Applying Bayesian networks in the game of Minesweeper*

Marta Vomlelová
Faculty of Mathematics and Physics
Charles University in Prague
http://kti.mff.cuni.cz/~marta/

Jiří Vomlel
Institute of Information Theory and Automation of the AS CR
Academy of Sciences of the Czech Republic
http://www.utia.cas.cz/vomlel

Abstract

We use the computer game of Minesweeper to illustrate few modeling tricks utilized when applying Bayesian network (BN) models in real applications. Among others, we apply rank-one decomposition (ROD) to conditional probability tables (CPTs) representing addition. Typically, this transformation helps to reduce the computational complexity of probabilistic inference with the BN model. However, in this paper we will see that (except for the total sum node) when ROD is applied to the whole CPT it does not bring any savings for the BN model of Minesweeper. Actually, in order to gain from ROD we need minimal rank-one decompositions of CPTs when the state of the dependent variable is observed. But this is not known and it is a topic for our future research.

1 Introduction

The game of Minesweeper is a one-player grid game. It is bundled with several computer operating systems, e.g., with Windows or with the KDE desktop environment. The game starts with a grid of $n \times m$ blank fields. During the game the player clicks on different fields. If the player clicks on a field containing a mine then the game is over. Otherwise the player gets information on how many fields in the neighborhood of the selected field contain a mine. The goal of the game is to discover all mines without clicking on any of them. In Figure 1 we present two screenshots from the game. For more information on Minesweeper see, e.g., Wikipedia [12].

*J. Vomlel was supported by grants number 1M0572 and 2C06019 (MŠMT ČR), ICC/08/E010 (Eurocores LogICCC), and 201/09/1891 (GA ČR).

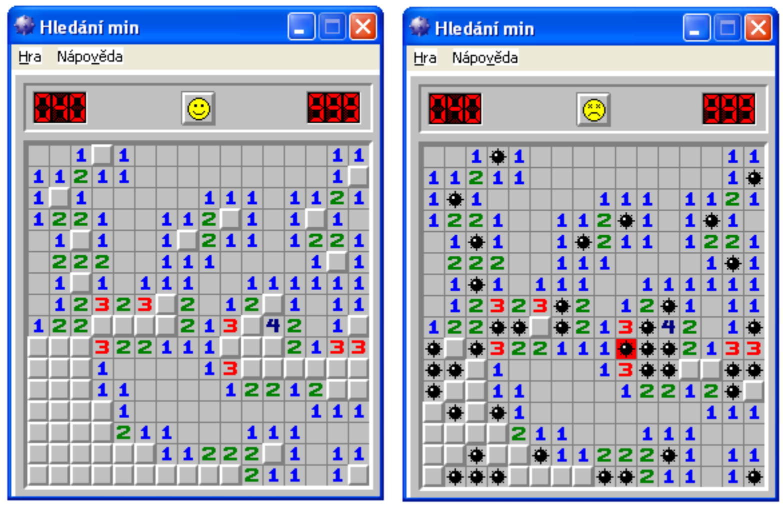

Figure 1: Two screenshots from the game of Minesweeper. The screenshot on the right hand side is taken after the player stepped on a mine. It shows the actual position of mines .

Bayesian networks (BNs) [8, 5, 4] are probabilistic graphical models that use acyclic directed graphs (DAGs) $G = (V, E)$ to encode conditional independence relations among random variables associated with the nodes of the graph. The quantitative part of a BN are conditional probability tables (CPTs) $P(X_i \mid pa(X_i)), i \in V$ that define together the joint probability distribution represented by the BN as the product

$$P(X_i, i \in V) \quad = \quad \prod_{i \in V} P(X_i | pa(X_i)) \ . \tag{1}$$

In many real applications the CPTs have a special local structure that can be further exploited to gain even more efficient inference. Examples of such CPTs are CPTs representing functional (i.e. deterministic) dependence or CPTs representing noisy functional dependence: noisy-or, noisy-and, noisy-max, noisy-min, etc. A method that allows to use the local structure within standard inference techniques (such as [6, 3]) is the rank-one decomposition of CPTs [2, 11, 10]. A typical task solved by BNs is the computation of conditional probabilities given an evidence inserted in the model. In the game of Minesweeper the player is interested in the probabilities of a mine at all fields of the grid given the observations she has made.

The reader may ask why for solving this deterministic game we use BNs instead of the standard techniques of constraint processing. The reason is that we aim at more general problems that correspond to real applications. An advantage of BN models is that they can also represent uncertainty in the modeled domain. This can, for example, model situations when the observations or measurements are noisy. In the game of Minesweeper the observations are the presence or absence of mines. For example, we can extend the BN model of Minesweeper so that the influence of mines from the neighboring fields on the observed count is noisy (i.e. non-deterministic).

