Learning Probabilistic and Possibilistic Networks: Theory and Applications

Rudolf Kruse and Christian Borgelt

Dept. of Information and Communication Systems
Otto-von-Guericke-University of Magdeburg
D-39106 Magdeburg, Germany
e-mail: {kruse,borgelt}@iik.cs.uni-magdeburg.de

Abstract. Inference networks, probabilistic as well as possibilistic, are popular techniques to make reasoning in complex domains feasible. Since constructing such networks by hand can be tedious and time consuming, a large part of recent research has been devoted to learning them from data. In this paper we review probabilistic and possibilistic networks and discuss the basic ideas used in learning algorithms for these types of networks. With an application in the automotive industry we demonstrate that the considered methods are not only of theoretical importance, but also relevant in practice.

1 Introduction

Since reasoning in multi-dimensional domains tends to be infeasible in the domains as a whole — and the more so, if uncertainty and/or imprecision are involved — decomposition techniques, that reduce the reasoning process to computations in lower dimensional subspaces, have become very popular. For example, decomposition based on dependence and independence relations between variables has been studied extensively in the field of graphical modeling [19]. Some of the best-known approaches are Bayesian networks [25], Markov networks [22], and the more general valuation-based networks [32]. But recently possibilistic networks also gained a lot of interest due to their close connection to fuzzy methods [20]. All approaches led to the development of efficient implementations, for example HUGIN [1], PULCINELLA [30], PATHFINDER [13] and POSSINFER [9].

In this paper we review probabilistic and possibilistic networks and discuss methods to learn them from data, i.e. to determine from a database of sample cases an appropriate decomposition of the probability or possibility distribution on the domain under consideration [7, 14, 10, 11]. Such automated learning is important, since constructing a network by hand can be tedious and time-consuming. If a database of sample cases is available, as it often is, learning algorithms can take over at least part of the construction task.

These new methods can be used to do "data mining", i.e. to discover useful knowledge that is hidden in the large amounts of data stored in data warehouses. We demonstrate the practical relevance of this approach with an application in the automotive industry, in which the induction of probabilistic and possibilistic networks was used to find weaknesses in Mercedes Benz vehicles and thus to improve the product quality.

2 Probabilistic and Possibilistic Networks

The basic presupposition underlying every inference network, probabilistic or possibilistic, is that a multi-dimensional distribution can be decomposed without much loss of information into a set of (overlapping) lower-dimensional distributions.[1] This set of lower-dimensional distributions is usually represented as a hypergraph, in which there is a node for each variable and a hyperedge for each distribution of the decomposition. To each node and to each hyperedge a projection of the multi-dimensional distribution (a *marginal distribution*) is assigned: to the node a projection to its variable and to a hypergraph a projection to the set of variables connected by it. Thus hyperedges represent direct influences that the connected variables have on each other, i.e. how constraints on the value of one variable affect the probabilities or possibilities of the values of the other variables in the hyperedge. Reasoning in such a hypergraph consists in propagating evidence, i.e. observed constraints on the values of some of the variables, along the hyperedges.

The idea of propagation can be understood best by a simple example. Imagine three variables, A, B, and C, and a (hyper)graph A—B—C. When evidence about A is fed into the network it is propagated like this: The constraints on the values of variable A stated by the evidence are extended to the space $A \times B$ to obtain constraints on tuples (a_i, b_j), which are then projected to the variable B to compute the constraints on the values of this variable. These constraints are then in turn extended to the subspace $B \times C$ and projected to variable C.

For this scheme to be feasible, the main operations, projection and extension of distributions,

[1] Of course, this presupposition need not hold. A distribution need not be decomposable, even if one accepts a certain limited loss of information. But in such a situation inference networks cannot be used.

