Data Mining with Graphical Models

Rudolf Kruse and Christian Borgelt

Department of Knowledge Processing and Language Engineering
Otto-von-Guericke-University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
E-mail: {kruse,borgelt}@iws.cs.uni-magdeburg.de

Abstract. *Data Mining*, or *Knowledge Discovery in Databases*, is a fairly young research area that has emerged as a reply to the flood of data we are faced with nowadays. It tries to meet the challenge to develop methods that can help human beings to discover useful patterns in their data. One of these techniques — and definitely one of the most important, because it can be used for such frequent data mining tasks like classifier construction and dependence analysis — is learning *graphical models* from datasets of sample cases. In this paper we review the ideas underlying graphical models, with a special emphasis on the less well known possibilistic networks. We discuss the main principles of learning graphical models from data and consider briefly some algorithms that have been proposed for this task as well as data preprocessing methods and evaluation measures.

1 Introduction

Today every company stores and processes its data electronically, in production, marketing, stock-keeping or personnel management. The data processing systems used were developed, because it is very important for a company to be able to retrieve certain pieces of information, like the address of a customer, in a fast and reliable way. Today, however, with ever increasing computer power and due to advances in database and software technology, we may think about using electronically stored data not only to retrieve specific information, but also to search for hidden patterns and regularities. If, for example, by analyzing customer receipts a supermarket chain finds out that certain products are frequently bought together, turnover may be increased by placing the products on the shelves of the supermarkets accordingly.

Unfortunately, in order to discover such knowledge in databases the retrieval capacities of normal database systems as well as the methods of classical data analysis are often insufficient. With them, we may retrieve arbitrary individual information, compute simple aggregations, or test the hypothesis whether the day of the week has an influence on the product quality. But more general patterns, structures, or regularities go undetected. These patterns, however, are often highly valuable and may be exploited, for instance, to increase sales. As a consequence a new research area has emerged in recent years—often called

Knowledge Discovery in Databases (KDD) or *Data Mining* (DM)—in which hypotheses and models describing the regularities in a given dataset are generated and tested automatically. The hypotheses and models found in this way can then be used to gain insight into the domain under consideration, to predict its future development, and to support decision making.

In this paper we consider two of the most important data mining tasks, namely the construction of classifiers and the analysis of dependences. Among the different methods for these tasks we concentrate on learning a graphical model from a dataset of sample cases. Furthermore, our emphasis is on possibilistic graphical models, which are a powerful tool for the analysis of imprecise data.

2 Graphical Models

An object or a case of a given domain of interest is usually described by a set of attributes. For instance, to describe a car we may use the manufacturer, the model name, the color etc. Depending on the specific object or case under consideration these attributes have certain values, for example, Volkswagen, Golf, red etc. Sometimes only certain combinations of attribute values are possible, for example, because certain special equipment items may not be chosen simultaneously, or certain combinations of attribute values are more frequent than others, for example, red VW Golf are more frequent than yellow BMW Z1. Such possibility or frequency information can be represented as a distribution on the Cartesian product of the attribute domains. That is, to each combination of attribute values we assign its possibility or probability.

Often a very large number of attributes is necessary to describe a given domain of interest appropriately. Since the number of possible value combinations grows exponentially with the number of attributes, it is often impossible to represent this distribution directly, for example, in order to draw inferences. One way to cope with this problem is to construct a graphical model. Graphical models are based on the idea that independences between attributes can be exploited to decompose a high-dimensional distributions into a set of (conditional or marginal) distributions on low-dimensional subspaces. This decomposition (as well as the independences that make it possible) is encoded by a graph: Each node represents an attribute. Edges connect nodes that are directly dependent on each other. In addition, the edges specify the paths on which evidence has to be propagated if inferences are to be drawn.

Since graphical models have been developed first in probability theory and statistics, the best-known approaches originated from this area, namely Bayes networks [Pearl 1988] and Markov networks [Lauritzen and Spiegelhalter 1988]. However, the underlying decomposition principle has been generalized, resulting in the so-called valuation-based networks [Shenoy 1992], and has been transferred to possibility theory [Gebhardt and Kruse 1996]. All of these approaches lead to efficient implementations, for example, HUGIN [Andersen *et al.* 1989], PULCINELLA [Saffiotti and Umkehrer 1991], PATHFINDER [Heckerman 1991], and POSSINFER [Gebhardt and Kruse 1996].

