

Estimation of Censored Data Water Quality Values Using Decomposable Markov Networks

Zoe J. Y. Zhu* and E. A. McBean

School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada

ABSTRACT. While application of probabilistic inference modeling to large and complex datasets has been limited both as a result of computational difficulties, and implicit/explicit assumptions of normality and lognormality, an alternative is developed herein, based on advancements in graphical modeling using decomposable Markov networks (DMNs). Uncertainties in estimates for censored and/or missing data, are reduced by quantifying dependencies among quality attributes using DMNs. The dependence structure is modeled by a DMN, and established using training data. The improvement from learning DMNs employing the training data is demonstrated using water quality information from water distribution systems. The approach provides a general alternative to traditional techniques for estimating values for censored data.

Keywords: Censored data, inductive learning, knowledge acquisition, Markov networks, model mining, water quality

1. Introduction

As utilized herein, the term ‘censored’ data includes both ‘less than’ or below detection data, and data for a particular constituent which are missing or considered suspect in magnitude. Below detection data arise as a result of the technological limits of laboratory instrumentation. Missing, or potentially incorrect data, arise as a result of the failure to collect and/or analyze for a particular constituent at a specific time or, possibly the assignment of an incorrect magnitude through an action such as transposition of digits; the result is a ‘gap’ in the data record. Regardless, the ability to infill or reliably estimate the value of censored data represents an important dimension assisting subsequent analyses of water quality data.

Examples of the need to develop estimates for censored data are numerous. Examples include responses to concerns with low concentration levels to allow characterization of upper tolerance limits. Further, allowable maximum concentration limits (MCLs) for some chemicals are similar in magnitude to laboratory detection levels (e.g., perchlorates). As well, current trends are for new regulations to be promulgated which require water treatment facilities to regularly monitor water quality for a substantial array of constituents. This may require the estimation of the magnitude of missing data and/or demonstrate that a particular reported data value has been incorrectly recorded.

The above demonstrates some of the rationale for procedures which can estimate magnitudes of uncertain data. In response, a number of alternative procedures have been described in the technical literature for creating such estimates. These include Monte Carlo simulation experiments (e.g.,

Gleit, 1985; Gilliom and Helsel, 1986; Haas and Scheff, 1990), plotting position procedures (e.g., McBean and Rovers, 1984, 1998), regression methods (Newman et al., 1989), the expectation maximization (EM) algorithm (Dempster et al., 1977), and Cohen’s test (Cohen, 1961, 1991). Many of the estimation procedures are based on implicit/explicit assumptions that the underlying distributions for the data are either normal or lognormal. An alternative to the preceding examples which does not require assumptions of normality or lognormality, is based on probabilistic graphical models.

Specifically, Markov network theory and data mining with the joint probability distribution (JPD) are used here to estimate values for the censored data. The entire set of variables, directly monitored and other relevant data if available, and the probabilistic correlations among them, can be incorporated into the analyses allowing very comprehensive base information for the estimation of censored data.

2. Background to Network Theory

A probability network encodes, in a concise manner, a probability distribution over a set of domain variables using a graphical representation. The network structure can be represented by a directed graph, such as in a Bayesian network, or an undirected graph, such as in a decomposable Markov network (DMN). Each node in the graph corresponds to a variable, and each link signifies probabilistic dependence or correlation between the corresponding variables. Graphical separation between two groups of variables represents their conditional independence.

First, graph-theoretic concepts are defined. A set of nodes is *complete* in a graph if they are pairwise connected. For G_1 in Figure 1, the set of nodes $\{c, e, f\}$ is complete, and so is the

