Chapter 1

Introduction

Due to modern information technology, which produces ever more powerful computers and faster networks every year, it is possible today to collect, transfer, combine, and store huge amounts of data at very low costs. Thus an ever-increasing number of companies and scientific and governmental institutions can afford to compile huge archives of tables, documents, images, and sounds in electronic form. The thought is compelling that if you only have enough data, you can solve any problem—at least in principle.

A closer examination reveals though, that data alone, however voluminous, are not sufficient. We may say that in large databases we cannot see the wood for the trees. Although any single bit of information can be retrieved and simple aggregations can be computed (for example, the average monthly sales in the Frankfurt area), general patterns, structures, and regularities usually go undetected. However, often these patterns are especially valuable, for example, because they can easily be exploited to increase turnover. For instance, if a supermarket discovers that certain products are frequently bought together, the number of items sold can sometimes be increased by appropriately arranging these products on the shelves of the market (they may, for example, be placed adjacent to each other in order to invite even more customers to buy them together, or they may be offered as a bundle).

However, to find these patterns and thus to exploit more of the information contained in the available data turns out to be fairly difficult. In contrast to the abundance of data there is a lack of tools to transform these data into useful knowledge. As John Naisbett remarked [Fayyad *et al.* 1996]:

> We are drowning in information, but starving for knowledge.

As a consequence a new area of research has emerged, which has been named *Knowledge Discovery in Databases (KDD)* or *Data Mining (DM)* and which has taken up the challenge to develop techniques that can help humans to discover useful patterns and regularities in their data.

Graphical Models: Representations for Learning, Reasoning and Data Mining, 2nd Edition
C. Borgelt, M. Steinbrecher and R. Kruse

In this introductory chapter we provide a brief overview on knowledge discovery in databases and data mining, which is intended to show the context of this book. In a first step, we try to capture the difference between "data" and "knowledge" in order to attain precise notions by which it can be made clear why it does not suffice just to gather data and why we must strive to turn them into knowledge. As an illustration we will discuss and interpret a well-known example from the history of science. Secondly, we explain the process of discovering knowledge in databases (the *KDD process*), of which data mining is just one, though very important, step. We characterize the standard data mining tasks and position the work of this book by pointing out for which tasks the discussed methods are well suited.

1.1 Data and Knowledge

In this book we distinguish between *data* and *knowledge*. Statements like "Columbus discovered America in 1492" or "Mrs Jones owns a VW Golf" are *data*. For these statements to qualify as data, we consider it to be irrelevant whether we already know them, whether we need these specific pieces of information at this moment, etc. For our discussion, the essential property of these statements is that they refer to single events, cases, objects, persons, etc., in general, to single instances. Therefore, even if they are true, their range of validity is very restricted and thus is their usefulness.

In contrast to the above, *knowledge* consists of statements like "All masses attract each other." or "Every day at 17:00 hours there runs an InterCity (a specific type of train of German Rail) from Magdeburg to Braunschweig." Again we neglect the relevance of the statement for our current situation and whether we already know it. The essential property is that these statements do not refer to single instances, but are general laws or rules. Therefore, provided they are true, they have a wide range of validity, and, above all else, they allow us to make predictions and thus they are very useful.

It has to be admitted, though, that in daily life statements like "Columbus discovered America in 1492." are also called knowledge. However, we disregard this way of using the term "knowledge", regretting that full consistency of terminology with daily life language cannot be achieved. Collections of statements about single instances do not qualify as knowledge.

Summarizing, data and knowledge can be characterized as follows:

Data

- refer to single instances
  (single objects, persons, events, points in time, etc.)
- describe individual properties
- are often available in huge amounts
  (databases, archives)

- are usually easy to collect or to obtain
  (for example cash registers with scanners in supermarkets, Internet)
- do not allow us to make predictions

Knowledge

- refers to *classes* of instances
  (*sets* of objects, persons, events, points in time, etc.)
- describes general patterns, structures, laws, principles, etc.
- consists of as few statements as possible
  (this is an objective, see below)
- is usually hard to find or to obtain
  (for example natural laws, education)
- allows us to make predictions

From these characterizations we can clearly see that usually knowledge is much more valuable than (raw) data. It is mainly the generality of the statements and the possibility to make predictions about the behavior and the properties of new cases that constitute its superiority.

