Learning Possibilistic Networks with a Global Evaluation Method

Christian Borgelt
Dept. of Computer Science
University of Magdeburg
D-39106 Magdeburg, Germany
borgelt@iik.cs.uni-magdeburg.de

Jörg Gebhardt
Dept. of Mathematics and Computer Science
University of Braunschweig
D-38106 Braunschweig, Germany
gebhardt@ibr.cs.tu-bs.de

Abstract: Inference networks, probabilistic as well as possibilistic, are popular techniques to make reasoning in multi-dimensional domains feasible. Since constructing them by hand can be tedious and time consuming, a large part of recent research has been devoted to learning inference networks from data. Most of the proposed methods are based on local, i.e. single hyperedge evaluation. In this paper we present a global evaluation method, which in combination with e.g. a simulated annealing search can be used to learn possibilistic inference networks from data.

1 Introduction

Inference networks, probabilistic as well as possibilistic, are popular techniques to make reasoning in complex domains feasible. They are based on the idea to decompose a multi-dimensional distribution into distributions on lower-dimensional subspaces; an idea that has been extensively studied in the field of graphical modeling [15], since the decomposition is usually represented by a hypergraph. For probability distributions the best known network types are Bayesian networks [19] and Markov networks [17]. But the idea has also been transferred to possibility distributions resulting in possibilistic networks [16], and has been generalized to so-called valuation-based networks [24]. All of these approaches led to the development of efficient implementations, for example HUGIN [1], PULCINELLA [23], PATHFINDER [10], and POSSINFER [7].

Since constructing inference networks by hand can be tedious and time consuming, a large part of recent research has been devoted to learning such networks from data [6, 11, 8, 9]. A learning algorithm for this task 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.

A favored property of an evaluation measure is a certain locality, i.e. the possibility to evaluate subgraphs, at best single hyperedges, separately. This is desirable, not only because it facilitates computation, but also because some search methods can make use of such locality. For example, to decompose a multi-dimensional probability distribution, in [5] the value of an evaluation measure (mutual information) is computed on all two-dimensional marginal distributions and then the Kruskal algorithm is applied to determine a maximum weight spanning tree. There are a lot of local evaluation measures for learning probabilistic as well as possibilistic networks, see [2, 3] for a survey. Several of them are transferred from measures used for the induction of decision trees [4, 21, 18, 14, 25]. Analogous measures for possibilistic learning are obtained e.g. by using the U-uncertainty measure of nonspecificity [12, 13] instead of the entropy.

In contrast to these approaches, we introduce in this paper a global evaluation measure for learning possibilistic networks, which is described in section 2. We combine it with a simulated annealing search as sketched in section 3. In section 4 we show some experimental results we obtained with our method on the Danish Jersey cattle blood group determination data [22] and compare them to results obtained with other methods. Finally, in section 5, we draw conclusions.

2 Global Evaluation of Inference Networks

For probabilistic networks a simple global evaluation scheme can be derived from the idea underlying the g-function, a Bayesian measure used in [6] for learning probabilistic networks. From a given network — dependence structure and (conditional) probabilities — the probability of each tuple in the database can be calculated. Multiplying these probabilities yields the probability of the database given the network structure,

provided that the tuples are independent. If we assume all networks to have the same prior probability, the probability of the database given the network can be interpreted as a direct indication of the network quality. Although it is not an absolute measure, since we cannot determine an upper bound for this probability, networks can be compared with it.

The only problem with this method is the treatment of missing values, since for tuples with missing values no definite probability can be calculated. A thorough treatment would be the following: Every missing value of a tuple is instantiated in turn with each possible value, and for each resulting (completely known) tuple the probability is determined. Then e.g. the minimum, average, and maximum of these probabilities are computed. We thus arrive at a minimum, average, and maximum value for the probability of a database, of which the average may be the best to use. It is obvious that this method is applicable only, if the number of missing values per tuple is fairly small, since otherwise the number of tuples to be examined gets too large.

Unfortunately this global evaluation measure cannot directly be turned into a learning method. Although we only need to add a search method to traverse the space of possible solutions, and then to evaluate each candidate solution with this measure, this is not a feasible approach. The main reason is that in the presence of missing values evaluating a network in the way described can take fairly long. If they abound, even a single network cannot be evaluated in reasonable time. Since during a search a large number of networks has to be inspected, the learning time can easily exceed reasonable limits.

We now turn to evaluating possibilistic networks. Although we cannot compute a degree of possibility for the whole database, we can use a similar approach. From the propagation method of possibilistic networks it is obvious that the degree of possibility derivable from a learned network can only be greater or equal to the (true) degree of possibility derivable from the database. (Note, that marginal distributions are calculated by determining the maximum over the dimensions removed.) Hence, the better a network approximates the possibility distribution represented by the database, the smaller the sum of the (network) possibility degrees over the tuples in the database should get.