2 A Bayesian network model for Minesweeper

Assume a game of Minesweeper on a $n \times m$ grid. The BN model of Minesweeper contains two variables for each field $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, m\}$ on the game grid. The variables in the first set $\mathcal{X} = \{X_{1,1}, \ldots, X_{n,m}\}$ are binary and correspond to the (originally unknown) state of each field of the game grid. They have state 1 if there is a mine on this field and state 0 otherwise. The variables in the second set $\mathcal{Y} = \{Y_{1,1}, \ldots, Y_{n,m}\}$ are observations made during the game. Each variables $Y_{i,j}$ has as its parents[1] from $\mathcal{X}$ that are on the neighboring positions in the grid, i.e.

$$pa(Y_{i,j}) \quad = \quad \left\{ \begin{array}{l} X_{k,\ell} \in \mathcal{X} : \quad k \in \{i-1, i, i+1\}, \ell \in \{j-1, j, j+1\}, \\ (k,l) \neq (i,j), 1 \leq k \leq n, 1 \leq \ell \leq m \end{array} \right\} .$$

The variables from $\mathcal{Y}$ provide the number of neighbors with a mine. Therefore their number of states is the number of their parents plus one. Their CPTs are defined by the addition function for all combinations of states $\mathbf{x}$ of $pa(Y_{i,j})$ and states y of $Y_{i,j}$ as

$$P(Y_{i,j} = y \mid pa(Y_{i,j}) = \mathbf{x}) \quad = \quad \begin{cases} 1 & \text{if } y = \sum_{x \in \mathbf{x}} x \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Whenever an observation of $Y_{i,j}$ is made the variable $X_{i,j}$ can be removed from the BN since its state is known. If its state is 1 the game is over, otherwise it is 0 and the player cannot click on the same field again. When evidence from an observation is distributed to its neighbors also the node corresponding to the observation can be removed from the DAG.

In addition, we need not include into the BN model observations $Y_{i,j}$ that were not observed yet. In order to compute a marginal probability of a variable $X_{i,j}$ we sum the probability values of the joint probability distribution over all combinations of states of remaining variables. It holds for all combinations of states $\mathbf{x}$ of $pa(Y_{i,j})$ that

$$\sum_{y} P(Y_{k,l} = y \mid pa(Y_{i,j}) = \mathbf{x}) \quad = \quad 1$$

therefore the CPT of $Y_{i,j}$ can be omitted from the joint probability distribution represented by the BN.

[1]Since there is a one-to-one mapping between nodes of the DAG of a BN and the variables of BN we will use the graph notion also when speaking about random variables.

The above considerations implies that in every moment of the game we will have at most one node for each field (i, j) of the grid. Each node of a graph corresponding to a grid position (i, j) will be indexed by a unique number $g = j + (i - 1) \cdot m$.

3 A standard approach for inference in BNs

In a standard inference technique such as [6] or [3] the DAG of the BN is transformed to an undirected graph so that each subgraph of the DAG of a BN induced by set variable Y and its parents, say $\{X_1, \ldots, X_m\}$, is replaced by the complete undirected graph on $\{Y, X_1, \ldots, X_m\}$. This step is called moralization. See Figure 2.

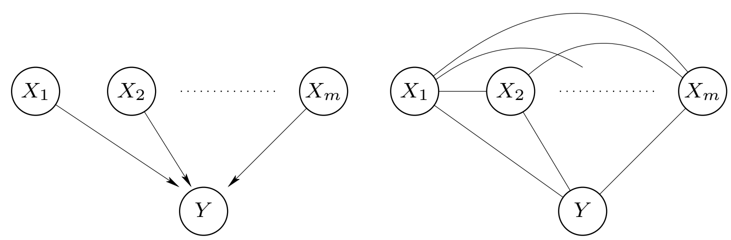

Figure 2: The original subgraph of the DAG induced by $\{Y, X_1, \ldots, X_m\}$ and its counterpart after moralization.

The second graph transformation is triangulation of the moral graph. An undirected graph is triangulated if it does not contain an induced subgraph that is a cycle without a chord of a length of at least four. A set of nodes $C \subseteq V$ of a graph $G = (V, E)$ is a clique if it induces a complete subgraph of G and it is not a subset of the set of nodes in any larger complete subgraph of G.