However, not just any kind of knowledge is as valuable as any other. Not all general statements are equally important, equally substantial, equally useful. Therefore knowledge must be evaluated and assessed. The following list, which we do not claim to be complete, names some important criteria:

Criteria to Assess Knowledge

- correctness (probability, success in tests)
- generality (range of validity, conditions for validity)
- usefulness (relevance, predictive power)
- comprehensibility (simplicity, clarity, parsimony)
- novelty (previously unknown, unexpected)

In science correctness, generality, and simplicity (parsimony) are at the focus of attention: One way to characterize science is to say that it is the search for a minimal correct description of the world. In business and industry greater emphasis is placed on usefulness, comprehensibility, and novelty: the main goal is to get a competitive edge and thus to achieve higher profit. Nevertheless, none of the two areas can afford to neglect the other criteria.

Tycho Brahe and Johannes Kepler

Tycho Brahe (1546–1601) was a Danish nobleman and astronomer, who in 1576 and in 1584, with the financial support of Frederic II, King of Denmark and Norway, built two observatories on the island of Sen, about 32 km to the north-east of Copenhagen. Using the best equipment of his time (telescopes were unavailable then—they were used only later by Galileo Galilei (1564–1642) and Johannes Kepler (see below) for celestial observations) he determined the positions of the sun, the moon, and the planets with a precision of less than one minute of arc, thus surpassing by far the exactitude of all measurements carried out earlier. He achieved in practice the theoretical limit for observations with the unaided eye. Carefully he recorded the motions of the celestial bodies over several years [Greiner 1989, Zey 1997].

Tycho Brahe gathered data about our planetary system. Huge amounts of data—at least from a 16th century point of view. However, he could not discern the underlying structure. He could not combine his data into a consistent scheme—to some extent, because be adhered to the geocentric system. He could tell exactly in what position Mars had been on a specific day in 1585, but he could not relate the positions on different days in such a way as to fit his highly accurate observational data. All his hypotheses were fruitless. He developed the so-called Tychonic planetary model, according to which the sun and the moon revolve around the earth, but all other planets revolve around the sun, but this model, though popular in the 17th century, did not stand the test of time. Today we may say that Tycho Brahe had a "data mining" or "knowledge discovery" problem. He had the necessary data, but he could not extract the knowledge contained in it.

Johannes Kepler (1571–1630) was a German astronomer and mathematician and assistant to Tycho Brahe. He advocated the Copernican planetary model, and during his whole life he endeavored to find the laws that govern the motions of the celestial bodies. He strove to find a mathematical description, which, in his time, was a virtually radical approach. His starting point were the catalogs of data Tycho Brahe had compiled and which he continued in later years. After several unsuccessful trials and long and tedious calculations, Johannes Kepler finally managed to condense Tycho Brahe's data into three simple laws, which have been named after him. Having discovered in 1604 that the course of Mars is an ellipse, he published the first two laws in "Astronomia Nova" in 1609, the third ten years later in his principal work "Harmonica Mundi" [Feynman *et al.* 1963, Greiner 1989, Zey 1997].

1. Each planet moves around the sun on an elliptical course, with the sun at one focus of the ellipse.
2. The radius vector from the sun to the planet sweeps out equal areas in equal intervals of time.
3. The squares of the periods of any two planets are proportional to the cubes of the semi-major axes of their respective orbits: $T \sim a^{\frac{3}{2}}$.

Tycho Brahe had collected a large amount of celestial data, Johannes Kepler found the laws by which they can be explained. He discovered the hidden knowledge and thus became one of the most famous "data miners" in history.

Today the works of Tycho Brahe are almost forgotten. His catalogs are merely of historical value. No textbook on astronomy contains extracts from his measurements. His observations and minute recordings are raw data and thus suffer from a decisive disadvantage: They do not provide us with any insight into the underlying mechanisms and therefore they do not allow us to make predictions. Kepler's laws, however, are treated in all textbooks on astronomy and physics, because they state the principles that govern the motions of planets as well as comets. They combine all of Brahe's measurements into three fairly simple statements. In addition, they allow us to make predictions: If we know the position and the velocity of a planet at a given moment, we can compute, using Kepler's laws, its future course.