2.1 Decomposition

The notion of *decomposition* is probably best-known from relational database theory. Thus it comes as no surprise that relational database theory is closely connected to the theory of graphical models. This connection is based on the notion of a relation being *join-decomposable*, which is used in relational database systems to decompose high-dimensional relations and thus to store them with less redundancy and (of course) using less storage space.

Join-decomposability means that a relation can be reconstructed from certain *projections* by forming the so-called *natural join* of these projections. Formally, this can be described as follows: Let $U = \{A_1, \ldots, A_n\}$ be a set of attributes with respective domains $\mathrm{dom}(A_i)$. Furthermore let r_U be a relation over U. Such a relation can be described by its *indicator function*, which assigns a value of 1 to all tuples that are contained in the relation and a value of 0 to all other tuples. The tuples themselves are represented as conjunctions $\bigwedge_{A_i \in U} A_i = a_i$, which state a value for each attribute. Then the *projection* onto a subset $M \subseteq U$ of the attributes can then be defined as the relation

$$r_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big) = \max_{\substack{\forall A_j \in U-M: \\ a_i \in \mathrm{dom}(A_j)}} r_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big),$$

where the somewhat sloppy notation under the maximum operator is meant to express that the maximum has to be taken over all values of all attributes in the set $U - M$. With this notation a relation r_U is called *join-decomposable* w.r.t. a family $\mathcal{M} = \{M_1, \ldots, M_m\}$ of subsets of U if and only if

$$\begin{aligned}&\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) : \\ &\quad r_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big) = \min_{M \in \mathcal{M}} r_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big).\end{aligned}$$

Note that the minimum of the projections is equivalent to the natural join of relational calculus, justifying the usage of the term "join-decomposable".

This decomposition scheme can easily be transferred to the probabilistic case: All we have to do is to replace the projection operation and the natural join by their probabilistic counterparts. Thus we arrive at the decomposition formula

$$\begin{aligned}&\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) : \\ &\quad p_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big) = \prod_{M \in \mathcal{M}} \phi_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big).\end{aligned}$$

The functions ϕ_M can be computed from the marginal distributions on the attribute sets M. This demonstrates that the computation of a marginal distribution takes the place of the projection operation. These functions are called *factor potentials* [Castillo *et al.* 1997]. Alternatively, one may describe a decomposition of a probability distribution by exploiting the (generalized) product rule of probability theory and by using conditional distributions.

The possibilistic case is even closer to the relational one, because the decomposition formula is virtually identical. The only difference is that the relations r are replaced by possibility distributions π, i.e., by functions which are not restricted to the values 0 and 1 (like indicator functions), but may take arbitrary values from the interval $[0, 1]$. In this way a "gradual possibility" is modeled with a generalized indicator function. As a consequence possibilistic graphical models may be seen as "fuzzifications"' of relational graphical models.

Of course, if such degrees of possibility are introduced, the question of their interpretation arises, because possibility is an inherently two-valued concept. In our research we rely on the *context model* [Gebhardt and Kruse 1993] to answer this question. However, since the common ways of justifying the maximum and minimum operations are not convincing, we have developed a different justification that is based on the goal of reasoning with graphical models. Details about this justification can be found in [Borgelt and Kruse 2002].

2.2 Graphical Representation

Decompositions can very conveniently be represented by graphs. In the first place, graphs can be used to specify the sets M of attributes underlying the decomposition. How this is done depends on whether the graph is directed or undirected. If it is undirected, the sets M are the maximal cliques of the graph, where a clique is a complete subgraph, which is called maximal if it is not a proper part of another complete subgraph. If the graph is directed, we can be more explicit about the distributions of the decomposition: We can employ conditional distributions, because the direction of the edges allows us to distinguish between conditioned and conditioning attributes. However, in the relational and the possibilistic case no changes result from this, since the conditional distributions are identical to their unconditional analogs (because in these calculi no renormalization is carried out).

Secondly, graphs can be used to represent (conditional) dependences and independences via the notion of *node separation*. What is to be understood by "separation" again depends on whether the graph is directed or undirected. If it is undirected, node separation is defined as follows: If X, Y, and Z are three disjoint sets of nodes, then Z separates X and Y if all paths from a node in X to a node in Y contain a node in Z.