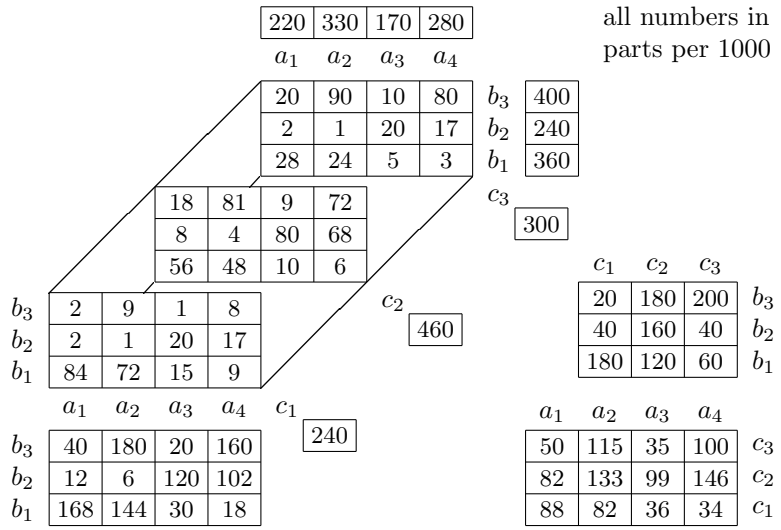


Figure 1: A three-dimensional probability distribution with its marginal distributions (sums over lines/columns). Since in this distribution the equations $\forall i, j, k : P(a_i, b_j, c_k) = \frac{P(a_i, b_j)P(b_j, c_k)}{P(b_j)}$ hold, it can be decomposed into the marginal distributions on the subspaces $\{A, B\}$ and $\{B, C\}$.

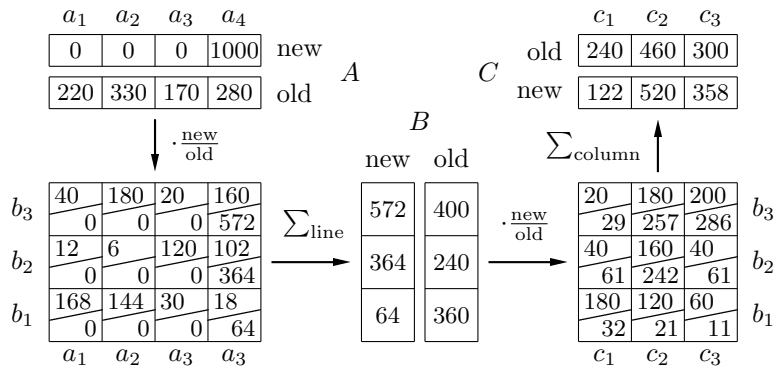


Figure 2: Product/sum propagation of the evidence that variable A has value a_4 in the three-dimensional probability distribution shown in figure 1 using the marginal probability distributions on the subspaces $A \times B$ and $A \times C$.

have to satisfy certain preconditions [31]. In probability theory a product/sum propagation method is used, in which the marginal distribution of e.g. a two-dimensional distribution is calculated by summing over one dimension, that is $P(a_i) = \sum_j P(a_i, b_j)$. Extension consists in multiplying the prior probability distribution on the superset with the quotient of posterior and prior probability on the subset.

An example is given in figures 1 and 2. Figure 1 shows a three-dimensional probability distribution on the joint domain of the variables $A = \{a_1, a_2, a_3, a_4\}$, $B = \{b_1, b_2, b_3\}$, and $C = \{c_1, c_2, c_3\}$, and the marginal distributions calculated by summing over lines/columns. Since in this distribution the equations

$$\forall i, j, k : P(a_i, b_j, c_k) = \frac{P(a_i, b_j)P(b_j, c_k)}{P(b_j)}$$

hold, it can be decomposed into the marginal distributions on the subspaces $A \times B$ and $B \times C$. Therefore it is possible to propagate the observation that variable A has value a_4 using the scheme in figure 2.[2] One can easily check that the resulting marginal distributions are the same as those that can be computed from the three-dimensional distribution directly.

We now turn to possibilistic networks. Our approach rests on an interpretation of a degree of possibility that is based on the context model [8, 20]. In this model possibility distributions are interpreted as information-compressed representations of (not necessarily nested) random sets, a degree of possibility as the one-point coverage of a random set [24]. With this interpretation we can construct possibilistic networks in much the same way as probabilistic networks. The only difference is that instead of a product/sum scheme, minimum/maximum propagation is used. That is, the projection of e.g. a two-dimensional distribution is calculated by determining the maximum over one dimension, extension by calculating the minimum of the prior joint distribution on the superset and the posterior marginal distribution.

An example is given in figures 3 and 4. Figure 3 shows a three-dimensional possibility distribution on the joint domain of the variables A, B, and C

[2] This scheme is a simplification and does not lend itself to direct implementation. Especially joining evidence from two (hyper)edges needs additional computations, which we omitted here.