1.2 Knowledge Discovery and Data Mining

How did Johannes Kepler discover his laws? How did he manage to extract from Tycho Brahe's long tables and voluminous catalogs those simple laws that revolutionized astronomy? We know only fairly little about this. He must have tested a large number of hypotheses, most of them failing. He must have carried out long and complicated computations. Presumably, outstanding mathematical talent, tenacious work, and a considerable amount of good luck finally led to success. We may safely guess that he did not know any universal method to discover physical or astronomical laws.

Today we still do not know such a method. It is still much simpler to gather data, by which we are virtually swamped in today's "information society" (whatever that means), than to obtain knowledge. We even need not work diligently and perseveringly any more, as Tycho Brahe did, in order to collect data. Automatic measurement devices, scanners, digital cameras, and computers have taken this load from us. Modern database technology enables us to store an ever-increasing amount of data. It is indeed as John Naisbett remarked: We are drowning in information, but starving for knowledge.

If it took such a distinguished mind like Johannes Kepler several years to evaluate the data gathered by Tycho Brahe, which today seem to be negligibly few and from which he even selected only the data on the course of Mars, how can we hope to cope with the huge amounts of data available today? "Manual" analysis has long ceased to be feasible. Simple aids like, for example, representations of data in charts and diagrams soon reach their limits. If we refuse to simply surrender to the flood of data, we are forced to look for intelligent computerized methods by which data analysis can be automated at least partially. These are the methods that are sought for in the research areas called *Knowledge Discovery in Databases (KDD)* and *Data Mining (DM)*. It is true, these methods are still very far from replacing people like Johannes Kepler, but it is not entirely implausible that he, if supported by these methods, would have reached his goal a little sooner.

Often the terms *Knowledge Discovery* and *Data Mining* are used interchangeably. However, we distinguish them here. By *Knowledge Discovery in Databases (KDD)* we mean a process consisting of several steps, which is usually characterized as follows [Fayyad *et al.* 1996]:

> Knowledge discovery in databases is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.

One step of this process, though definitely one of the most important, is *Data Mining*. In this step modeling and discovery techniques are applied.

1.2.1 The KDD Process

In this section we structure the KDD process into two preliminary and five main steps or phases. However, the structure we discuss here is by no means binding: it has proven difficult to find a single scheme that everyone in the scientific community can agree on. However, an influential suggestion and detailed exposition of the KDD process, which is close to the scheme presented here and which has had considerable impact, because it is backed by several large companies like NCR and DaimlerChrysler, is the CRISP-DM model (CRoss Industry Standard Process for Data Mining) [Chapman *et al.* 1999].

Preliminary Steps

- estimation of potential benefit
- definition of goals, feasibility study

Main Steps

- check data availability, data selection, if necessary, data collection
- preprocessing (usually 60–90% of total overhead)
  - unification and transformation of data formats
  - data cleaning
    (error correction, outlier detection, imputation of missing values)
  - reduction / focusing
    (sample drawing, feature selection, prototype generation)
- **Data Mining** (using a variety of methods)
- visualization
  (also in parallel to preprocessing, data mining, and interpretation)
- interpretation, evaluation, and test of results
- deployment and documentation

The preliminary steps mainly serve the purpose to decide whether the main steps should be carried out. Only if the potential benefit is high enough and the demands can be met by data mining methods, can it be expected that some profit results from the usually expensive main steps.

In the main steps the data to be analyzed for hidden knowledge are first collected (if necessary), appropriate subsets are selected, and they are transformed into a unique format that is suitable for applying data mining techniques. Then they are cleaned and reduced to improve the performance of the algorithms to be applied later. These preprocessing steps usually consume the greater part of the total costs. Depending on the data mining task that was identified in the goal definition step (see below for a list), data mining methods are applied (see farther below for a list), the results of which, in order to interpret and evaluate them, can be visualized. Since the desired goal is rarely achieved in the first go, usually several steps of the preprocessing phase (for example feature selection) and the application of data mining methods have to be reiterated in order to improve the result. If it has not been obvious before, it is clear now that KDD is an interactive process, rather than completely automated. A user has to evaluate the results, check them for plausibility, and test them against hold-out data. If necessary, he/she modifies the course of the process to make it meet his/her requirements.