* Corresponding author: zoe@cis.uoguelph.ca

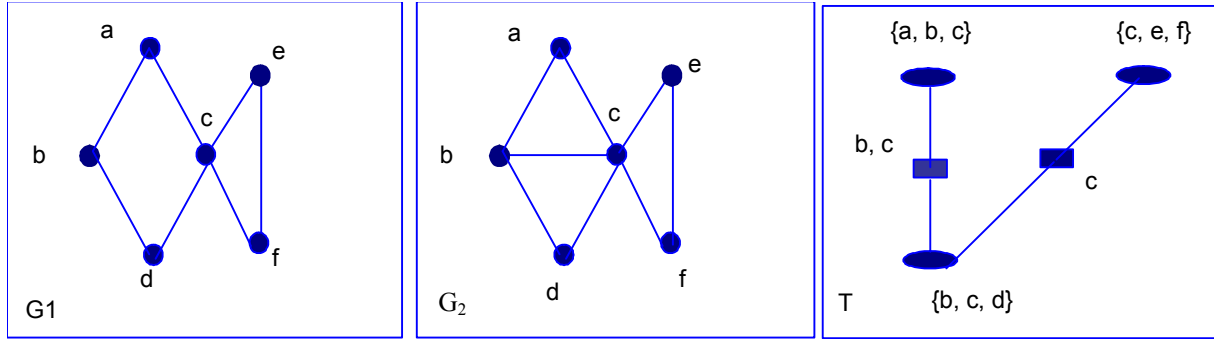


Figure 1. G_1 : a non-chordal graph; G_2 : a chordal graph; T : a junction tree.

set $\{b, d\}$. However, the set $\{a, b, c\}$ is not complete since b and c are not connected. A (maximum) set of nodes which are pairwise-linked (complete) is called a **clique**. G_1 in Figure 1, contains five cliques $\{a, b\}$, $\{b, d\}$, $\{a, c\}$, $\{c, d\}$, and $\{c, e, f\}$. The set $\{c, e\}$ is not a clique because $\{c, e, f\}$ is complete.

A **path** in a graph is a sequence of nodes such that there is a link in the graph between each pair of adjacent nodes in the sequence. A **cycle** is a path where the first node is identical to the last node. A path in an undirected graph has a **chord** if there is a link between two nonadjacent nodes on the path. For example, the path (a, b, d, c, a) of length 4 in G_2 has a chord $\{b, c\}$. A graph is **chordal** if every cycle of length > 3 has a chord. Graph G_1 is not chordal because the path (a, b, d, c, a) of length 4 has no chord. If we add the chord (b, c) to G_1 , it becomes G_2 which is a chordal graph.

Let G be a connected chordal graph. A **junction tree (JT)** can be derived from the chordal graph, G where JT is a tree whose nodes are labeled by cliques of G . The tree is organized such that for each pair of cliques in the tree, their intersection is contained in every clique on the path between them. In a junction tree, each link is labeled by the intersection of the two cliques being connected, and is called a **separator**. A connected graph G has a JT if and only if (iff) G is chordal. Graph G_2 has 3 cliques. They are organized into a tree in Graph T , where each clique is shown as a large oval. Each separator is shown as a box. The tree is a JT because, for example, the intersection of clique $\{c, e, f\}$ and clique $\{a, b, c\}$ is $\{c\}$ which is contained in the clique $\{b, c, d\}$ between them.

Consider three subsets U , V , and Z of nodes in a graph. We use $\langle U|Z|V \rangle$ to denote that every path between a node in U and a node in V contains a node in Z . That is, Z separates U from V . The key which indicates the value of the network approach, is that networks can be used to represent probabilistic information very concisely because they encode probabilistic conditional independence/dependence through graphical separation.

If N is a set of discrete variables in a problem domain, and $V \subseteq N$, a configuration v of V is an assignment of values to every variable in V . A probabilistic domain model (PDM)

over N is an encoding of probabilistic information that defines the probability of every configuration of V for every $V \subseteq N$. A PDM over N can be specified by a joint probability distribution (JPD) over N denoted by $P(N)$. The marginal distribution over $V \subseteq N$ is denoted by $P(V)$. If $U, V, Z \subset N$ are disjoint subsets, U and V are conditionally independent given Z , and denoted as $\text{Ind}(U, Z, V)$ iff $P(u | v, z) = P(u | z)$ whenever $P(v, z) > 0$.

Graphs may be used to encode probabilistic conditional independence. A graph G is an independence map of PDM over N if there is a one-to-one correspondence between nodes of G and variables in N , such that for all disjoint subsets U, V , and Z of N $\langle U|Z|V \rangle \Rightarrow \text{Ind}(U, Z, V)$. That is, in an independence map, variables that are graphically separated (i.e. no link) are conditionally independent.

