	n.a.
Interactivity	Graphical UI	Graphical UI
Community, e.g., forum, blog	All supported by a variety of documentations and fora	

**Table A.5** Big data integration tools: OrientDB, BaseX, and Apache Storm

	OrientDB	BaseX	Apache Storm
<i>Availability</i>			
Link, url	[29]	[30]	[25]
Existing documentation such as white papers	Documentation available on [29]	[31]	[26]
Licensing	Apache 2.0	BSD	Apache 2.0
Existing evaluations			
<i>Technical criteria</i>			
Operating system	Linux, Windows, Mac OSX	Linux, Windows, Mac OSX	Java-based framework
Supported data formats	Key value pairs, graphs	XML	Streams of key value pairs
Extensibility	Several APIs, e.g., Java API, SQL	Java-based API	Implementation in java or another language possible
<i>User interfaces</i>			
Evaluation	Graphical UI, Web frontend	Graphical UI	No GUI
<i>Functionality</i>			
Algorithms, techniques, visualizations	Supported query languages: SQL and Gremlin (graph-based)	Tree-based visualization of XML documents; support of XPath and XQuery	Enables the integration of data streams from different sources
Data export/import, interfaces	Import from RDBMS and Neo4J (graph database)	Import: XML, export: XML, HTML, csv	Can be used to feed streaming data into other systems such as Hive
Interactivity	Query language	Query language	n.a.
community, e.g., forum, blog	All supported by a variety of documentations and fora		

### A.3 Visualization, Visual Mining, and Reporting

Modeling and layouting process models and instances is described in Sect. 4.2, and several tools are mentioned. As these tools provide much more functionality, an evaluation of their layouting functionality is presented directly in Sect. 4.2.2.

For the visualization of cross-sectional data, Chap. 4 used a number R packages for graphics, in particular, the packages `lattice` and `ggplot2`. The latter is probably one of the most advanced tools for producing statistical and other graphics. A tool for dynamic graphics for data exploration is GGobi [33]. GGobi can be used as stand-alone software or in connection with R in the package `ggobi`.

For dynamic and interactive graphics, the application of HighChart was shown. HighChart is a Javascript library which requires a HTTP server for local visualiza-

**Table A.6** Visualization tools: R, HighCharts, Tableau Public

	R-graphics	HighChart	Tableau Public
<i>Availability</i>			
Link, url	[5]	[34]	[35]
Existing documentation such as white papers	On the website, [9]	On the website tutorial and publications	On the website tutorial
Licensing	GPL	Creative commons-NonCommercial	Free
<i>Technical criteria</i>			
Operating system	Linux, Mac OS, Unix, Windows	Javascript,jQuery, HTTP-Server	Windows, Mac OS X
Supported data formats	csv, excel	csv, excel, json, xml	csv, excel
Extensibility	Yes		
<i>User interfaces</i>			
Evaluation	Command line	JavaScript	GUI
<i>Functionality</i>			
Algorithms, techniques, visualizations	Statistical graphics, dynamic graphics for data exploration	Interactive graphics, dashboards	Interactive graphics, dashboards
Data export/import, interfaces	Interface to all DB	Export to JPEG, PNG, pdf, SVG	Web
Data preprocessing	Yes	Yes	Yes
Interactivity	Yes	Yes	Yes
Community, e.g., forum, blog	All supported by a variety of documentations and fora		

tion. For personal use and nonprofit organizations, high chart is freely available. An open-source tool for reporting and infographics is Tableau Public. Tableau Public has an easy-to-use interface and proposes a data visualization after parsing the uploaded data. Afterwards, the user can customize this basic layout in drag-and-drop style. The produced infographic can be published on the Web.

From the commercial products for visualization of cross-sectional data, we want to mention the SAS data mining software and the IBM SPSS Modeler which integrate visualization in the data mining activities. For an overview on R-graphics, HighChart, and Tableau Public see Table A.6.

There are numerous tools for Web-based graphics and infographics. Table A.7 lists ManyEyes, Gapminder, and Piktochart.