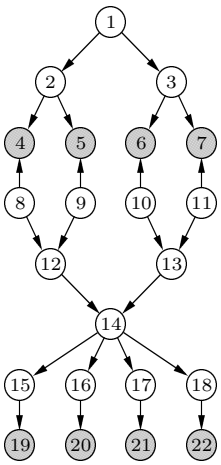
For tuples with missing values one can use a similar approach as above. For each completely known tuple compatible with the tuple missing some values, the degree of possibility is determined from the network and e.g. the minimum, average, and maximum of these degrees is computed. Then these are summed for all tuples in the database. To be in accordance with the ideas underlying possibility theory, the maximum value may be the proper quality measure. Fortunately, if one commits to using the maximum, computation is significantly simplified, since a completely known tuple compatible with a tuple with missing values and having the maximum degree of possibility of all such tuples can easily be determined without inspecting all compatible tuples. We only have to apply the normal propagation method for possibilistic networks.

To further reduce the complexity of the computation, we can use approximations. E.g. the maximal degree of possibility can be determined on each hyperedge and then the minimum of the resulting possibility degrees can be determined. Although with such an approximation we are moving towards local evaluation again, it may be preferable because of the reduced computation time. The experiments we conducted seem to indicate that learning based on such an approximation yields results no worse than those obtained from learning based on exact evaluation.

3 Generating and Modifying Hypertrees

As already pointed out in section 1, a learning algorithm for an inference network always consists of two parts: an evaluation measure and a search method. To apply our global evaluation measure, we decided on a simulated annealing search. That is, a network structure (a hypergraph, or — due to certain restrictions resulting from the propagation scheme — a hypertree) is evaluated, then it is modified and the modified structure is evaluated. If the modified structure is better than the original structure, the original structure is replaced. If it is worse, the original structure is replaced only sometimes, with a probability that depends on how much worse the modified structure is and on a term that can be seen as representing a temperature and which decreases over time (hence the name simulated annealing). Then the structure is modified again. During the whole process it is kept track of the best structure found so far.

In order to apply this scheme we need a method to generate a random hypertree and to (randomly) modify an existing one. We used the following method: First an empty hypertree is generated, i.e. a hypertree with no edges. A hyperedge size (the number of attributes it connects) is chosen at random (not exceeding a given limit, of course), and the corresponding number of attributes is selected. Then it is checked, whether it is possible to add the hyperedge to the hypertree without destroying the hypertree property, i.e. without introducing cycles. If this is possible, the hyperedge is added, otherwise another hyperedge is generated and tested. The process



1 – parental error
2 – dam correct?
3 – sire correct?
4 – stated dam ph.gr. 1
5 – stated dam ph.gr. 2
6 – stated sire ph.gr. 1
7 – stated sire ph.gr. 2
8 – true dam ph.gr. 1
9 – true dam ph.gr. 2
10 – true sire ph.gr. 1
11 – true sire ph.gr. 2
12 – offspring ph.gr. 1
13 – offspring ph.gr. 2
14 – offspring genotype
15 – factor 40
16 – factor 41
17 – factor 42
18 – factor 43
19 – lysis 40
20 – lysis 41
21 – lysis 42
22 – lysis 43

The grey nodes correspond to observable attributes. Node 1 can be removed to simplify constructing the clique tree for propagation.

Figure 1: Domain expert designed probabilistic network for the Danish Jersey cattle blood type determination example. Please note, that this is the structure of a *probabilistic* network and thus need not coincide with a good structure for a *possibilistic* network.

stops, if all attributes are connected.

Since an exact check whether adding a hyperedge maintains the hypertree property is costly, we used a simplified test that sometimes rejects edges although they do not result in a cycle. Due to limits of space we cannot give the details here. Suffice it to say that no structures are excluded, but some structures can be constructed only from a subset of all possible sequences of adding their hyperedges.

Modifying existing hypertrees can be done in a similar fashion. A certain percentage of hyperedges is discarded from the hypertree. This percentage is a parameter of the method, although in our experiments the results were fairly independent of this parameter. But this may be due to special properties of the considered domain. The resulting hypertree, which usually is not connected anymore, is then filled with random hyperedges in the same way as described above for generating a random hypertree.

4 Experimental Results

The experiments described in this section were conducted with a prototype program called INES (Induction of NEtwork Structures) and a prototype implementation of our global evaluation/simulated annealing method. The former contains two search methods (optimum weight spanning tree construction and greedy attribute selection) and all evaluation measures listed in [3].

As a test case we selected the Danish Jersey cattle blood type determination example [22]. This example consists of the domain expert designed probabilistic network, whose structure is shown in figure 1, and a database containing 500 tuples over the twenty-two attributes of the network. Only eight of the attributes, those shaded in the network, are observable. Several tuples of the database contain missing values.