For directed acyclic graphs node separation is defined as follows [Pearl 1988]: If X, Y, and Z are three disjoint sets of nodes, then Z separates X and Y if there is no path (disregarding the directionality of the edges) from a node in X to a node in Y along which the following two conditions hold:

1. Every node, at which the edges of the path converge, either is in Z or has a descendant in Z, and
2. every other node is not in Z.

With the help of these separation criteria we can define *conditional independence graphs*: A graph is a conditional independence graph w.r.t. a given (multidimensional) distribution if it captures by node separation only valid conditional independences. Conditional independence means (for three attributes A, B, and C with A being independent of C given B; the generalization is obvious), that

$$P(A = a, B = b, C = c) = P(A = a \mid B = b) \cdot P(C = c \mid B = b)$$

in the probabilistic case and

$$\pi(A = a, B = b, C = c) = \min\{\pi(A = a \mid B = b), \pi(C = c \mid B = b)\}$$

in the possibilistic and the relational case.

These formula also indicate that conditional independence and decomposability are closely connected. Formally, this connection is established by theorems, which state that a distribution is decomposable w.r.t. a given graph if the graph is a conditional independence graph. In the probabilistic case such a theorem is usually attributed to [Hammersley and Clifford 1971]. In the possibilistic case an analogous theorem can be proven, although some restrictions have to be introduced on the graphs [Gebhardt 1997, Borgelt and Kruse 2002].

Finally, the graph underlying a graphical model is very useful to derive evidence propagation algorithms, because transmitting evidence information can be implemented by node processors that communicate by sending message to each other along the edges of the graph. Details about these methods can be found, for instance, in [Castillo *et al.* 1997].

3 Learning Graphical Models from Data

Since a graphical model represents the dependences and independences that hold in a given domain of interest in a very clear way and allows for efficient reasoning, it is a very powerful tool—once it is constructed. However, its construction by human experts can be tedious and time-consuming. As a consequence recent research in graphical models has placed a strong emphasis on learning graphical models from a dataset of sample cases. Although it has been shown that this learning task is NP-hard in general [Chickering *et al.* 1994], some very successful heuristic algorithms have been developed [Cooper and Herskovits 1992, Heckerman *et al.* 1995, Gebhardt and Kruse 1995].

However, some of these approaches, especially probabilistic ones, are restricted to learning from *precise* data. That is, the description of the sample cases must contain neither missing values nor set-valued information. There must be exactly one value for each attribute in each of the sample cases. Unfortunately, this prerequisite is rarely met in applications: Real-world databases are often incomplete and useful imprecise information (sets of values for an attribute) is frequently available (even though it is often neglected, because common database systems cannot handle it adequately). Therefore we face the challenge to extend the existing learning algorithms to incomplete and imprecise data.

Research in probabilistic graphical models tries to meet this challenge mainly with the expectation maximization (EM) algorithm [Dempster *et al.* 1977, Bauer *et al.* 1997]. In our own research, however, we focus on possibilistic graphical models, because possibility theory [Dubois and Prade 1988] allows for a very convenient treatment of missing values and imprecise data. For possibilistic networks no iterative procedure like the EM algorithm is necessary, so that considerable gains in efficiency can result [Borgelt and Kruse 2002].

3.1 Learning Principles

There are basically three approaches to learn a graphical model from data:

- Test whether a given distribution is decomposable w.r.t. a given graph.
- Construct a conditional independence graph through conditional independence tests.
- Choose edges based on a measurement of the strength of marginal dependence of attributes.

Unfortunately, none of these approaches is perfect. The first approach fails, because the number of possible graphs grows over-exponentially with the number of attributes and so it is impossible to inspect all of these graphs. The second approach usually starts from the strong assumption that the conditional independences can be represented perfectly and may require independence tests of high order, which are sufficiently reliable only if the datasets are very large. Examples in which the third approach yields a suboptimal result can easily be found [Borgelt and Kruse 2002]. Nevertheless, the second and the third approach, enhanced by additional assumptions, lead to good heuristic algorithms, which usually consists of two ingredients:

1. an *evaluation measure* (to assess the quality of a given model) and
2. a *search method* (to traverse the space of possible models).