An important parameter for the inference efficiency is the total table size after triangulation. The table size of a clique C in an undirected graph is $\prod_{i \in C} |X_i|$, where $|X_i|$ is the number of states of a variable X_i corresponding to a node i. The total table size (tts) of a triangulation is defined as the sum of table sizes for all cliques of the triangulated graph. Therefore, it is desirable to find a triangulation of the original graph having the total table size as small as possible. Since this problem is known to be NP-hard different heuristics are often used. In this paper we use minweight heuristics [7] (called H1 in [1]), which, in case of binary variables, is equivalent to the minwidth heuristics [9].

See Figure 3 for an example of the triangulated moral graph for the standard method after twenty observations in the game of Minesweeper on a 10×10 grid. During the triangulation we added five edges only. This is a typical behavior unless we get long cycles in the moral graph.

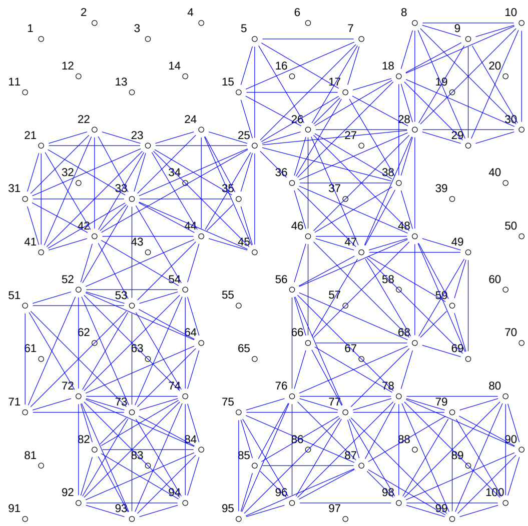

Figure 3: An example of the triangulated moral graph of the BN after twenty observations in the game of Minesweeper on a 10×10 grid for the standard method.

4 Rank-one decomposition of tables representing addition

In the BN model of Minesweeper the CPTs have a special local structure (they correspond to addition). Therefore, we can transform the CPTs using tensor rank-one decomposition (ROD) [2, 11, 10]. A minimal ROD of CPTs representing addition was proposed in [10].

Assume that $X_1, \ldots, X_m$ are parents of Y, and CPT $P(Y \mid X_1, \ldots, X_m)$ represents addition and is defined as in formula (2). Then using Theorem 3 from [10] we can write

$$P(Y = y \mid X_1 = x_1, \ldots, X_m = x_m) \quad = \quad \sum_{b=0}^{m} \xi_b(y) \cdot \prod_{i=1}^{m} \varphi_{i,b}(x_i) \ , \qquad (3)$$

where for $b = 0, \ldots, m$

- ξ_b are real valued functions defined for states $\{0, 1, \ldots, m\}$ of Y and
- for $i = 1, \ldots, m$ $\varphi_{i,b}$ are real valued functions defined for states $\{0, 1\}$ of X_i.

This decomposition is called rank-one decomposition (ROD) of addition. In graphical terms it means that each subgraph of the DAG of a BN induced by set

$\{Y, X_1, \ldots, X_m\}$ is replaced by an undirected graph containing one additional variable B that is connected by an undirected edge with all variables from $\{Y, X_1, \ldots, X_m\}$, see Figure 4. The minimum number of states of B that allows to write a table in the form of equation (3) is called the rank of the table. In case of CPT for addition the rank is $m + 1$, which means we cannot do this decomposition with less than $m + 1$ states of B.

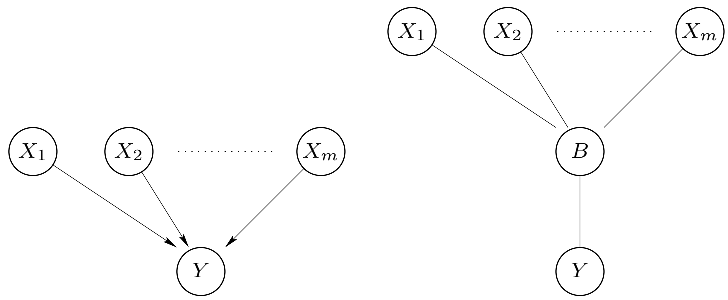

Figure 4: The original subgraph of the DAG induced by $\{Y, X_1, \ldots, X_m\}$ and its counterpart after ROD.

In our current approach we use only a part of table ξ_b corresponding to the observed state y. This implies we can eliminate node corresponding to the variable Y from the graph, so that there is only one node in the graph (i.e, node corresponding to varible B) for the corresponding field of the grid.

The second graph transformation is again triangulation, but this time applied to the undirected graph after ROD.