ManyEyes is an advanced visualization tool from IBM. The main emphasis is on sharing graphics within the ManyEyes community. Users can create their own graphics in easy steps or modify the graphics of other community members.

Gapminder is based on the Trendalyzer software developed for the animated presentation of statistics, so-called *motion charts*. These charts show impressively the development of demographic, economic, or environmental facts. Many time

**Table A.7** Web based visualizations: ManyEyes, Gapminder, Piktochart

	ManyEyes	Gapminder	Piktochart
<i>Availability</i>			
Link, url	[36]	[37]	[38]
Existing documentation such as white papers	Not much documentation available, introduction see [8]	On website	On website
Licensing	Free, data and visualizations are directly shared, copyright should be cleared	Free	Free
<i>Technical criteria</i>			
Operating system	Web browser		
Supported data formats	csv, spread sheet	Google spreadsheet	csv, spreadsheet
Extensibility			
<i>User interfaces</i>			
Evaluation	Interactive graphical user interface		
<i>Functionality</i>			
Algorithms, techniques, visualizations	Various basic layouts for graphics		
Data export/import, interfaces	Web publishing or download		
Data preprocessing	Limited		
Interactivity	Interactive editing of visualization		
Community, e.g., forum, blog	All supported by a community		

series at the national level as well as from international organizations are available on the site. The Trendalyzer software is now available as interactive chart in the Google spreadsheet. This allows users the creation of motions charts with their own data.

Piktochart is an easy-to-use tool for creation of infographics. Numerous templates for infographics are available which can be adapted by the user. The created infographics allow interactive elements and are readable by search engines.

In addition, there are several other tools that enable the creation of infographics, e.g.:

- <http://www.hongkiat.com/blog/infographic-tools/>
- <http://www.coolinfographics.com/tools/>
- <http://www.fastcodesign.com/3029239/infographic-of-the-day/30-simple-tools-for-data-visualization>

## A.4 Data Mining

In this book, we used R for data mining applications. Strictly speaking, R is a programming language for statistical computing and statistical graphics. It is a popular data mining tool for scientists, researchers, and students. Consequently, there exists a large community with forums and blogs which helps to learn how to use the numerous packages necessary for data mining. Besides data mining, a rich set of statistical methods for data preparation and graphics is available. R has strong object-oriented programming facilities which allow the extension of the software as soon as one has mastered the R language.

For usage of R as a BI production tool, the package `DBI` offers an interface to relational database systems. For big data, a number of solutions are provided. The package `data.table` is a fast tabulation tool as long as the data fit in the memory, e.g., 100 GB in RAM. For using Hadoop and the MapReduce approach, a number of packages have to be installed. For details, we refer to [4]. For a number of algorithms, there are also parallel implementations available.

From a more practical point of view, big data problems can be handled by sampling data from a database, develop a decision rule for the sample, and deploy the learned rule afterwards in the database. Thus, R can be used as an analysis tool in connection with an analytical sandbox. Alternatively, many a time it may be useful to aggregate the data and analyze the aggregated data.

Basically R is command line oriented, but a number of GUIs exist. For the development, RStudio offers an IDE, for data mining the Rattle GUI can be used, and Revolution Analytics provides a visual studio-based IDE. Further, the RWeka interface facilitates the application of Weka data mining algorithms in within R.

Weka is a Java-based data mining software which offers analysis tools similar to R. It also provides numerous data preprocessing techniques. With respect to data visualization, the facilities are not so comprehensive. The main user interface of Weka is the *Explorer* which provides in several panels access to the different data mining tasks. There exist panels for preprocessing, for variable selection, for visualization, and for different data mining techniques like classification, clustering, or association analysis. Weka supports two other BI tools: the Pentaho Business Analytics Platform uses Weka for data mining and predictive analytics; inside ProM Weka can be used for data mining, for example, in decision point analysis.