1.2.2 Data Mining Tasks

In the course of time typical tasks have been identified, which data mining methods should be able to solve (although, of course, not every single method is required to be able to solve all of them—it is the combination of methods that makes them powerful). Among these are especially those named in the—surely incomplete—list below. We tried to characterize them not only by their name, but also by a typical question [Nakhaeizadeh 1998b].

- classification
  Is this customer credit-worthy?
- segmentation, clustering
  What groups of customers do I have?
- concept description
  Which properties characterize fault-prone vehicles?
- prediction, trend analysis
  What will the exchange rate of the dollar be tomorrow?
- dependence/association analysis
  Which products are frequently bought together?
- deviation analysis
  Are there seasonal or regional variations in turnover?

Classification and prediction are by far the most frequent tasks, since their solution can have a direct effect, for instance, on the turnover and the profit of a company. Dependence and association analysis come next, because they can be used, for example, to do shopping basket analysis, that is, to discover which products are frequently bought together, and are therefore also of considerable commercial interest. Clustering and segmentation are also not infrequent.

1.2.3 Data Mining Methods

Research in data mining is highly interdisciplinary. Methods to tackle the tasks listed in the preceding section have been developed in a large variety of research areas including—to name only the most important—statistics, artificial intelligence, machine learning, and soft computing. As a consequence there is an arsenal of methods, based on a wide range of ideas, and thus there is no longer such a lack of tools. To give an overview, we list some of the more prominent data mining methods. Each list entry refers to a few publications on the method and points out for which data mining tasks the method is especially suited. Of course, this list is far from being complete. The references are necessarily incomplete and may not always be the best ones possible, since we are clearly not experts for all of these methods and since, obviously, we cannot name everyone who has contributed to the one or the other.

- classical statistics (discriminant analysis, time series analysis, etc.)
[Larsen and Marx 2005, Everitt 2006, Witte and Witte 2006]
[Freedman *et al.* 2007]
classification, prediction, trend analysis

- decision/classification and regression trees
[Breiman *et al.* 1984, Quinlan 1993, Rokach and Maimon 2008]
classification, prediction

- naive Bayes classifiers
[Good 1965, Duda and Hart 1973, Domingos and Pazzani 1997]
classification, prediction

- probabilistic networks (Bayesian networks/Markov networks)
[Lauritzen and Spiegelhalter 1988, Pearl 1988, Jensen and Nielsen 2007]
classification, dependence analysis

- artificial neural networks
[Anderson 1995, Bishop 1996, Rojas 1996, Haykin 2008]
classification, prediction, clustering (Kohonen feature maps)

- support vector machines and kernel methods
[Cristianini and Shawe-Taylor 2000, Schölkopf and Smola 2001]
[Shawe-Taylor and Cristianini 2004, Abe 2005]
classification, prediction

- k-nearest neighbor/case-based reasoning
  [Kolodner 1993, Shakhnarovich *et al.* 2006, Hüllermeier 2007]
  classification, prediction
- inductive logic programming
  [Muggleton 1992, Bergadano and Gunetti 1995, de Raedt *et al.* 2007]
  classification, association analysis, concept description
- association rules
  [Agrawal and Srikant 1994, Agrawal *et al.* 1996, Zhang and Zhang 2002]
  association analysis
- hierarchical and probabilistic cluster analysis
  [Bock 1974, Everitt 1981, Cheeseman *et al.* 1988, Xu and Wunsch 2008]
  segmentation, clustering
- fuzzy cluster analysis
  [Bezdek *et al.* 1999, Höppner *et al.* 1999, Miyamoto *et al.* 2008]
  segmentation, clustering
- neuro-fuzzy rule induction
  [Wang and Mendel 1992, Nauck and Kruse 1997, Nauck *et al.* 1997]
  classification, prediction
- and many more