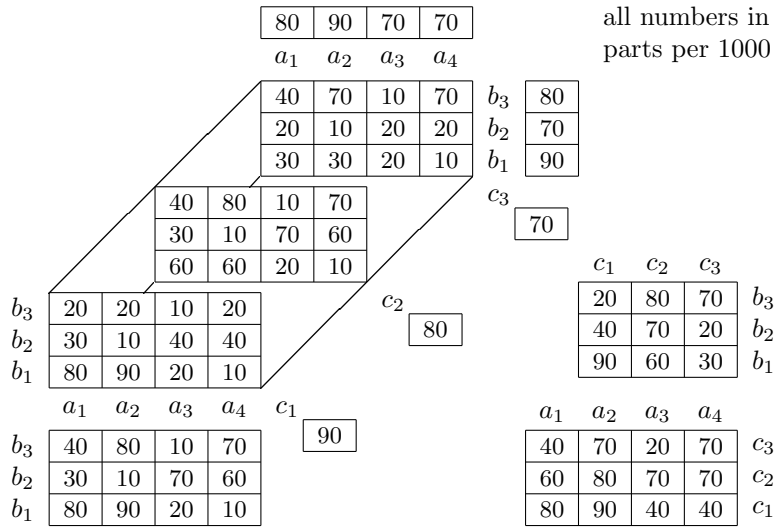


Figure 3: A three-dimensional possibility distribution with maximum projections. Since in this distribution the equations
$\forall i, j, k : \pi(a_i, b_j, c_k) =$
$\min_j(\max_i \pi(a_i, b_j, c_k),$
$\max_k \pi(a_i, b_j, c_k))$
hold, it can be decomposed into the two projections to the subspaces $\{A, B\}$ and $\{B, C\}$.

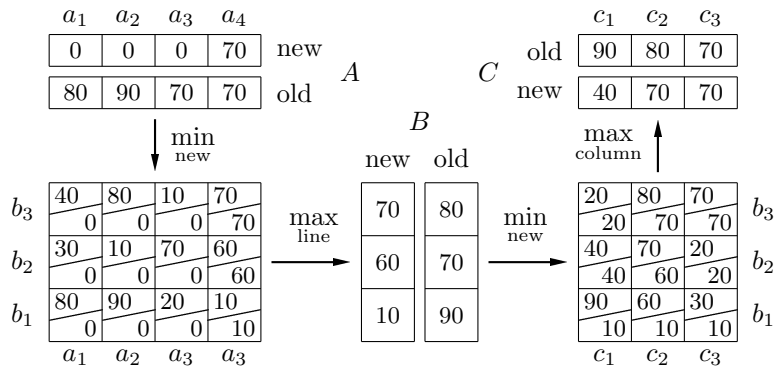


Figure 4: Minimum/maximum propagation of the evidence that variable A has value a_4 in the three-dimensional possibility distribution shown in figure 3 using the marginal distributions on the subspaces $A \times B$ and $A \times C$.

and various marginal distributions determined by computing the maximum over lines/columns. Since in this distribution the equations

$$\forall i, j, k : \pi(a_i, b_j, c_k) = \min_j(\max_i \pi(a_i, b_j, c_k), \max_k \pi(a_i, b_j, c_k))$$

hold, it can be decomposed into marginal distributions on the subspaces $A \times B$ and $B \times C$. Therefore it is possible to propagate the observation that variable A has value a_4 using the scheme shown in figure 4. Again the marginal distributions obtained are the same as those that can be computed directly from the three-dimensional distribution.

3 Learning Inference Networks from Data

An algorithm for learning inference networks consists always of two parts: an evaluation measure and a search method. The evaluation measure estimates the quality of a given decomposition (a given hypergraph) and the search method determines which decompositions (which hypergraphs) are inspected. Often the search is guided by the value of the evaluation measure, since it is usually the goal to maximize (or to minimize) its value.