A DMN of a domain has a chordal graph structure that is an independence map. As reviewed above, cliques of a chordal graph can be organized into a JT. The JPD of the domain is specified through the DMN as follows: Each clique is associated with a probability distribution over its member variables, and so is each separator. The JPD is the product of clique distributions divided by the product of separator distributions.

In a problem domain, some variables are observable, while others are not. Using a DMN, posterior probability distributions of unobservable variables may be computed given observed values of other variables. Such reasoning allows the values of hypothesis variables to be estimated from values of observable variables, based on their probabilistic dependence encoded in the DMN.

Values of observable variables are entered into the DMN at the cliques that contain these variables. Distributions of the corresponding cliques are then updated. Distributions of separators connected to these cliques are updated subsequently. The updates are then propagated to the remaining cliques and separators of the DMN. When the propagation terminates, the posterior probability of each hypothesis variable can be computed from the distribution at the clique that contains it.

To illustrate probability updating in DMN, consider the following simple example: DMN of two cliques $Q = \{c, h\}$ and $R = \{t, c\}$ with the separator $S = \{c\}$. Each variable has the binary space $\{0, 1\}$. The graphical structure of the DMN encodes that h and t are conditionally independent, given c . That is, $P(h | c, t) = P(h | c)$, which implies that $P(h, c, t) = P(c, h) \times P(t, c) / P(c)$.

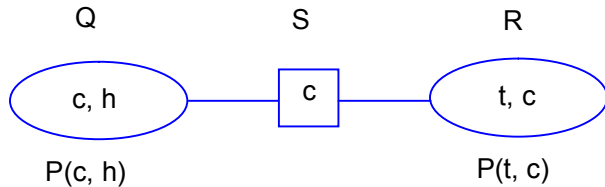


Figure 2. DMN with two cliques.

The following distributions are associated with Q (Table 1.1) and R (Table 1.2). Note that from Table 1.1, $P(h = 0) = P(h = 0, c = 0) + P(h = 0, c = 1) = 0.62$ and $P(h = 1) = 0.38$. In what follows, the DMN graph structure is learned from data. Once the graph structure is obtained, the distributions associated with the cliques can be estimated from the data.

Table 1.1. Prior Distribution of Q

Prior of Q		
h	c	P(h, c)
0	0	0.06
0	1	0.56
1	0	0.24
1	1	0.14

Table 1.2. Prior Distribution of R

Prior of R		
t	c	P(t, c)
0	0	0.255
0	1	0.035
1	0	0.045
1	1	0.665

Suppose t is observed as $t = 1$. Message passing can be used to compute $P(h | t = 1)$. First, $t = 1$ is entered into R . For each probability whose configuration is inconsistent with $t = 1$, the probability value is set to 0. This updates the above $P(t, c)$ into $P(t = 1, c)$ in Table 2.1 below.

After normalization, we obtain $P(t, c | t = 1)$ as shown in Table 2.2: We can compute $P(c | t = 1)$ as $P(c = 0 | t = 1) = 0.063$ and $P(c = 1 | t = 1) = 0.937$. It will be sent as a message over the separator/link S to Q . When Q received $P(c | t = 1)$, it updates $P(h, c)$ as $P(h, c | t = 1) = P(h, c) \times P(c | t = 1) / P(c)$.

Table 2.1. Updating by $t=1$

Updating		
t	c	P(t = 1, c)
0	0	0
0	1	0
1	0	0.045
1	1	0.665

Table 2.2. Posterior Distribution of R

Posterior of R		
t	c	P(t, c t = 1)
0	0	0
0	1	0
1	0	0.063
1	1	0.937

Table 2.3. Posterior Distribution of Q

Posterior of Q		
h	c	P(h, c t = 1)
0	0	$0.06 \times 0.063 / 0.3 = 0.0126$
0	1	$0.56 \times 0.937 / 0.7 = 0.7496$
1	0	$0.24 \times 0.063 / 0.3 = 0.0504$
1	1	$0.14 \times 0.937 / 0.7 = 0.1874$

In the right-hand side, $P(h, c)$ is associated with Q , from which $P(c)$ can be computed as $P(c = 0) = 0.3$ and $P(c = 1) = 0.7$, and $P(c | t = 1)$ is the message that Q received from R . The result is as follows, from which $P(h | t = 1)$ can be obtained as $P(h = 0 | t = 1) = 0.762$ and $P(h = 1 | t = 1) = 0.238$.