Although for each data mining task there are several reliable methods to solve it, there is, as already indicated above, no single method that can solve all tasks. Most methods are tailored to solve a specific task and each of them exhibits different strengths and weaknesses. In addition, usually several methods must be combined in order to achieve good results. Therefore commercial data mining products like, for instance, Clementine (SPSS Inc., Chicago, IL, USA), SAS Enterprise Miner (SAS Institute Inc., Cary, NC, USA), DB2 Intelligent Miner (IBM Inc., Armonk, NY, USA), or free platforms like KNIME (Konstanz Information Miner, http://www.knime.org/) offer several of the above methods under an easy to use graphical interface. However, as far as we know there is still no tool that contains all of the methods mentioned above.

A compilation of a large number of data mining suites and individual programs for specific data mining tasks can be found at:

`http://www.kdnuggets.com/software/index.html`

Generally, the KDnuggets web site at

`http://www.kdnuggets.com/`

is a valuable source of information for basically all topics related to data mining and knowledge discovery in databases. Another web site well worth visiting for information about data mining and knowledge discovery is:

`http://www.dmoz.org/Computers/Software/Databases/Data_Mining/`

1.3 Graphical Models

This book deals with two data mining tasks, namely *dependence analysis* and *classification*. These tasks are, of course, closely related, since classification can be seen as a special case of dependence analysis: it concentrates on specific dependences, namely on those between a distinguished attribute—the class attribute—and other, descriptive attributes. It then tries to exploit these dependences to classify new cases. Within the set of methods that can be used to solve these tasks, we focus on techniques to induce *graphical models* or, as we will also call them, *inference networks* from data.

The ideas of graphical models can be traced back to three origins (according to [Lauritzen 1996]), namely statistical mechanics [Gibbs 1902], genetics [Wright 1921], and the analysis of contingency tables [Bartlett 1935]. Originally, they were developed as means to build models of a domain of interest. The rationale underlying such models is that, since high-dimensional domains tend to be unmanageable as a whole (and the more so if imprecision and uncertainty are involved), it is necessary to *decompose* the available information. In *graphical modeling* [Whittaker 1990, Kruse *et al.* 1991, Lauritzen 1996] such a decomposition exploits (conditional) dependence and independence relations between the attributes used to describe the domain under consideration. The structure of these relations is represented as a network or graph (hence the names graphical model and inference network), often called a *conditional independence graph*. In such a graph each node stands for an attribute and each edge for a direct dependence between two attributes.

However, such a conditional independence graph turns out to be not only a convenient way to represent the content of a model. It can also be used to facilitate reasoning in high-dimensional domains, since it allows us to draw inferences by computations in lower-dimensional subspaces. Propagating evidence about the values of observed attributes to unobserved ones can be implemented by locally communicating node processors and therefore can be made very efficient. As a consequence, graphical models were quickly adopted for use in expert and decision support systems [Neapolitan 1990, Kruse *et al.* 1991, Cowell 1992, Castillo *et al.* 1997, Jensen 2001]. In such a context, that is, if graphical models are used to draw inferences, we prefer to call them *inference networks* in order to emphasize this objective.

Using inference networks to facilitate reasoning in high-dimensional domains has originated in the probabilistic setting. *Bayesian networks* [Pearl 1986, Pearl 1988, Jensen 1996, Jensen 2001, Gamez *et al.* 2004, Jensen and Nielsen 2007], which are based on directed conditional independence graphs, and *Markov networks* [Isham 1981, Lauritzen and Spiegelhalter 1988, Pearl 1988, Lauritzen 1996, Wainwright and Jordan 2008], which are based on undirected graphs, are the most prominent examples. Early efficient implementations include HUGIN [Andersen *et al.* 1989] and PATHFINDER [Heckerman 1991], and early applications include the interpretation of electromyographic findings (MUNIN) [Andreassen *et al.* 1987], blood group determination of Danish Jersey cattle for parentage verification (BOBLO) [Rasmussen 1992], and troubleshooting non-functioning devices like printers and photocopiers [Heckerman *et al.* 1994]. Nowadays, successful applications of graphical models, in particular in the form of Bayesian network classifiers, can be found in an abundance of areas, including, for example, domains as diverse as manufacturing [Agosta 2004], finance (risk assessment) [Neil *et al.* 2005], steel production [Pernkopf 2004], telecommunication network diagnosis [Khanafar *et al.* 2008], handwriting recognition [Cho and Kim 2003], object recognition in images [Schneiderman 2004], articulatory feature recognition [Frankel *et al.* 2007], gene expression analysis [Kim *et al.* 2004], protein structure identification [Robles et al 2004], and pneumonia diagnosis [Charitos *et al.* 2007].