As a third open-source data mining software, we want to mention RapidMiner. Due to the fact that it has an easy-to-use interface, it is one of the most popular data mining tools in BI. It captures the entire life cycle of a BI application, allows model management, and is well designed for the collaboration between the business analyst and the data scientist. With respect to analysis capacities, it offers algorithms for data preparation and for analysis. Algorithms from external sources like R or Weka can be included in the analysis. Further, it supports the analysis of data in the memory, in databases, in the cloud, and supports Hadoop. For an overview on R, RapidMiner, and Weka see Table A.8.

**Table A.8** Data mining tools: R, RapidMiner, Weka

	R	RapidMiner	Weka
<i>Availability</i>			
Link, url	[5]	[40]	[42]
Existing documentation such as white papers	On the website: manuals, R journal, FAQs	On the website: documentation	[43]
Licensing	GPL	AGPL	GPL
<i>Technical criteria</i>			
Operating system	Linux, (Mac) OSX, Windows	All platforms (Java based)	All platforms (Java based)
Supported data formats	Basically csv, but various other data formats are supported		
Extensibility	Yes	Yes	Yes
<i>User interfaces</i>			
Evaluation	Command line/various GUIs	GUI	Command line, various GUIs
<i>Functionality</i>			
Algorithms, techniques, visualizations	Algorithms for all data mining algorithms, various visualization techniques		
Data export/import, interfaces	Interfaces to all DB systems		
Data preprocessing	Supported by various algorithms		
Interactivity	Depending on the application		
Community, e.g., forum, blog	[32]	[41]	[44]

From the commercial products, the SAS data mining software and the IBM SPSS Modeler are two powerful data mining tools. Both products offer a visual interface and allow applications without programming.

## A.5 Process Mining

Table A.9 summarizes details on the process mining tool ProM which is applied in Chap. 7. There is no comparable tool available as open-source solution; hence, only ProM is introduced here. Nonetheless, one can mention Disco [45] as commercial process mining tool which developed from ProM.

**Table A.9** Process mining tool: ProM

ProM	
<i>Availability</i>	
Link, url	[39]
Existing documentation such as white papers	[7]
Licensing	
Existing evaluations	ProM 5.2: CPL, ProM 6.2: LGPL, ProM 6.3: LGPL, ProM 6.4: GPL
<i>Technical criteria</i>	
Operating system	All platforms
Supported data formats	Log formats: MXML, XES, and csv; process model formats: PNML, YAWL specification, BPEL, CPN
Extensibility	Development of java-based plug ins
<i>User interfaces</i>	
Evaluation	n.a.
<i>Functionality</i>	
Algorithms, techniques, visualizations	Several algorithms for process discovery, conformance checking, filtering, organizational mining, etc.; visualizations as, e.g., graph-based process models or dotted charts
Data export/import, interfaces	Import: MXML, XES, csv, PNML, etc.; export: process models as graphics, e.g., eps, svg; Petri Nets: pnml; logs: MXML, XES; reports: HTML
Data preprocessing	Filtering
Interactivity	Partly, e.g., mouseover and dragging of social networks
Community, e.g., forum, blog	Supported by fora, developer support, ProM task force

## A.6 Text Mining

All the data mining software products reviewed in Sect. A.4 offer text mining facilities for classification and cluster analysis of text data represented as document term matrices. The applicability of these tools essentially depend on the data sources which can be read by the software, the available transformations for preprocessing, the availability of linguistic knowledge for the language under consideration, and the analysis algorithms. For example, the package `tm` can process a number of formats by using plugins. Regarding the linguistic knowledge, part-of-speech tagging and stemming can be done, and WordNet can be accessed as English lexical database. For analysis, a number of advanced statistical models like topic maps or specific classification and cluster algorithms can be used.

An open-source tools which puts more emphasis on natural language processing is **GATE**. GATE stands for General Architecture for Text Engineering and was

developed at the University Sheffield. On the homepage [46], one can find an extensive documentation.

GATE consists of a number of components. A core component of GATE is an information extraction tool which offers modules for tokenization, part-of-speech tagging, sentence splitting, entity identification, semantic tagging, or referencing between entities. A number of plugins offer applications for data mining algorithms or the management of ontologies. Another important component allows indexing and searching of the linguistic and semantic information generated by the applications.