This characterization is apt, even though not all algorithms search the space of possible graphs directly. For instance, some search for conditional independences and some for the best set of parents for a given attribute. Nevertheless, all employ some search method and an evaluation measure.

3.2 Computing Projections

Apart from the ingredients of a learning algorithm for graphical models that are mentioned in the preceding section, we need an operation for a technical task, namely the estimation of the conditional or marginal distributions from a dataset of sample cases. This operation is often neglected, because it is trivial in the relational and the probabilistic case, at least for precise data. In the former it is an operation of relational calculus (namely the relational projection operations, which is why we generally call this operation a projection), in the latter it consists in counting sample cases and computing relative frequencies. Only if imprecise information is present, this operation is more complex. In this case the expectation maximization algorithm [Dempster *et al.* 1977, Bauer *et al.* 1997] is drawn upon, which can be fairly costly.

In possibility theory the treatment of imprecise information is much simpler, especially if it is based on the context model. In this case each example case can be seen as a context, which allows to handle the imprecision conveniently inside the context. Unfortunately, computing projections in the possibilistic case is also not without problems: There is no simple operation (like simple counting), with which the marginal possibility distribution can be derived directly from the dataset to learn from. A simple example illustrating this can be found in [Borgelt and Kruse 2002]. However, we have developed a preprocessing method, which computes the *closure under tuple intersection* of the dataset of sample cases. From this closure the marginal distributions can be computed with a simple maximum operation in a highly efficient way [Borgelt and Kruse 2002].

3.3 Evaluation Measures

An *evaluation measure* (or *scoring function*) serves the purpose to assess the quality of a given candidate graphical model w.r.t. a dataset of sample cases, so that it can be determined which model best fits the data. A desirable property of an evaluation measure is decomposability. That is, the quality of the model as a whole should be computable from local scores, for instance, scores for cliques or even scores for single edges. Most evaluation measures that exhibit this property measure the strength of dependence of attributes, because this is necessary for the second as well as the third approach to learning graphical models from data (cf. section 3.1), either to assess whether a conditional independence holds or to find the strongest dependences between attributes.

For the probabilistic case there is a large variety of evaluation measures, which are based on a wide range of ideas and which have been developed for very different purposes. In particular all measures that have been developed for the induction of decision trees can be transferred to learning graphical models, even though this possibility is rarely fully recognized and exploited accordingly. In our research we have collected and studied several measures (e.g., information gain (ratio), Gini index, relieff measure, K2 metric and its generalization, minimum description length etc). This collection together with detailed explanations of the underlying ideas can be found in [Borgelt and Kruse 2002]. Furthermore we have developed an extension of the K2 metric [Cooper and Herskovits 1992, Heckerman *et al.* 1995] and an extension of measure that is based on the minimum description length principle [Rissanen 1983]. In these extensions we added a "sensitivity parameter", which enables us to control the tendency to add further edges to the model. Such a parameter has proven highly useful in applications (cf. the application at DaimlerChrysler, briefly described in section 4).

Evaluation measures for possibilistic graphical models can be derived in two ways: In the first place, the close connection to relational networks can be exploited by drawing on the notion of an α-cut, which is well known from the theory of fuzzy sets [Kruse *et al.* 1994]. With this notion possibility distributions can be interpreted as a *set of relations*, with one relation for each possibility degree α. Then it is easy to see that a possibility distribution is decomposable if and only if each of its α-cuts is decomposable. As a consequence evaluation measures for possibilistic graphical models can be derived from corresponding measures for relational graphical models by integrating over all possible values α. An example of such a measure is the specificity gain [Gebhardt 1997], which can be derived from the Hartley information gain [Hartley 1928], a measure for relational graphical models. Variants of the specificity gain, which results from different ways of normalizing it, are discussed in [Borgelt and Kruse 2002].

Another possibility to obtain evaluation measures for possibilistic networks is to form analogs of probabilistic measures. In these analogs usually a product is replaced by a minimum and a sum by a maximum. Examples of measures derived in this way can also be found in [Borgelt and Kruse 2002].