In summary, a DMN decomposes a JPD into a set of local distributions. The independence semantics specify how to combine these distributions into JPD. The result is that given n discrete variables (where continuous variables have been discretized) each into k possible values, the JPD requires k^n probability values, which is intractable to specify, and to update. A DMN allows a much more concise representation of the JPD.

Consider a simple example. Agricultural fertilizers as well as treated wastewater contamination may cause excess nitrate. Childrens' sickness (methaemoglobinaemia) may occur due to excess nitrate. Assume treated wastewater is monitored by sampling once every 3 months and the result is recorded in a public report.

This simple domain problem has five variables, agricultural fertilizers (A), treated wastewater contamination (C), excess nitrate (E), monitoring report (R), and sick children (S), each with a space of $\{true, false\}$. The JPD for this simple problem needs $2^5 = 32$ probabilities. Figure 3 shows the chordal graph of the DMN and the JT. We need only to specify $P(A, C, E)$, $P(C, R)$, and $P(S, E)$ with $2^2 + 2^3 + 2^2 = 16$

probabilities to obtain the JPD as $P(A, C, E, S, R) = P(A, C, E) \times P(C, R) \times P(S, E) / [P(S) \times P(C)]$. Note that the number of probabilities of a DMN grows linearly as the number of cliques, and exponentially as the size of the largest clique. Therefore, as long as the largest clique in DMN is reasonably small, DMN modeling is practical. For instance, a clique of 20 binary variables requires about 1 million probabilities to specify its distribution. It occupies 1 MByte memory during its manipulation.

With a CPU that can perform 1 million floating point multiplication operations, it takes about one second to update. A DMN with about a dozen of such cliques is still practical. For most applications, the clique sizes are much smaller than 20 variables. Furthermore, the resolution of discretization determines the base of each exponential (e.g., base 2 in the above term, 2^3). As the resolution is refined, the base increases and the number of required probabilities increases. Hence, a tradeoff between the discretization resolution and efficiency in representation must be made in practice. For instance, if we use 1 million probability values per clique as a limit, then we can have a clique of 12 ternary variables plus one binary variable. If each variable has 4 possible values, then the clique must be no larger than 10 variables in order for the computation to be practical.

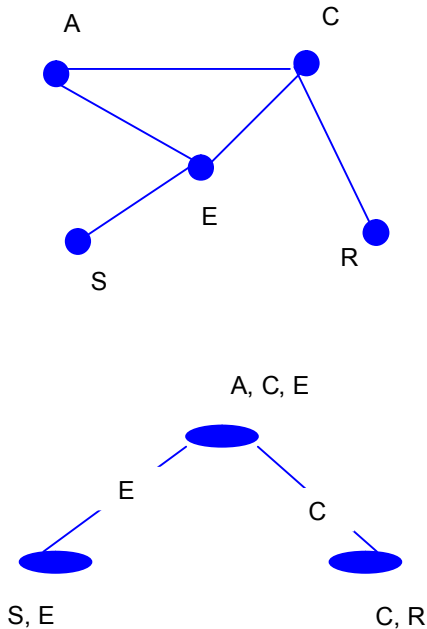


Figure 3. Markov Network.

3. Learning Markov Network From Data

When a DMN model is developed, the qualitative structure of the domain (the graph which encodes the dependence

and independence relations among variables) is specified, as well as the influences (the probability parameters) are quantified. Typically, the most difficult and time-consuming part of the task in building a DMN model is deriving the structure. Learning probabilistic networks from data provide an automatic way to obtain such models.