GATE supports analysis of text documents in different languages and in various data formats. The GATE Developer is the main user interface that supports the loading of documents, the definition of a corpus, the annotations of the documents, and the definition of applications.

## References

1. Arnold P, Rahm E (2014) Enriching ontology mappings with semantic relations. *Data Knowl Eng* 93:1–18
2. COMA 3.0 CE, program description (2012) Database chair. University of Leipzig, Leipzig
3. Fridman NN, Tudorache T (2008) Collaborative ontology development on the (semantic) web. In: Symbiotic relationships between semantic web and knowledge engineering. Papers from the 2008 AAAI spring symposium, Technical Report SS-08-07, AAAI
4. Prajapati V (2013) Big data analytics with R and hadoop. <http://it-ebooks.info/book/31577>. Accessed 11 Nov 2014
5. R Core Team (2014) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. <http://www.R-project.org>. Accessed 12 Dec 2014
6. Tunçer O, van den Berg J (2012) Implementing BI concepts with Pentaho, an evaluation. Delft University of Technology, Delft
7. van der Aalst WMP (2011) Process mining: discovery, conformance and enhancement of business processes. Springer, Heidelberg
8. Viegas FB, Wattenberg M, van Ham F, Kriss J, McKeon M (2007) Manyeyes: a site for visualization at internet scale. *IEEE Trans Vis Comput Graph* 13(6):1121–1128
9. Wickham H (2009) ggplot2: Elegant graphics for data analysis. Springer, New York
10. [http://www.databaseanswers.org/modelling\\_tools.htm](http://www.databaseanswers.org/modelling_tools.htm). Accessed 4 Dec 2014
11. <http://www.etltools.net/free-etl-tools.html>. Accessed 4 Dec 2014
12. <http://community.pentaho.com/>. Accessed 4 Dec 2014
13. <http://www.talend.com/products/big-data>. Accessed 4 Dec 2014
14. <http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation>. Accessed 4 Dec 2014
15. <http://www.talendforge.org/tutorials/menu.php>. Accessed 4 Dec 2014
16. [https://de.talend.com/resources/whitepapers?field\\_resource\\_type\\_tid\[%\]=79](https://de.talend.com/resources/whitepapers?field_resource_type_tid[]=%79). Accessed 4 Dec 2014
17. <http://www.cloveretl.com/>. Accessed 4 Dec 2014
18. <http://www.jitterbit.com/>. Accessed 4 Dec 2014
19. <http://protege.stanford.edu/>. Accessed 4 Dec 2014
20. <http://dbs.uni-leipzig.de/Research/coma.html>. Accessed 4 Dec 2014
21. <http://www.altova.com/mapforce.html>. Accessed 4 Dec 2014
22. <http://protegewiki.stanford.edu/wiki/ProtegeDesktopUserDocs>. Accessed 4 Dec 2014

23. <http://hadoop.apache.org/>. Accessed 5 Dec 2014
24. <http://docs.0xdata.com/>. Accessed 5 Dec 2014
25. <https://storm.apache.org/>. Accessed 5 Dec 2014
26. <https://storm.apache.org/documentation/Home.html>. Accessed 5 Dec 2014
27. <http://docs.0xdata.com/>. Accessed 5 Dec 2014
28. <http://docs.0xdata.com/benchmarks/benchmarks.html>. Accessed 5 Dec 2014
29. <http://www.orienttechnologies.com/orientdb/>. Accessed 5 Dec 2014
30. <http://basex.org/>. Accessed 5 Dec 2014
31. [http://docs.basex.org/wiki/Main\\_Page](http://docs.basex.org/wiki/Main_Page). Accessed 5 Dec 2014
32. <http://www.inside-r.org/>. Accessed 12 Dec 2014
33. <http://www.ggobi.org/>. Accessed 5 Dec 2014
34. <http://www.highcharts.com/>. Accessed 12 Dec 2014
35. <http://www.tableausoftware.com/public/>. Accessed 12 Dec 2014
36. <http://www-969.ibm.com/software/analytics/maneyes/>. Accessed 9 Dec 2014
37. <http://www.gapminder.org/>. Accessed 12 Dec 2014
38. <http://piktochart.com/>. Accessed 12 Dec 2014
39. <http://www.processmining.org/>. Accessed 11 Dec 2014
40. <https://rapidminer.com/>. Accessed 12 Dec 2014
41. <http://forum.rapid-i.com/>. Accessed 12 Dec 2014
42. <http://www.cs.waikato.ac.nz/ml/weka/>. Accessed 12 Dec 2014
43. <http://www.cs.waikato.ac.nz/ml/weka/book.html>. Accessed 12 Dec 2014
44. <http://www.cs.waikato.ac.nz/ml/weka/help.html>. Accessed 12 Dec 2014
45. <http://www.fluxicon.com/disco/>. Accessed 19 Dec 2014
46. <https://gate.ac.uk/>. Accessed 12 Dec 2014