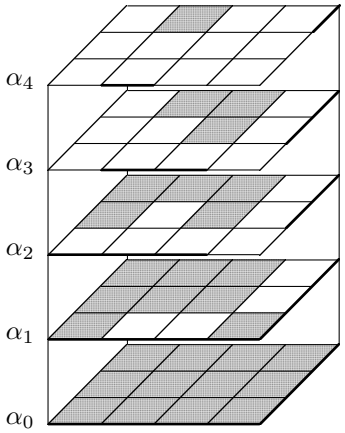
There are several evaluation measures for learning probabilistic as well as for learning possibilistic networks. We can only list some of them here, since limits of space do not allow us to discuss all of them in detail.

Probabilistic Measures

- χ^2-measure
- information gain/mutual information [21, 26, 27]
- (symmetric) information gain ratio [26, 27, 23]
- Gini-index [5]
- symmetric Gini-index [34]
- minimum description length based on relative or on absolute frequency coding [28, 17]
- stochastic complexity [18, 29]
- g-function (a Bayesian measure) [7]

Possibilistic Measures

- d_{χ^2}, a derivate of the χ^2-measure [3, 4]
- d_{mi}, a derivate of mutual information [3, 4]
- specificity gain [10, 2]
- (symmetric) specificity gain ratio [2]



$\log_2 1 + \log_2 1 - \log_2 1 = 0$

$\log_2 2 + \log_2 2 - \log_2 3 \approx 0.42$

$\log_2 3 + \log_2 2 - \log_2 5 \approx 0.26$

$\log_2 4 + \log_2 3 - \log_2 8 \approx 0.58$

$\log_2 4 + \log_2 3 - \log_2 12 = 0$

Figure 5: Illustration of the idea of specificity gain. A two-dimensional possibility distribution is seen as a set of relational cases, one for each α-level. In each relational case, stating the allowed coordinates is compared to stating the allowed value pairs. Specificity gain aggregates the gain in Hartley information that can be achieved on each α-level by computing the integral over all α-levels.

	a_1	a_2	a_3	a_4
b_3	40	80	10	70
b_2	30	10	70	60
b_1	80	90	20	10

$S_{\text{gain}}(A, B) = 0.055$

a_1	a_2	a_3	a_4	
80	80	70	70	b_3
70	70	70	70	b_2
80	90	70	70	b_1

	a_1	a_2	a_3	a_4
c_3	40	70	20	70
c_2	60	80	70	70
c_1	80	90	40	40

$S_{\text{gain}}(A, C) = 0.026$

a_1	a_2	a_3	a_4	
70	70	70	70	c_3
80	80	70	70	c_2
80	90	70	70	c_1

	b_1	b_2	b_3
c_3	20	80	70
c_2	40	70	20
c_1	90	60	30

$S_{\text{gain}}(B, C) = 0.048$

b_1	b_2	b_3	
70	70	70	c_3
80	70	80	c_2
90	70	80	c_1

Figure 6: The specificity gain of the three attribute pairs of the possibility distribution shown in figure 3. On the left is the maximum projection as calculated from the whole distribution, on the right the independent distribution, i.e. the distribution calculated as the minimum of the maximum projections to the single variable domains. Specificity gain can be seen as measuring the difference of the two.

We illustrate the idea underlying these measures by discussing one of them as an example. Since in our research we focus on possibilistic networks, we choose a possibilistic measure: specificity gain. This measure is based on the U-uncertainty measure of *nonspecificity* of a possibility distribution [16], which is defined as

$$\text{nsp}(\pi) = \int_0^{\sup(\pi)} \log_2 |[\pi]_\alpha| d\alpha$$

and can be justified as a generalization of Hartley information [12] to the possibilistic setting [15]. $\text{nsp}(\pi)$ reflects the expected amount of information (measured in bits) that has to be added in order to identify the actual value within the set $[\pi]_\alpha$ of alternatives, assuming a uniform distribution on the set $[0, \sup(\pi)]$ of possibilistic confidence levels α [11].

The role nonspecificity plays in possibility theory is similar to that of Shannon entropy in probability theory. Thus the idea suggests itself to construct an evaluation measure from nonspecificity in the same way as information gain is constructed from Shannon entropy, i.e. by computing the gain in information/specificity that results from using the joint distribution instead of the marginal ones. Therefore we define for two variables A and B the *specificity gain* as

$$S_{\text{gain}} = \text{nsp}(\pi_{\max A}) + \text{nsp}(\pi_{\max B}) - \text{nsp}(\pi_{AB}).$$

Generalizations to more than two variables are easy to find [3, 4]. This measure is equivalent to the one defined in [11].