3.4 Search Methods

The *search method* used determines which graphs are considered. Since an exhaustive search incurs prohibitively large costs due to the extremely high number of possible graphs, heuristic methods have to be drawn upon. These methods usually restrict the set of considered graphs considerably and use the value of the evaluation measure to guide the search. In addition, they are often greedy w.r.t. the model quality in order to speed up the search.

The simplest search methods is the construction of an optimal spanning tree for given edges weights. This method was used first by [Chow and Liu 1968] with Shannon information gain providing the edge weights. In the possibilistic case the information gain may be replaced with the abovementioned specificity in order to obtain an analogous algorithm [Gebhardt 1997]. However, almost all other measures (probabilistic as well as possibilistic) are usable as well.

A straightforward extension of this method is a greedy search for parent nodes in directed graphs, which often starts from a topological order of the attributes that is fixed in advance: At the beginning the evaluation measure is computed for a parentless node. Then parents are added step by step, each time selecting the attribute that yields the highest value of the evaluation measure. The search is terminated if no other parent candidates are available, a user-defined maximum number of parents is reached, or the value of the evaluation measures does not improve anymore. This search method is employed in the K2 algorithm [Cooper and Herskovits 1992] together with the K2 metric as the evaluation measure. Like optimum weight spanning tree this learning approach can easily be transferred to the possibilistic case by replacing the evaluation measure.

In our research we have also developed two other search methods. The first starts from an optimal spanning tree (see above) and adds edges if conditional independences that are represented by the tree not hold. However, the edges that may be added have to satisfy certain constraints, which ensure that the cliques of the resulting graph contain at most three nodes. In addition, these constraints guarantee that the resulting graph has hypertree structure. (A hypertree is an acyclic hypergraph, and in a hypergraph the restriction that an edge connects exactly two nodes is relaxed: A hyperedge may connect an arbitrary number of nodes.) The second methods uses the well-known simulated annealing approach to learn a hypertree directly. The main problem in developing this approach was to find a method for randomly generating and modifying hypertrees that is sufficiently unbiased. These two search methods are highly useful, because they allow us to control the complexity of later inferences with the graphical model at learning time. The reason is that this complexity depends heavily on the size of the hyperedges of the learned hypertree, which can be easily constrained in these approaches.

4 Application

In a cooperation between the University of Magdeburg and the DaimlerChrysler corporation we had the opportunity to apply our algorithms for learning graphical models to a real-world car database. The objective of the analysis was to uncover possible causes for faults and damages. Although the chosen approach was very simple (we learned a two-layered network with one layer describing the equipment of the car and the other possible faults and damages), it was fairly successful. With a prototype implementation of several learning algorithms, we ran benchmark tests against human expert knowledge. We could easily and efficiently find hints to possible causes, which had taken human experts weeks to discover. The sensitivity parameters which we introduced into two evaluation measures (cf. section 3.3) turned out to be very important for this success.

References

[Andersen *et al.* 1989] S.K. Andersen, K.G. Olesen, F.V. Jensen, and F. Jensen. HUGIN — A Shell for Building Bayesian Belief Universes for Expert Systems. *Proc. 11th Int. J. Conf. on Artificial Intelligence (IJCAI'89, Detroit, MI, USA)*, 1080–1085. Morgan Kaufmann, San Mateo, CA, USA 1989

[Baldwin *et al.* 1995] J.F. Baldwin, T.P. Martin, and B.W. Pilsworth. *FRIL — Fuzzy and Evidential Reasoning in Artificial Intelligence.* Research Studies Press/J. Wiley & Sons, Taunton/Chichester, United Kingdom 1995

[Bauer *et al.* 1997] E. Bauer, D. Koller, and Y. Singer. Update Rules for Parameter Estimation in Bayesian Networks. *Proc. 13th Conf. on Uncertainty in Artificial Intelligence (UAI'97, Providence, RI, USA)*, 3–13. Morgan Kaufmann, San Mateo, CA, USA 1997

[Borgelt and Kruse 2002] C. Borgelt and R. Kruse. *Graphical Models — Methods for Data Analysis and Mining.* J. Wiley & Sons, Chichester, United Kingdom 2002

[Castillo *et al.* 1997] E. Castillo, J.M. Gutierrez, and A.S. Hadi. *Expert Systems and Probabilistic Network Models.* Springer-Verlag, New York, NY, USA 1997