# Index

- Absorbing state, 226  
Accuracy (data quality), 79  
Activity (business process), 54, 248, 257  
Actor, 54, 256, 277  
Adjacency matrix, 52, 225  
Adjusted R-squared, 164  
Agglomerative method, 196  
Aggregation, (data schema), 100  
 $\alpha$ -Algorithm, 256, 257, 262  
Alternative hypothesis, 69, 162  
Analysis technique, 20, 119  
Analytical business model, 17  
Analytical format, 98  
Analytical goal, 12, 38, 42, 211, 214, 217, 235  
Analytical sandbox, 23, 337  
Analytical technique, 21, 41, 159  
Animation, 250  
Auditing, 266  
Authority, 229  
Average linkage, 195
- Backpropagation, 168  
Bagging, 184  
Bag of concepts, 312  
Bag of words, 299  
Balanced score card, 5, 150  
Bandwidth, 169  
Bar chart, 134  
Bayes Theorem, 65, 178  
Bias-variance trade-off, 158  
Big data, 93  
Bigrams, 298  
Binomial distribution, 66  
Bins, 130, 137
- BI perspectives, 6, 12, 15, 17, 19, 41, 120, 123  
Biplot, 143  
Boosting, 190  
Bootstrap, 184  
Boxplot, 137, 138, 144  
BPMN, *See* Business Process Modeling and Notation (BPMN)  
Business  
    analytics, 2, 3, 21  
    cockpit, 149  
    model, 4, 23  
    understanding, 16  
Business process, 6, 11, 12, 36, 39, 119  
    compliance, 246  
    views, 8  
Business Process Modeling and Notation (BPMN), 54, 121, 247
- CART, 183  
Causal matrix, 261  
Censored data, 220  
Centrality, 280  
Chapman–Kolmogorov equations, 225  
Circle, 52  
Circular layout, 281  
Clarity of a model, 44  
Classification, 156  
Closeness, 281  
Cluster, 193  
    analysis, 193  
    tree, 196  
Co-clustering, 305  
Coherence (data quality), 79  
Comparability of a model, 44