However, fairly early on graphical modeling was also generalized to be usable with uncertainty calculi other than probability theory [Shafer and Shenoy 1988, Shenoy 1992b, Shenoy 1993], for instance in the so-called valuation-based networks [Shenoy 1992a], and was implemented, for example, in PULCINELLA [Saffiotti and Umkehrer 1991]. Due to their connection to fuzzy systems, which in the past have successfully been applied to solve control problems and to represent imperfect knowledge, possibilistic networks gained attention too. They can be based on the context model interpretation of a degree of possibility, which focuses on imprecision [Gebhardt and Kruse 1993a, Gebhardt and Kruse 1993b], and were implemented, for example, in POSSINFER [Gebhardt and Kruse 1996a, Kruse *et al.* 1994].

Initially the standard approach to construct a graphical model was to let a human domain expert specify the dependences in the domain under consideration. This provided the network structure. Then the human domain expert had to estimate the necessary conditional or marginal distribution functions that represent the quantitative information about the domain. This approach, however, can be tedious and time consuming, especially if the domain under consideration is large. In some situations it may even be impossible to carry out, because no, or only vague, expert knowledge is available about the (conditional) dependence and independence relations that hold in the considered domain, or the needed distribution functions cannot be estimated reliably.

As a consequence, learning graphical models from databases of sample cases became a main focus of attention in the 1990s (cf., for example, [Herskovits and Cooper 1990, Cooper and Herskovits 1992, Singh and Valtorta 1993, Buntine 1994, Heckerman *et al.* 1995, Cheng *et al.* 1997, Jordan 1998] for learning probabilistic networks and [Gebhardt and Kruse 1995, Gebhardt and Kruse 1996b, Gebhardt and Kruse 1996c, Borgelt and Kruse 1997a, Borgelt and Kruse 1997b, Borgelt and Gebhardt 1997] for learning possibilistic networks), and thus graphical models entered the realm of data mining methods. Due to its considerable success, this research direction continued to attract a lot of interest after the turn of the century (cf., for instance, [Steck 2001, Chickering 2002, Cheng *et al.* 2002, Neapolitan 2004, Grossman and Domingos

2004, Taskar *et al.* 2004, Roos *et al.* 2005, Niculescu *et al.* 2006, Tsamardinos *et al.* 2006, Jakulin and Rish 2006, Castillo 2008]).

This success does not come as a surprise: graphical models have several advantages when applied to knowledge discovery and data mining problems. In the first place, as already pointed out, the network representation provides a comprehensible qualitative (network structure) and quantitative description (associated distribution functions) of the domain under consideration, so that the learning result can be checked for plausibility against the intuition of human experts. Secondly, learning algorithms for inference networks can fairly easily be extended to incorporate the background knowledge of human experts. In the simplest case a human domain expert specifies the dependence structure of the domain to be modeled and automatic learning is used only to determine the distribution functions from a database of sample cases. More sophisticated approaches take a prior model of the domain and modify it (add or remove edges, change the distribution functions) w.r.t. the evidence provided by a database of sample cases [Heckerman *et al.* 1995]. Finally, although fairly early on the learning task was shown to be NP-complete in the general case [Chickering *et al.* 1994, Chickering 1995], there are several good heuristic approaches that have proven to be successful in practice and that lead to very efficient learning algorithms.

