An Approximate Tensor-Based Inference Method Applied to the Game of Minesweeper*

Jiří Vomlel and Petr Tichavský

Institute of Information Theory and Automation of the AS CR,
Pod vodárenskou věží 4,
Prague, 182 08, Czech Republic
http://www.utia.cas.cz/

Abstract. We propose an approximate probabilistic inference method based on the CP-tensor decomposition and apply it to the well known computer game of Minesweeper. In the method we view conditional probability tables of the exactly ℓ-out-of-k functions as tensors and approximate them by a sum of rank-one tensors. The number of the summands is $\min\{l+1, k-l+1\}$, which is lower than their exact symmetric tensor rank, which is k. Accuracy of the approximation can be tuned by single scalar parameter. The computer game serves as a prototype for applications of inference mechanisms in Bayesian networks, which are not always tractable due to the dimensionality of the problem, but the tensor decomposition may significantly help.

Keywords: Bayesian Networks, Probabilistic Inference, CP Tensor Decomposition, Symmetric Tensor Rank.

1 Introduction

In many applications of Bayesian networks [1,2,3], conditional probability tables (CPTs) have a certain local structure. Canonical models [4] form a class of CPTs with their local structure being defined either by:

– a deterministic function of the values of the parents (*deterministic models*),
– a combination of the deterministic model with independent probabilistic influence on each parent variable (*ICI models*), or
– a combination of the deterministic model with probabilistic influence on a child of the deterministic model (*simple canonical models*).

In this paper we will pay special attention to deterministic models.

A common task solved efficiently with the help of Bayesian networks is probabilistic inference, which is the computation of marginal conditional probabilities of all unobserved variables given observations of other variables. During the inference the special local structure of deterministic CPTs can be exploited. Díez

* This work was supported by the Czech Science Foundation through projects 13–20012S and 14–13713S.

L.C. van der Gaag and A.J. Feelders (Eds.): PGM 2014, LNAI 8754, pp. 535–550, 2014.

and Galán [5] suggested to rewrite each CPT of a noisy-max model as a product of two-dimensional potentials $\psi_i, i = 1, \ldots, k$. Later, Savický and Vomlel [6] generalized the method to any CPT. Assume a CPT $P(Y = y|X_1, \ldots, X_k)$ with the state y of variable Y being observed, then we can write

$$P(Y = y|X_1, \ldots, X_k) = \sum_B \prod_{i=1}^{k} \psi(B, X_i) \ , \tag{1}$$

where B is an auxiliary variable and the summation proceeds over all its values.

The above equality can be always satisfied if the number of states of B is the product of the number of states of variables $X_1, \ldots, X_k$. The transformation becomes computationally advantageous if the number of states of B is low. It was observed in [6] that each CPT can be understood as a tensor and the minimum number of states of B equals the rank of tensor A defined as

$$\mathcal{A}_{i_1,\ldots,i_k} = P(Y = y|X_1 = x_{i_1}, \ldots, X_k = x_{i_k}),$$

for all combinations of states $(x_{i_1}, \ldots, x_{i_k})$ of variables $X_1, \ldots, X_k$. The decomposition of tensors into the form corresponding to the right hand side of formula (1) is known now as Canonical Polyadic (CP) or CANDECOMP-PARAFAC (CP) decomposition [7,8].

In [9] we have shown how the CP decomposition can be applied to the noisy threshold model of the probabilistic tables. We have presented exact CP decomposition of these tensors, which have rank k if the table size is $2 \times 2 \times \ldots \times 2$ $(k\times)$ in real domain, and slightly lower rank in complex domain. Similar decompositions were derived for the probabilistic tables that represent deterministic exact ℓ-out-of-k functions. The tensor rank is about the same. It was shown that using the CP decomposition approach it is possible to perform probabilistic inference also in cases where the classical method cannot be applied because of a large dimensionality of the probabilistic tables. Next, it was shown that the complexity reduction using CP decomposition is better than in the popular parent divorcing method. Finally, it was shown that the tensor decomposition approach can be combined with another alternative mechanism for Bayesian inference, which is Weighted Model Counting (WMC) [9].