The idea of specificity gain is illustrated in figure 5. The joint possibility distribution is seen as a set of relational cases, one for each α-level. Specificity gain aggregates the gain in Hartley information for these relational cases by computing the integral over all α-levels.

To demonstrate the application of specificity gain figure 6 states the specificity gain for the three-dimensional possibility distribution shown in figure 3. It is easy to see that interpreting the specificity gain as a (hyper)edge weight and applying the Kruskal algorithm yields the correct decomposition of this distribution.

All of the mentioned measures can be used in combination with a large variety of search methods. Two of the most common methods are optimum weight spanning tree construction [6] and greedy parent selection [7] (K2 algorithm). But in general any heuristic search method can be used, like e.g. simulated annealing, genetic algorithms etc.

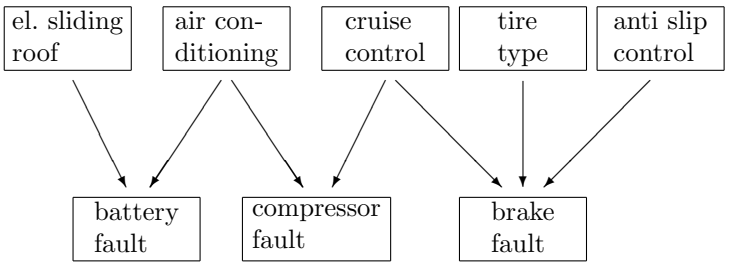


Figure 7: A section of a possible two-layered network for the dependences of faults (bottom) on vehicle properties (top). Since real data and learned networks are highly confidential, this network is fictitious. Any resemblance to actual dependences is purely coincidental.

(fictitious) frequency of battery faults		air conditioning with	air conditioning without
electrical sliding roof	with	9 %	3 %
	without	3 %	2 %

Figure 8: A fictitious example subnet showing a dependence of battery faults on the presence of an electrical sliding roof and an air conditioning system.

4 Application in the Automotive Industry

Even high quality products like Mercedes-Benz vehicles sometimes show undesired behaviour. As a major concern of the Mercedes-Benz AG is to further improve the quality of their products, a lot of effort is dedicated to finding the causes of these faults in order to be able to prevent similar faults from occurring in the future. Therefore the Mercedes-Benz AG maintains a quality information database to control the quality of produced vehicles. In this database for every produced vehicle it is recorded its configuration (product line, motor type, special equipment etc.) and any faults detected during production or maintenance.

In a cooperation with the Data Mining Group of the Daimler-Benz AG Research and Technology Center Ulm we applied INES (Induction of NEtwork Structures), a prototype implementation of the described methods, to this database. This program contains all mentioned evaluation measures and two search methods, optimum weight spanning tree construction and greedy parent selection.

The idea used in this application is very simple. Since we are interested in causes of faults, we learn a two-layered network, in which the top layer contains attributes describing the vehicle configuration and the bottom layer contains attributes describing possible vehicle faults. This is illustrated in figures 7 and 8. (Since real dependences and numbers are, of course, highly confidential, these figures show fictitious examples. Any resemblance to actual dependences and numbers is purely coincidental.) Figure 7 shows a possible learned two-layered network, figure 8 the frequency distribution associated with the first of its subnets. Since in this example the fault rate for cars with an air conditioning system and an electrical sliding roof is considerably higher than that of cars without one or both of these items, we can conjecture that the increased consumption of electrical energy due to installed air conditioning and electrical sliding roof is a cause of increased battery faults.

Although specific results are confidential, we can remark here that on a truck database INES easily found a dependence pointing to a possible cause of a fault, which was already known to the domain experts, but had taken them considerable effort to discover "by hand". Other dependences found were considered by the domain experts as valuable starting points for further technical investigations. Hence we can conclude that learning probabilistic and possibilistic networks is a very useful method to support the detection of product weaknesses.

Acknowledgments

We are grateful to Prof. G. Nakhaeizadeh and his colleagues at the Daimler-Benz AG Research and Technology Center Ulm for supporting this project.