- Comparison cloud, 301  
 Complete linkage, 195  
 Completeness (data quality), 79  
 Concept drift, 258  
 Conceptual modeling, 41  
 Conditional distribution, 63  
 Confidence, 235
  - bands, 69, 141, 170
  - interval, 69
  - regions, 69
 Conformance checking, 246, 255  
 Confusion matrix, 174  
 Consistency (data quality), 79  
 Contour plot, 138  
 Control flow, 9, 55, 122  
 Coordinates (visualization), 130  
 Corpus, 295  
 Correctness of a model, 44  
 Correlation, 65, 140, 146, 163  
 Correlation matrix, 143  
 Covariance, 65  
 Cox regression, 223  
 Critical layer, 96  
 CRM use case
  - classification, 191
  - clustering, 200
  - data quality, 148
  - description, 30
  - prediction, 165, 166
  - principal components, 143
  - variable description, 138
 Cross entropy, 174  
 Crossover, 262  
 Cross-sectional view, 9, 10, 12, 16, 120, 129, 155  
 Cross-validation, 170  
 Cross-validation, k-fold, 176  
 Curse of dimensionality, 160, 163, 178  
 Customer perspective, 6
- Daisy, 194  
 Dashboard, 149  
 Data
  - cleaning, 81, 113
  - flow, 55
  - fusion, 112
  - integration, 108
  - mashup, 114
  - modeling technique, 15
  - provenance, 115
  - quality, 113, 120, 147
 understanding, 119  
 understanding technique, 16  
 variety, 93, 96  
 velocity, 93, 95  
 veracity, 93, 96  
 volume, 93, 94
- Degree, 52
  - centrality (sociogram), 280
  - of a node, 280
 Delta analysis, 314  
 Dendrogramm, 196  
 Density, 280, 281  
 Density estimate, 137  
 Dependent variable, 59  
 Deviance, 174  
 Dice (OLAP), 103  
 Dimension, 100  
 Directed graph, 52, 278  
 Dirichlet distribution, 228  
 Distance-based method, 193  
 Distance, in graphs, 281  
 Distribution
  - continuous, 63, 146
  - discrete, 63
  - empirical, 68
 Distribution function, 62  
 Document term matrix (DTM), 299  
 Domain semantics, 41  
 Drill across, 103  
 Drill down, 100, 103  
 Dublin Core (DCMI), 294  
 Dummy variable, 73, 193, 202  
 Dyad (sociogram), 277  
 Dynamic process analysis, 248  
 Dynamic time warping, 215
- EBMC<sup>2</sup> use case
  - data considerations, 88
  - data extraction, 92
  - description, 25
  - Markov chain clustering, 230
  - process warehousing, 254
  - time to event analysis, 222
 Economic efficiency of a model, 44  
 Edges, 51  
 Ego (sociogram), 281  
 Ego-centric measures (sociogram), 281  
 Elementary functions, 59  
 EM-algorithm, 202  
 Ergodic Markov chain, 226  
 Event-driven Process Chains (EPCs), 57, 121

- Event  
log, 99, 104, 105, 246  
sequence, 208  
set, 208  
view, 8, 12, 16, 39, 56, 78, 120, 129, 208, 210
- Explanatory variable, 59, 156, 162, 163, 173, 180, 195
- Exponential loss, 190
- Ex post analysis, 247
- eXtensible Event Stream (XES), 99, 257
- Extract-load-transform (ELT), 97
- Extract-transform-load (ETL), 90
- Facet (visualization), 130, 132
- Fact (OLAP), 100
- Feature extraction, 208
- Fitness (of process model), 270
- Fitness function, 261
- Flat structure, 99
- Frames, 49
- Frequency distribution, 68, 137
- Fruchterman Reingold layout, 282
- Generalization (of process model), 270
- Generalization error, 157
- Generalized linear models, 72
- Generic questions, 39
- Genetic miner, 256, 260
- Granularity level, 100
- Graph  
bipartite, 53, 56  
database, 94  
series-parallel, 53, 54
- Hamming distance, 193
- Hazard function, 221
- Heat map, 140
- HEP use case  
clustering, 198  
data anonymization, 89  
description, 28  
dynamic visualization, 132, 146  
process mining, 258  
variable description, 134
- Heuristic miner, 256, 258, 262
- Hidden Markov chain, 231
- Hierarchical method, 194
- Hierarchical structure, 99
- Histogram, 137
- HITS, 229
- Hubness, 229
- Hybrid structure, 99
- iMine, 21, 38, 119
- Impurity measure, 183
- In-degree, 280
- Independent random variables, 65
- Independent variables, 59
- Influential factors, 11
- Infographics, 151
- Integration  
format, 98  
strategy, 109
- Irreducible Markov chain, 226
- Item set, 233
- Jittering, 130, 137
- Joint distribution, 63, 138
- Kernel(s), 60  
function, 169  
trick, 60, 188
- Key performance indicator (KPI), 11, 41, 71, 78, 159, 255
- Key value store, 94
- Key word extraction algorithm (KEA), 309
- KKLayout, 282
- K-means, 199
- K-nearest neighbor, 220
- K-nearest neighbor classification, 185
- KPI, *See* Key performance indicator (KPI)
- Lasso, 164
- Latent Dirichlet allocation (LDA), 306
- Likelihood, 63
- Linear function, 60
- Linear regression, 159
- Linear temporal logic (LTL), 76
- Linkage, 195
- Linked data, 113
- Loading, 92
- Load shedding, 95
- Log format, 104
- Logistic regression, 72, 168, 180, 191
- Logistics use case  
change mining, 264  
description, 29  
time warping, 216
- Logit, 180
- Log structure, 99, 104
- Loop, 52