See Figure 5 for an example of the graph for the model after ROD and after twenty observations in the game of Minesweeper on a 10×10 grid. Note, that while for variables corresponding to unobserved fields of the game grid the number of states is two, for the variables corresponding to the observed fields of the game grid the number of states is typically higher (up to eight states) in the model after ROD. However, there are fewer edges than in the standard approach but during triangulation we added 73 edges, which is much more than for the triangulated moral graph produced by the standard approach. For the triangulated graph of the graph from Figure 5 see Figure 6.

During the computations with the model after ROD we decrease the number of states of observed nodes $Y_{i,j}$ and of corresponding $B_{i,j}$ by one for each observed parent and modify the CPT accordingly. At the same time we delete edges from parents with evidence to observed nodes.

The node for the total number of mines in the game

Typically, in the game of Minesweeper the total number of mines z is known in advance. If we want to exploit this information we need to extend the model by one auxiliary node Z that will impose this constraint. This node has the

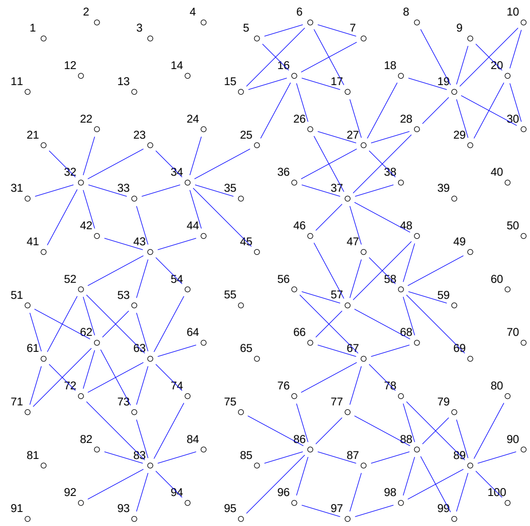

Figure 5: An example of the graph of the BN after ROD and after the same twenty observations as in Figure 3 in the game of Minesweeper on a 10×10 grid.

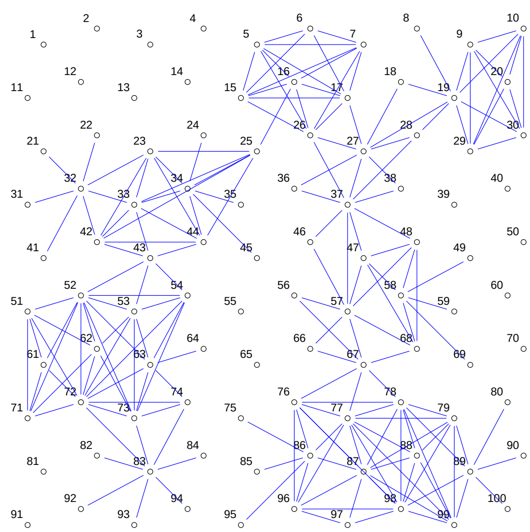

Figure 6: Triangulated graph of the graph from Figure 5.

number of states equal to the total number of fields on the game grid plus one, which is $n \cdot m + 1$. It has all binary nodes from the set $\mathcal{X} = \{X_{1,1}, \ldots, X_{n,m}\}$ as its parents and we enter the evidence $Z = z$ in the BN model.

If we used the standard approach described in Section 3 for node Z we would get a complete graph over all nodes from $\mathcal{X}$. This would make the BN model of the game intractable for larger values of $n \cdot m$.

However, since we use ROD for node Z we connect it with all nodes from $\mathcal{X}$ but we do not require nodes from $\mathcal{X}$ to be pairwise connected. The total table size of the BN with node Z after ROD is $n \cdot m$ larger than the total table size of the BN without node Z. Since this is a constant for a fixed grid we do all experiments without the auxiliary Z node.

5 Experiments

We performed the experiments with the game of Minesweeper for two different grid sizes 10×10 and 20×20. For simplicity we used a random selection of fields to be played and we assumed we never hit the mine during the game. Of course, this is an unrealistic assumption, but since we are interested in the complexity of inference in the BN models it suffices to get an upper bound on the worst case computational complexity. For the results of experiments see Figures 7 and 8, where each value is the average over twenty different games.