The common approach to developing the learning structure from data is to introduce a scoring metric and a search procedure. The search procedure generates alternative graphical structures that encode alternative sets of dependence and independence relations. The scoring metric evaluates each structure with respect to the training data and selects the best structure. Several learning algorithms have been proposed such as those by Cooper and Herskovits (1992), Heckman et al. (1995), and Lam and Bacchus (1994).

Here we 'learn' a DMN from a database using a multi-link lookahead learning algorithm (after Xiang et al., 1997). The algorithm selects alternative structures that minimize the entropy of the corresponding DMN. The entropy of a set of variables X , given the probability distribution $P(X)$, is

$$H(X) = - \sum_x P(x) \log P(x) \tag{1}$$

where x is a configuration of X . The JPD of a DMN is

$$P(N) = \prod_C P(C) / \prod_S P(S) \tag{2}$$

where each C is a clique of the DMN and each S is a separator. The distributions $P(C)$ and $P(S)$ are directly estimated from the database. Given this JPD, the entropy of the DMN is

$$H(N) = \sum_C H(C) - \sum_S H(S) \tag{3}$$

where $H(C)$ is the entropy derived from $P(C)$. $H(N)$ is used as the scoring metric. The algorithm starts with an empty graph with all the nodes, but without links. The first level of search is a single-link lookahead. During this level, a single link is added to the empty graph and the score is computed. The DMN of a single link with the best score will be selected to be the current DMN, which terminates one pass in this level. In each subsequent pass, one additional link is added to the current DMN which best improves the score. When no single link can improve the current score by a predefined threshold δh , the current level of search terminates. The next level is the double-link lookahead, which adds two links to the current DMN at each pass. The entire search halts when a predetermined number of levels are completed. With higher numbers of levels used, the computation becomes more expensive; the most effective number of levels to use depends on a tradeoff between the accuracy of learned model and the available

Table 3. Discretized Intervals for Individual Variables

<i>CHLO</i>	<i>ALKA</i>	<i>PH</i>	<i>TEMP</i>	<i>TURB</i>	<i>CONDU</i>	<i>HAAS</i>	<i>TCA</i>
0.28-0.87	10.00-42.00	6.00-7.06	0.00-3.00	0.01-0.04	64.10-147.00	1.00-10.50	0.10-2.65
0.87-1.01	42.00-59.80	7.06-7.15	3.00-6.10	0.04-0.06	147.00-214.00	10.50-16.30	2.65-4.85
1.01-1.20	59.80-76.00	7.15-7.35	6.10-10.00	0.06-0.09	214.00-286.00	16.30-28.60	4.85-11.90
1.20-1.40	76.00-87.50	7.35-7.50	10.00-17.00	0.09-0.14	286.00-323.00	28.60-43.10	11.90-20.80
1.40-1.69	87.50-116.00	7.50-7.73	17.00-22.00	0.14-0.32	323.00-358.00	43.10-59.10	20.80-30.50
1.69-2.67	116.0-230.98	7.73-9.40	22.00-27.50	0.32-2.78	358.00-1120.00	59.10-198.00	30.50-109

computational resource.

In practice, the learning must be established using a finite database. Such a database may contain false correlations that do not exist in the underlying problem domain. They cause the generation of a third type of superfluous link, which are referred to as false links. The probability values associated with false links tend to encode noise contained in the database. The threshold δh used in the algorithm control false links as well as redundant links. As the threshold increases, the algorithm becomes less tolerant to false links and each link 'learned' must correspond to a strong correlation in the data.

4. Development and Application of Markov Network Theory to Water Quality

Application of DMNs to water quality is divided into two stages in the following: a learning stage and an inference stage. In the learning stage, we discover a DMN model over a set of variables of interest from the training data. The 'training data', namely the data where every field (variable) has a definite value, is collected. The data for each variable are discretized with the resultant set of discrete variables known as the domain. The uncertain correlations between the variables are 'learned' from the learning data and encoded by a DMN. As mentioned above, the DMN concisely specifies the JPD over the domain through its chordal graph and the local probability distribution associated with each clique.

In the inference stage, the learned DMN is used to estimate the value of unknown variables given the observed values of other variables. For a given case, the values of some variables are observed, while values of other variables may be unknown or censored, or uncertain.