- Machine learning, 19, 204  
 Mapping (data schema), 100  
 Mapping (visualization), 128  
 MapReduce, 94  
 Margin, 186  
 Marginal distribution, 63  
 Market basket analysis, 211  
 Markov chain, 70, 225
  - aperiodic state, 226
  - connected state, 226
  - periodic state, 226
  - reachable states, 226
  - recurrent state, 226
 Markov property, 70  
 Maximum likelihood estimation, 68, 219, 227  
 Mean, 129, 138  
 Mean square error (MSE), 161  
 Median, 62, 137, 138  
 Medoid, 200  
 Meta-analysis, 313  
 Metadata, 81, 147, 294  
 Meta model, 43, 121, 122, 124  
 MHLAP, 101  
 Missing value, 80, 81, 138, 147, 184  
 Mixed models, 219  
 Model-based method, 193  
 Modeling
  - method, 39, 41, 42
  - task, 156
  - technique, 18, 43, 70
 Models
  - analogical models, 37
  - complexity, 157
  - of data, 158
  - elements, 39
  - idealized models, 37
  - language, 39
  - language semantics, 39
  - phenomenological models, 37
  - quality criteria, 44
  - structure, 40, 156
 MOLAP, 101  
 Monitoring, 113  
 Mosaic plot, 134  
 Motion chart, 133, 335  
 Multidimensional structure, 99  
 Multidimensional tables, 129  
 Multiple R-squared, 162  
 Multi-purpose automatic topic indexing (MAUI), 309  
 Mutation, 262  
 MXML, 257  
 Naive Bayes, 178  
 Neural nets, 159  
*n*-grams, 298  
 Nodes, 51  
 Nonparametric models, 159  
 Nonparametric regression, 159  
 Normal distribution, 66, 137  
 Null hypothesis, 69, 162  
 Objectivity of a model, 44  
 Observable variable, 67  
 Observational studies, 75  
 Odds, 62, 180  
 Offline analysis, 247  
 Online analysis, 246  
 Online Analytical Processing (OLAP), 101  
 Ontology, 109  
 Operational measurement, 76  
 Operational model, 35  
 Opinion mining, 276  
 Organizational perspective, 6, 38, 55  
 Out-degree, 280  
 Overfitting, 158  
 Page rank algorithm, 229  
 PAIS, 283  
 Parallel coordinates, 145  
 Partition around medoids (PAM), 200, 305  
 Partitioning method, 194  
 Part-of-speech tagging (POS), 308  
 Path, 52  
 Path, vertex-disjoint, 52  
 Pattern (local behavior of business process), 45  
 Petri nets, 56, 121  
 Phenomenon, 36  
 Pie chart, 134  
 Pivot table, 129
  - dimensions, 129
  - summary attributes, 129
 Polarity, 311  
 Population, 67  
 Posterior distribution, 228  
 Posterior probability, 65, 178  
 Post-processing, 259  
 Precision (of process model), 270  
 Predicate logic, 47  
 Predictive modeling, 156  
 Pre-eclampsia use case, description, 27  
 Pre-eclampsia use case, prediction, 170  
 Pre-eclampsia use case, response feature analysis, 218