[Chickering *et al.* 1994] D.M. Chickering, D. Geiger, and D. Heckerman. *Learning Bayesian Networks is NP-Hard (Technical Report MSR-TR-94-17).* Microsoft Research, Advanced Technology Division, Redmond, WA, USA 1994

[Chow and Liu 1968] C.K. Chow and C.N. Liu. Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Trans. on Information Theory* 14(3):462–467. IEEE Press, Piscataway, NJ, USA 1968

[Cooper and Herskovits 1992] G.F. Cooper and E. Herskovits. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning* 9:309–347. Kluwer, Dordrecht, Netherlands 1992

[Dempster *et al.* 1977] A.P. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society (Series B)* 39:1–38. Blackwell, Oxford, United Kingdom 1977

[Dubois and Prade 1988] D. Dubois and H. Prade. *Possibility Theory.* Plenum Press, New York, NY, USA 1988

[Dubois *et al.* 1996] D. Dubois, H. Prade, and R. Yager, eds. *Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications.* J. Wiley & Sons, New York, NY, USA 1996

[Gebhardt 1997] J. Gebhardt. *Learning from Data: Possibilistic Graphical Models.* Habilitation Thesis, University of Braunschweig, Germany 1997

[Gebhardt and Kruse 1993] J. Gebhardt and R. Kruse. The Context Model — An Integrating View of Vagueness and Uncertainty. *Int. Journal of Approximate Reasoning* 9:283–314. North-Holland, Amsterdam, Netherlands 1993

[Gebhardt and Kruse 1995] J. Gebhardt and R. Kruse. Learning Possibilistic Networks from Data. *Proc. 5th Int. Workshop on Artificial Intelligence and Statistics (Fort Lauderdale, FL, USA)*, 233–244. Springer-Verlag, New York, NY, USA 1995

[Gebhardt and Kruse 1996] J. Gebhardt and R. Kruse. POSSINFER — A Software Tool for Possibilistic Inference. In: [Dubois *et al.* 1996], 407–418

[Hartley 1928] R.V.L. Hartley. Transmission of Information. *The Bell System Technical Journal* 7:535–563. Bell Laboratories, Murray Hill, NJ, USA 1928

[Hammersley and Clifford 1971] J.M. Hammersley and P.E. Clifford. *Markov Fields on Finite Graphs and Lattices.* Unpublished manuscript, 1971. Cited in: [Isham 1981]

[Heckerman 1991] D. Heckerman. *Probabilistic Similarity Networks.* MIT Press, Cambridge, MA, USA 1991

[Heckerman *et al.* 1995] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning* 20:197–243. Kluwer, Dordrecht, Netherlands 1995

[Isham 1981] V. Isham. An Introduction to Spatial Point Processes and Markov Random Fields. *Int. Statistical Review* 49:21–43. Int. Statistical Institute, Voorburg, Netherlands 1981

[Kruse *et al.* 1994] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*, J. Wiley & Sons, Chichester, United Kingdom 1994.

[Lauritzen and Spiegelhalter 1988] S.L. Lauritzen and D.J. Spiegelhalter. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *Journal of the Royal Statistical Society, Series B*, 2(50):157–224. Blackwell, Oxford, United Kingdom 1988

[Pearl 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, USA 1988 (2nd edition 1992)

[Rissanen 1983] J. Rissanen. A Universal Prior for Integers and Estimation by Minimum Description Length. *Annals of Statistics* 11:416–431. Institute of Mathematical Statistics, Hayward, CA, USA 1983

[Saffiotti and Umkehrer 1991] A. Saffiotti and E. Umkehrer. PULCINELLA: A General Tool for Propagating Uncertainty in Valuation Networks. *Proc. 7th Conf. on Uncertainty in Artificial Intelligence (UAI'91, Los Angeles, CA, USA)*, 323–331. Morgan Kaufmann, San Mateo, CA, USA 1991

[Shenoy 1992] P.P. Shenoy. Valuation-based Systems: A Framework for Managing Uncertainty in Expert Systems. In: [Zadeh and Kacprzyk 1992], 83–104

[Zadeh and Kacprzyk 1992] L.A. Zadeh and J. Kacprzyk. *Fuzzy Logic for the Management of Uncertainty.* J. Wiley & Sons, New York, NY, USA 1992