References

[1] 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*, 1080–1085, 1989

[2] C. Borgelt, J. Gebhardt, and R. Kruse. Concepts for Probabilistic and Possibilistic Induction of Decision Trees on Real World Data. *Proc. of the EUFIT'96*, Vol. 3:1556–1560, 1996

[3] C. Borgelt and R. Kruse. Evaluation Measures for Learning Probabilistic and Possibilistic Networks. *Proc. of the FUZZ-IEEE'97*, 1997, to appear

[4] C. Borgelt and R. Kruse. Some Experimental Results on Learning Probabilistic and Possibilistic Networks with Different Evaluation

Measures. *Proc. of the ECSQARU/FAPR'97*, 1997, to appear

[5] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*, Wadsworth International Group, Belmont, CA, 1984

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

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

[8] J. Gebhardt and R. Kruse. A Possibilistic Interpretation of Fuzzy Sets in the Context Model. *Proc. IEEE Int. Conf. on Fuzzy Systems*, 1089-1096, San Diego 1992.

[9] J. Gebhardt and R. Kruse. POSSINFER — A Software Tool for Possibilistic Inference. In: D. Dubois, H. Prade, and R. Yager, eds. *Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications*, Wiley 1995

[10] J. Gebhardt and R. Kruse. Learning Possibilistic Networks from Data. *Proc. 5th Int. Workshop on Artificial Intelligence and Statistics*, 233–244, Fort Lauderdale, 1995

[11] J. Gebhardt and R. Kruse. Tightest Hypertree Decompositions of Multivariate Possibility Distributions. *Proc. Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems*, 1996

[12] R.V.L. Hartley. Transmission of Information. *The Bell Systems Technical Journal* 7:535–563, 1928

[13] D. Heckerman. *Probabilistic Similarity Networks*. MIT Press 1991

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

[15] M. Higashi and G.J. Klir. Measures of Uncertainty and Information based on Possibility Distributions. *Int. Journal of General Systems* 9:43–58, 1982

[16] G.J. Klir and M. Mariano. On the Uniqueness of a Possibility Measure of Uncertainty and Information. *Fuzzy Sets and Systems* 24:141–160, 1987

[17] I. Kononenko. On Biases in Estimating Multi-Valued Attributes. *Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining*, 1034–1040, Montreal, 1995

[18] R.E. Krichevsky and V.K. Trofimov. The Performance of Universal Coding. *IEEE Trans. on Information Theory*, IT-27(2):199–207, 1983

[19] R. Kruse, E. Schwecke, and J. Heinsohn. *Uncertainty and Vagueness in Knowledge-based Systems: Numerical Methods*. Series: Artificial Intelligence, Springer, Berlin 1991

[20] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*, John Wiley & Sons, Chichester, England 1994

[21] S. Kullback and R.A. Leibler. On Information and Sufficiency. *Ann. Math. Statistics* 22:79–86, 1951

[22] 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, 1988

[23] R. Lopez de Mantaras. A Distance-based Attribute Selection Measure for Decision Tree Induction. *Machine Learning* 6:81–92, Kluwer 1991

[24] H.T. Nguyen. Using Random Sets. *Information Science* 34:265–274, 1984

[25] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (2nd edition)*. Morgan Kaufman, New York 1992

[26] J.R. Quinlan. Induction of Decision Trees. *Machine Learning* 1:81–106, 1986

[27] J.R. Quinlan. *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1993

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

[29] J. Rissanen. Stochastic Complexity and Its Applications. *Proc. Workshop on Model Uncertainty and Model Robustness*, Bath, England, 1995

[30] A. Saffiotti and E. Umkehrer. PULCINELLA: A General Tool for Propagating Uncertainty in Valuation Networks. *Proc. 7th Conf. on Uncertainty in AI*, 323–331, San Mateo 1991

[31] G. Shafer and P.P. Shenoy. Local Computations in Hypertrees. Working Paper 201, School of Business, University of Kansas, Lawrence 1988

[32] P.P. Shenoy. Valuation-based Systems: A Framework for Managing Uncertainty in Expert Systems. Working Paper 226, School of Business, University of Kansas, Lawrence, 1991

[33] L. Wehenkel. On Uncertainty Measures Used for Decision Tree Induction. *Proc. IPMU*, 1996

[34] X. Zhou and T.S. Dillon. A statistical-heuristic Feature Selection Criterion for Decision Tree Induction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-13:834–841, 1991