The task of the inference stage is to infer the uncertain values from the observed values of other variables, and the learned DMN. This is accomplished by entering the observed values into the corresponding cliques of the DMN. After probabilistic reasoning in the DMN, the posterior probability distribution of each unknown variable can be retrieved from the DMN. Due to the concise representation of the JPD using DMN, the computation time grows linearly with the number of cliques, and exponentially with the size of the largest clique. Without using the learned DMN, such computations are exponential in relation to the total number of variables, and

become intractable for a domain with a large number of variables.

5. Experimental Findings

5.1. Application Scenarios

Consider the situation where, for purposes of model training, the uncertain data are ignored and the models are built using the known data. The resulting model is then used to extrapolate the model to develop estimates for the uncertain values.

The case study dataset was derived from data collected by the Ontario Ministry of the Environment (MOE). Total haloacetic acids were selected as the target variable. Haloacetic acids (HAAs) are of concern for public health in drinking water. Haloacetic acids include many different compounds, such as monochloroacetic, dichloroacetic, trichloroacetic and trifluoroacetic acids. At high enough concentrations, they are poisonous to plants, and some are suspected carcinogens.

Some sampling times, and sampling data for total haloacetic acid were missing. There are also 'less than' values.

5.2. Learning from Data

Step (1). The following set of variables for DMN model training was selected: Chlorine (CHLO), Alkalinity (ALKA), pH (PH), Temperature (TEMP), Conductivity (CONDU), Trichloroacetic Acid (TCA), and Haloacetic Acids (HAAs). To establish the training data, any sampling rounds with censored data were removed, with those remaining then being the training dataset. The remaining data set of about 755 cases comprise the training data. The training data was discretized into the following intervals, as summarized in Table 3.

The selection of variables to be included in the training data affects only the computation time but not the training result. For instance, a variable on whether a data record was collected on Monday or Wednesday may be contained in the raw data (before variable selection). The learning algorithm will be able to identify that it is independent of the other variables. However, by including such variables, extra computational time is involved.

The second step of the application then involves applica-

tion of the learning tool in Webweaver III (after Xiang et al., 1997) to extract the decomposable Markov networks from training data as summarized in Figure 4 (ignore the histograms for the present). From the training model results, if two variables are correlated, there is a link between the nodes. For example, HAAs is correlated with the TCA.

5.3. Value Inference

Step (1). Run the Webweaver III inference software to compute the prior probability distribution for each variable using the learned DMN. The software accepts a DMN as the input. It then uses the probability distribution over each clique in the DMN to compute the prior probability for each variable contained in the clique.

Step (2). There are censored data for HAAs. To estimate values for these censored data, all reported values for other parameters are entered, and the posterior distribution for HAAs given the values of other variables is obtained. The histograms in Figure 4 summarize the results.

Step (3). Compute the probability distributions of target variables for each uncertain data record. HAAs are selected as a demonstrative example. From the histogram in Figure 4, we obtain the probabilities of each interval, $P(x_1) = 0.00$; $P(x_2) = 0.0956$; $P(x_3) = 0.0498$; $P(x_4) = 0.463$; $P(x_5) = 0.273$; $P(x_6) = 0.118$. The expected value of the estimate is obtained from (4):

$$Expvalue = \sum_{i=1}^n P(x_i) * x'_i \quad (4)$$

where x'_i is the middle of the interval x_i , and i is the index of the individual intervals. For example, the expected value for HAAs is $5.75 \times 0.00 + 13.4 \times 0.0956 + 22.45 \times 0.0498 + 35.85 \times 0.463 + 51.1 \times 0.273 + 128.55 \times 0.118 = 48.14 \mu\text{g/L}$.

Step (4). To determine the effectiveness of the DMN estimation procedure, test experiments were undertaken. Forty individual monitoring rounds were randomly selected ("hidden"), and not included in the learning dataset. The learned model was then used to predict the "hidden" values of the target HAAs variable, given the other variables (e.g., chlorine, alkalinity, pH, temperature, etc.) The values of the other variables are entered into the DMN model and the posterior probability of the target variables is calculated. The value with the highest posterior probability is chosen as the estimated value. Note that due to the discretization, each value represents a real interval. The hidden value of the record is then compared with the estimated interval. Out of the 40 estimations, in 34 of them (85%) the hidden value fell in the correct interval. In the remaining 6 estimations (15%), the hidden value fell in the interval next to the estimated interval and thus has the lowest possible error given the discretization.