- Pre-eclampsia use case, variable description, 144  
Pre-processing, 259  
Principal component, 142  
Prior distribution, 228  
Prior probabilities, 65, 177  
Probabilistic semantic index (PLSI), 306  
Probability density, 63  
Process  
    actor, 7  
    discovery, 255  
    instance, 6, 48, 54, 62, 67, 155  
    owner, 7  
    subject, 7  
Process-aware information system (PAIS), 247  
Process-Aware Information Systems, 283  
Process model  
    change time, 13  
    design time, 13, 121  
    run time, 13, 123  
Production perspective, 6, 38, 43, 44, 54, 67  
Profiling, 113  
Projection, 60, 142  
Proportional hazard model, 223  
Propositional logic, 46  
Publication bias, 313
- QQ Plot, 138, 163  
Qualitative analysis, 245  
Quality dimensions, 79, 147  
Quantile, 62  
Quantitative analysis, 245  
Quartile, 62, 137
- Radar plot, 149  
Radial basis kernel, 61, 188  
Random variable, 62  
Relevance of a model, 44  
Reliability (data quality), 79  
Reliability of a model, 44, 195  
Representational measurement, 76  
Residual, 160  
    analysis, 160  
Response feature analysis, 211  
Response variable, 59, 159, 166, 172, 173  
Ridge regression, 165  
ROC-curve, 175  
ROLAP, 101  
Role (organizational perspective), 7, 277  
Roll up, 100, 103
- Sample, 67  
Sample distribution, 68  
Sampling, 95  
Scale (visualization), 129  
Scatter plot, 140  
Scatter plot matrix, 141  
Schema  
    conflict, 109  
    integration, 108, 109  
    mapping, 109  
    matching, 109  
Scree plot, 196  
Secondary data, 74  
Self Organizing Map, 201  
Sensitivity (ROC-curve), 175  
Sequential data, 207  
Sequential OLAP, 96  
Silhouettes, 197  
Simulation, 248  
Sketching, 95  
Slice, 103  
Sliding window, 95  
Smoother, 159  
Snapshot, 90  
Snowflake schema, 102  
Social entity, 277  
Social network, 278  
Sociogram, 277  
Specificity (ROC-curve), 175  
Staging area, 90  
Standard deviation, 137  
Standard error, 68  
Star schema, 102  
State variables, 9  
State view, 9, 10, 12, 56, 127, 129, 170, 208, 210, 233  
Static process analysis, 248  
Stationary Markov chain, 70  
Statistical experiments, 76  
Statistical test, 69, 218  
Statistical units, 67  
Stochastic matrix, 71, 225  
Streaming data, 95  
Structure (of process model), 270  
Summary measure, 135  
Supervised learning, 12, 155  
Support constraint, 236  
Support vector, 187  
Surveys, 75  
Survival function, 221  
Swimlanes, BPMN, 124  
Synset, 308  
Syntactic constraint, 235

- Table structure, 99
- Task (business process), 54, 253, 256
- Temporal data, 10, 76, 104, 143, 170
  - transaction time, 80, 207
  - valid time, 80, 207
- Temporal database, 207
- Test error, 157
- Text mining, 99
- TF-IDF, 300
- Tilted time frame, 96
- Timeliness (data quality), 79
- Time sequences, 208
- Time stamp, 207
- Time-stamped data, 10, 207
- Tokenization, 298
- Training error, 157
- Transformation (data schema), 100
- Transformation, statistical, 129
- Transient state, 226
- Tree, 53
  - binary, 53
  - map, 135
- Triad, 277
- Type-token relation, 298
- Underfitting, 158
- Undirected graph, 278
- Unsupervised learning, 12, 193
- Use cases, domain semantics, 41
- Validation (of process model), 245
- Validity of a model, 44, 195
- Vapnik-Chervonenkis dimension, 187
- Variance, 137
- Vector
  - calculus, 59
  - quantization, 194
- Verification, 245
- Visual analytics, 119
- Waiting queue, 249
- Ward linkage, 195
- Warping path, 215
- Web service
  - choreography, 114
  - orchestration, 114
- Weibull distribution, 221
- Window, 241
- Workflow nets (WF nets), 57
- XML database, 94
- Zipf's law, 298