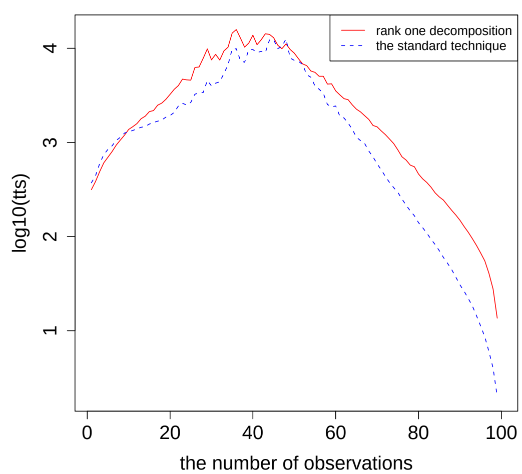

Figure 7: The development of the total table size (tts) for the ROD and standard methods during the game of Minesweeper on the 10×10 grid.

From the results of experiments we can see that the ROD as it is currently applied does not bring any advantages of the standard method. But, recall that

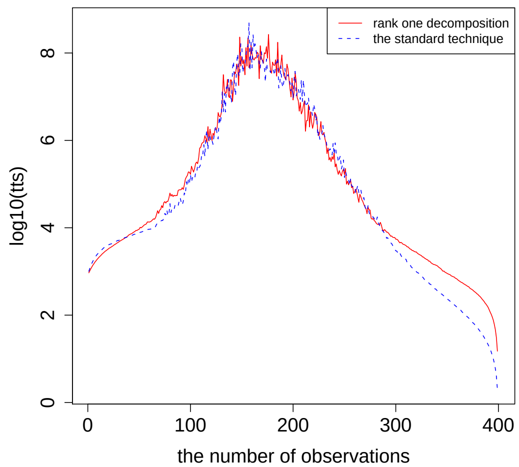

Figure 8: The development of the total table size (tts) for the ROD and standard methods during the game of Minesweeper on the 20×20 grid.

ROD is very useful for the auxiliary node Z, which stands for the total number of mines.

However, the way we apply ROD in Section 4 is not the most efficient utilization of ROD. Given a state y of Y we could find more compact factorizations of $P(Y = y \mid X_1, \ldots, X_m)$. For example in two extreme cases, if $y = m$ or $y = 0$ we know that the state of all $X_i, i = 1, \ldots, m$ is 1 or 0, respectively. Consequently, the rank of $P(Y = y \mid X_1, \ldots, X_m)$ for $y = 0, m$ is one. However, we do not know the rank and minimal rank-one decompositions of $P(Y = y \mid X_1, \ldots, X_m)$ for other values of y. This is a topic for our future research. We conjecture that for most values of y the rank of $P(Y = y \mid X_1, \ldots, X_m)$ will be lower than $m + 1$, which will lessen the total table size for the ROD method.

6 Conclusions

In this paper we report experiments with Bayesian networks build for the game of Minesweeper. During the construction of BN we use several modeling tricks that allow more efficient inference in the BNs for this game. We compare the computational complexity of the inference in the BN model for the standard approach and after the rank-one decomposition. We observed that there are no significant differences in the computational complexity of the two approaches. Probably, the most important theoretical observation is that we miss minimal rank-one decompositions for the CPTs with a local structure when the state of the child variable is observed. We believe this would help to reduce the computational complexity with the BN model after ROD. This is an issue for our future research.

References

[1] A. Cano and S. Moral. Heuristic algorithms for the triangulation of graphs. In B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh, editors, *Advances in Intelligent Computing – IPMU '94: Selected Papers*, pages 98–107. Springer, 1994.

[2] F. J. Díez and S. F. Galán. An efficient factorization for the noisy MAX. *International Journal of Intelligent Systems*, 18:165–177, 2003.

[3] F. V. Jensen, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in recursive graphical models by local computation. *Computational Statistics Quarterly*, 4:269–282, 1990.

[4] Finn Verner Jensen. *Bayesian networks and decision graphs.* Statistics for Engineering and Information Science. Springer Verlag, New York, Berlin, Heidelberg, 2001.

[5] S. L. Lauritzen. *Graphical Models.* Clarendon Press, Oxford, 1996.

[6] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society, Series B*, 50:157–224, 1988.

[7] S. M. Olmsted. *On representing and solving decision problems.* PhD thesis, Stanford University, 1983.

[8] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufman, San Mateo, CA, 1988.

[9] D. J. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. *Graph Theory and Computing*, pages 183–217, 1972.

[10] P. Savicky and J. Vomlel. Exploiting tensor rank-one decomposition in probabilistic inference. *Kybernetika*, 43(5):747–764, 2007.

[11] J. Vomlel. Exploiting functional dependence in Bayesian network inference. In *Proceedings of the 18th Conference on Uncertainty in AI (UAI)*, pages 528–535. Morgan Kaufmann Publishers, 2002.

[12] Wikipedia. Minesweeper (computer game). `http://en.wikipedia.org/wiki/Minesweeper_(computer_game)`, 2009.