As illustrated in Figure 5, the hidden (real) values of HAAs are highly related to the predicted or learned values. This demonstrates that the DMN is very effective at estimating uncertain/censored values.

6. Conclusions

A novel approach has been developed for estimating magnitudes of uncertain data. The probabilistic dependencies between attributes of interest, measured or censored, are represented as a DMN. The DMN is extracted from data by automated learning. Using the learned DMN, for each data record with uncertain data, the values of unknown attributes are predicted. The predictions were accomplished using probabilistic reasoning, and based on the measured values of the record. The DMN graphical model is general, and can be used with any parent distribution of data and is therefore, very powerful in terms of their applicability to the estimation of uncertain/censored data. Experiments that estimate the hidden value of a target variable in 40 records showed 85% perfect match and 15% minimum error given the discretization.

Acknowledgments. Financial support from the Canada Research Chairs program is gratefully acknowledged. Provision of data by the Ontario Ministry of Environment is very much appreciated.

References

- Cohen, A.C. (1991). *Truncated and Censored Samples*, Marcel Dekker Inc., New York, NY, 119.
- Cohen, A. (1961). Tables for maximum likelihood estimates from single truncated and singly-censored samples. *Technometrics*, 3, 535-541.
- Cooper, G.F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine learn.*, 9, 309-347.
- Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc.*, 39(B), 1-38.
- Druzdzal, M.J., Van der Gaag, L.C., Henrion, M. and Jensen, F.V. (Eds.) (1995). Working notes of the IJCAI-95 workshop on building probabilistic networks: Where do the number come from?, in *International Joint Conferences on Artificial Intelligence*, Menlo Park, CA.
- Gilliom, R.J. and Helsel, D.R. (1986). Estimation of distributional parameters for censored trace level water quality data. 1. Estimation techniques. *Water Resour. Res.*, 22, 135-146.
- Gilbert, R.O. (1987). *Statistical Methods for Environmental Pollution Monitoring*, Van Nostrand Reinhold, New York.
- Gleit, A. (1985). Estimation for Small Normal Data Sets with Detection Limits. *Environ. Sci. Technol.*, 19, 1206-1213.
- Haas, C.H. and Scheff, P.A. (1990). Estimation of averages in truncated samples. *Environ. Sci. Technol.*, 24, 912-919.
- Heckman, D., Geiger, D. and Chickering D.M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20, 197-243.
- Helsel, D.R. (1990). Less than obvious, Statistical treatment of data below the detection limit. *ES&T Features Environ. Sci. Technol.*, 24(12), 1767-1774.
- Lam, W. and Bacchus, F. (1994). Learning Bayesian belief networks: An approach based on the MDL principle. *Comput. Intell.*, 10(4), 269-293.
- McBean, E.A. and Rovers, F.A. (1984). Alternatives for handling detection limit data in impact assessments. *Ground Water Monitoring Review*, 42-44.
- McBean, E.A. and Rovers, F.A. (1998). *Statistical Procedures for Analysis of Environmental Monitoring Data and Risk Assess-*