In addition to these practical advantages, graphical models provide a framework for some of the data mining methods named above: Naive Bayes classifiers are probabilistic networks with a special, star-like structure (cf. Chapter 6). Decision trees can be seen as a special type of probabilistic network in which there is only one child attribute and the emphasis is on learning the local structure of the network (cf. Chapter 8). Furthermore there are some interesting connections to fuzzy clustering [Borgelt *et al.* 2001] and neuro-fuzzy rule induction [Nürnberger *et al.* 1999] through naive Bayes classifiers, which may lead to powerful hybrid systems.

1.4 Outline of this Book

This book covers three types of graphical models: relational, probabilistic, and possibilistic networks. Relational networks are mainly discussed to provide more comprehensible analogies, but also to connect graphical models to database theory. The main focus, however, is on probabilistic and possibilistic networks. In the following we give a brief outline of the chapters.

In Chapter 2 we review very briefly relational and probabilistic reasoning (in order to provide all fundamental notions) and then concentrate on possibility theory, for which we provide a detailed semantical introduction based on the *context model*. In this chapter we clarify and at some points modify the context model interpretation of a degree of possibility where we found its foundations to be weak or not spelt out clearly enough.

In Chapter 3 we study how relations as well as probability and possibility distributions, under certain conditions, can be decomposed into distributions on lower-dimensional subspaces. By starting from the simple case of relational networks, which, sadly, are usually neglected entirely in introductions to graphical modeling, we try to make the theory of graphical models and reasoning in graphical models more easily accessible. In addition, by developing a peculiar formalization of relational networks a very strong formal similarity can be achieved to possibilistic networks. In this way possibilistic networks can be introduced as simple ‘‘fuzzyfications’’ of relational networks.

In Chapter 4 we explain the connection of decompositions of distributions to graphs, as it is brought about by the notion of *conditional independence*. In addition we briefly review two of the best-known propagation algorithms for inference networks. However, although we provide a derivation of the evidence propagation formula for undirected trees and a brief review of join tree propagation, this chapter does not contain a full exposition of evidence propagation. This topic has been covered extensively in other books, and thus we only focus on those components that we need for later chapters.

With Chapter 5 we turn to learning graphical models from data. We study a fundamental learning operation, namely how to estimate projections (that is, marginal distributions) from a database of sample cases. Although trivial for the relational and the probabilistic case, this operation is a severe problem in the possibilistic case (not formally, but in terms of efficiency). Therefore we explain and formally justify an efficient method for computing maximum projections of database-induced possibility distributions.

In Chapter 6 we study naive Bayes classifiers and derive a naive possibilistic classifier in direct analogy to a naive Bayes classifier.

In Chapter 7 we proceed to *qualitative* or *structural learning*. That is, we study how to induce a graph structure from a database of sample cases. Following an introduction to the principles of global structure learning, which is intended to provide an intuitive background (like the greater part of Chapter 3), we discuss several evaluation measures (or scoring functions) for learning relational, probabilistic, and possibilistic networks. By working out the underlying principles as clearly as possible, we try to convey a deep understanding of these measures and strive to reveal the connections between them. Furthermore, we review several search methods, which are the second core ingredient of a learning algorithm for graphical models: they specify which graph structures are explored in order to find the most suitable one.

In Chapter 8 we extend qualitative network induction to learning local structure. We explain the connection to decision trees and decision graphs and suggest study approaches to local structure learning for Bayesian networks.

In Chapter 9 we study the causal interpretation of learned Bayesian networks and in particular the so-called *inductive causation algorithm*, which is claimed to be able to uncover, at least partially, the causal dependence structure underlying a domain of interest. We carefully study the assumptions underlying this approach and reach the conclusion that such strong claims cannot be justified, although the algorithm is a useful heuristic method.

In Chapter 10 visualization methods for probability functions are studied. In particular, we discuss a visualization approach that draws on the formal similarity of conditional probability distributions to association rules.

In Chapter 11 we show how graphical models can be used to derive a diagnostic procedure for (analog) electrical circuits that is able to detect so-called *soft faults*. In addition, we report about some successful applications of graphical models in the telecommunications and automotive industry.

Software and additional material that is related to the contents of this book can be found at the following URL:

`http://www.borgelt.net/books/gm/`