The results of our experiments are shown in table 1. Minimum and average of the possibility degree sum (obtained with another program on the final networks) are added to the results of the global evaluation based learning program (which, as stated above, yields only the maximum). The column labeled 'cnt' states the number of hyperedges, the column labeled 'size' the sum of the hyperedge sizes of the network.

As a baseline for comparisons we first evaluated a network without any hyperedges (isolated nodes) and the domain expert designed network with possibility degrees determined from the database. The results are shown in the first two lines of the table. It is not surprising that the expert designed network performs badly, since the possibilistic scheme exploits a different type of dependence then the probabilistic one.

The next section of the table shows the results of constructing an optimum weight spanning tree for each of the symmetric evaluation measures described in [3] (the network column states the measure used). Although the obtained networks are restricted to two-attribute (hyper)edges, two of them yield remarkably good results compared to networks resulting from other methods.

In a third step we induced networks with a greedy attribute selection method, which is derived from the K2 algorithm for learning probabilistic networks [6]. (Note, that this learning method can lead to a hypergraph

method	network	cnt	size	$\sum \pi_{\min}$	$\sum \pi_{\text{avg}}$	$\sum \pi_{\max}$
none	indep.	0	0	139.8	141.1	158.2
	db. poss.	17	46	137.3	137.8	157.2
optimum weight spanning tree	d_{mi}	21	42	117.6	119.4	144.3
	d_{χ^2}	21	42	120.3	122.5	143.5
	S_{gain}	21	42	123.3	124.9	148.8
	S_{sgr}	21	42	121.1	123.9	148.4
greedy attribute selection	d_{mi}	18	55	115.6	118.2	147.2
	d_{χ^2}	18	57	122.3	123.3	145.2
	S_{gain}	18	57	122.9	123.8	146.1
	S_{gr}	18	49	130.0	131.4	154.1
	S_{sgr}	18	55	123.6	124.7	147.2
global evaluation/ simulated annealing	2/100	21	42	118.2	121.9	143.9
	2/1000	21	42	117.5	120.9	142.5
	3/100	16.2	39.0	115.7	119.2	141.6
	3/1000	15.8	38.7	113.8	117.4	140.7
	4/100	12.3	35.9	113.6	117.1	140.5
	4/1000	12.2	35.9	111.9	115.6	139.5

Table 1: Evaluation of possibilistic networks obtained by different learning algorithms on the Danish Jersey cattle blood type determination data.

that is not a hypertree, since it was originally devised for directed networks.) We selected a topological order compatible with the domain expert designed network and restricted the size of the hyperedges to three attributes. Evaluations of the learned networks are shown in the third section of table 1. At first sight it is surprising that allowing larger edges to be learned by using the greedy attribute selection method seems not to improve the results over optimum weight spanning trees. But a closer inspection reveals that this is due to the restrictions imposed by the topological order. When using a topological order compatible with a very good network obtained from an application of our global evaluation learning method, the results were significantly better (unfortunately there is not enough space to show these results here).

Finally we learned several networks with our global evaluation method. The results are shown at the bottom of table 1. All results are determined by averaging over ten learned networks. The numbers in the network column state the maximal size of the hyperedges and the number of trials performed (i.e. the number of hypertrees evaluated). Obviously the global evaluation leads to better results than the other methods, the more so, if one takes into account the smaller size of the hyperedges. It is worth mentioning that even for hyperedges of size 2 the results are better than those of the optimum weight spanning tree construction.

5 Conclusions

Considering the results we obtained on the Danish Jersey cattle blood group determination data, it is plausible to infer that our global evaluation based learning method can indeed improve on the results of local evaluation based learning. Although greedy attribute selection can lead to comparable results, if an appropriate order of the attributes is chosen, the networks learned with our method are usually much less complex (in terms of hyperedge size). The results of learning normal graphs (edges connecting only two nodes) seem to indicate that global evaluation may be able to perform better than local evaluation, even if optimal use is made of a local evaluation measure.

Acknowledgments

We are grateful to S.L. Lauritzen and L.K. Rasmussen for making the Danish Jersey cattle blood type determination example available for us.

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 and R. Kruse. Evaluation Measures for Learning Probabilistic and Possibilistic Networks. *Proc. FUZZ-IEEE'97*, Barcelona, 1997

[3] C. Borgelt and R. Kruse. Some Experimental Results on Learning Probabilistic and Possibilistic Networks with Different Evaluation Measures. *Proc. ECSQARU'97*, Bad Honnef, Springer, 1997

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

[5] 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

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

[7] 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

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

[9] 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

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

[11] 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

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

[13] 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

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

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

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

[17] 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

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

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

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

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

[22] L.K. Rasmussen. *Blood Group Determination of Danish Jersey Cattle in the F-blood Group System*. Dina Research Report no. 8, 1992

[23] 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

[24] 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

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