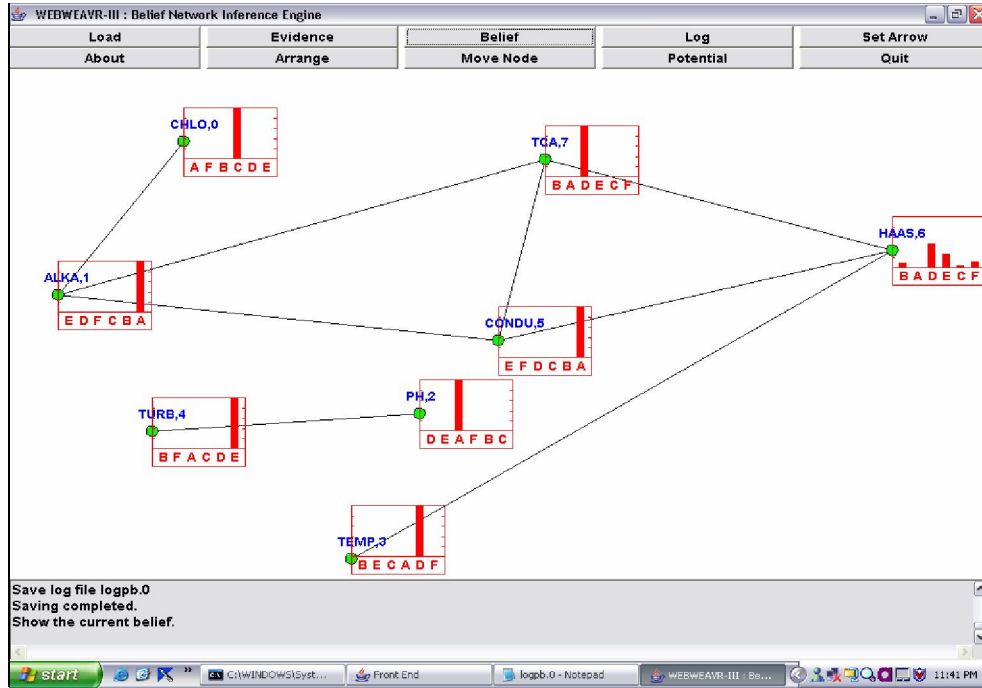


Figure 4. From observable variables to infer the 'less than' value.

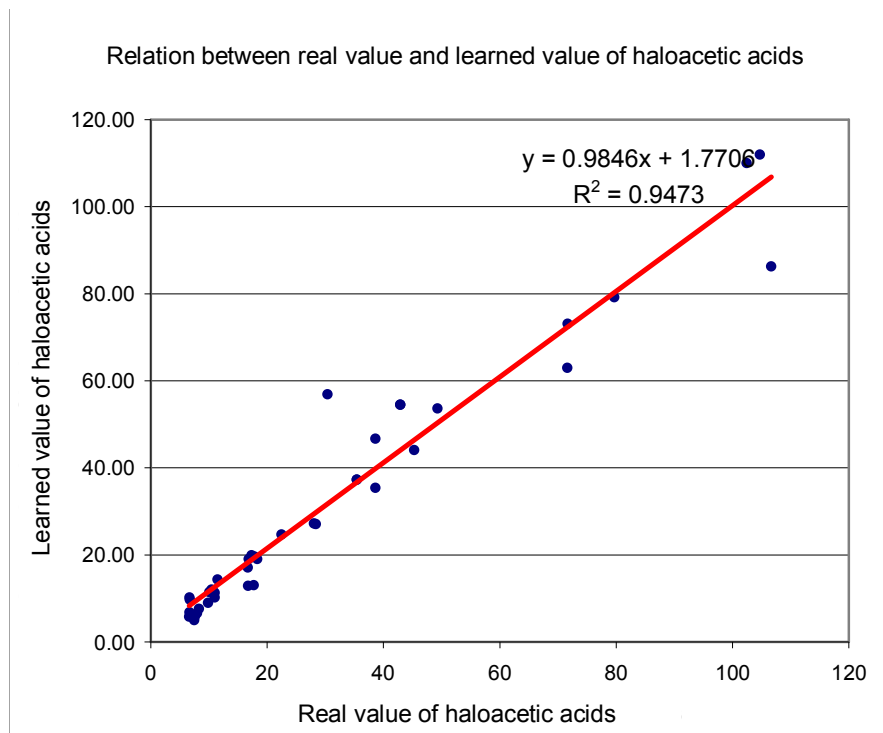


Figure 5. Experimental results.

ment, Prentice Hall PRT, Upper Saddle River, NJ.

Xiang, Y., Wong, S.K.M and Cercone, N.A. (1997). Microscopic study of minimum entropy search in learning decomposable

Markov networks. *Machine Learning*, 26(1), 65-92.

Xiang, Y. (2002). *Probabilistic Reasoning in Multiagent Systems: A Graphical Models Approach*, Cambridge University Press.