In this paper we take a closer look at tensors representing one specific type of a canonical model – deterministic exact ℓ-out-of-k functions. An ℓ-out-of-k function is a function of k binary arguments that takes the value one if exactly ℓ out of its k arguments take value one – otherwise the function value is zero. These tensors appear naturally in Bayesian network models with CPTs $P(y|X_1, \ldots, X_k)$ representing the addition of binary parent variables $X_1, \ldots, X_k$ and with evidence $Y = y$ on the child variable. We suggest a new approximation by a sum of rank-one tensors, where the number of the summands is $\min\{l+1, k-l+1\}$ and the approximation error can be tuned by a single scalar parameter. This means that we propose less complex (lower rank) approximations, which are computationally simpler, but they approach the desired probabilistic table (tensor) quite accurately, with an arbitrarily small error. The main advantage is the lower rank of the approximation, which is much lower than the true rank (k), if ℓ is low or ℓ is close to k.

The paper is organized as follows. In Section 2 we introduce the necessary tensor notation, define tensors of the exact ℓ-out-of-k functions, and present their basic properties. Section 3 represents the main original contribution of this paper. We propose two approximate CP decompositions of tensors of the ℓ-out-of-k functions based on the symmetric border rank of these tensors. We present a comparison of the CP decomposition with the parent divorcing method in Section 4. In Section 5 we introduce our Bayesian network model for the game of Minesweeper. In Section 6 we apply the suggested decomposition to Minesweeper and compare the computational efficiency and the approximation error of the suggested approximate CP decompositions, the exact CP decomposition, and the standard inference approach based on moralization of parent variables.

2 Preliminaries

Tensor is a mapping $\mathcal{A} : \mathbb{I} \to \mathbb{X}$, where $\mathbb{X} = \mathbb{R}$ or $\mathbb{X} = \mathbb{C}$, $\mathbb{I} = I_1 \times \ldots \times I_k$, k is a natural number called the order of tensor $\mathcal{A}$, and $I_j, j = 1, \ldots, k$ are index sets. Typically, I_j are sets of integers of cardinality n_j. Then we can say that tensor $\mathcal{A}$ has dimensions $n_1, \ldots, n_k$. In this paper all index sets will be $\{0, 1\}$.

Example 1. A visualization of a tensor of order $k = 4$ and dimensions $n_1 = n_2 = n_3 = n_4 = 2$ with successive dimensions alternating between rows and columns[1]:

$$\mathcal{A} = \begin{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \end{pmatrix}$$

Tensor $\mathcal{A}$ has rank one if it can be written as an outer product of vectors:

$$\mathcal{A} = \boldsymbol{a}_1 \otimes \ldots \otimes \boldsymbol{a}_k \quad ,$$

with the outer product being defined for all $(i_1, \ldots, i_k) \in I_1 \times \ldots \times I_k$ as

$$\mathcal{A}_{i_1,\ldots,i_k} = a_{1,i_1} \cdot \ldots \cdot a_{k,i_k} \quad ,$$

where $\boldsymbol{a}_j = (a_{j,i})_{i \in I_j}$, $j = 1, \ldots, k$ are real or complex valued vectors.

Each tensor can be decomposed as a linear combination of rank-one tensors:

$$\mathcal{A} = \sum_{i=1}^{r} b_i \cdot \boldsymbol{a}_{i,1} \otimes \ldots \otimes \boldsymbol{a}_{i,k} \quad , \tag{2}$$

The rank of a tensor $\mathcal{A}$, denoted $rank(\mathcal{A})$, is the minimal r over all such decompositions. The decomposition of a tensor $\mathcal{A}$ to tensors of rank one that sum up to $\mathcal{A}$ is called CP tensor decomposition.

[1] The first dimension is the row of the outer matrix, the second is the column of the outer matrix, the third is the row of the inner matrix, and the fourth is the column of the inner matrix.

Example 2. The tensor $\mathcal{A}$ from Example 1 can be written as:

$$\begin{aligned}\mathcal{A} = \quad & (0,1) \otimes (1,0) \otimes (1,0) \otimes (1,0) \\ & +(1,0) \otimes (0,1) \otimes (1,0) \otimes (1,0) \\ & +(1,0) \otimes (1,0) \otimes (0,1) \otimes (1,0) \\ & +(1,0) \otimes (1,0) \otimes (1,0) \otimes (0,1) \ .\end{aligned}$$

This implies that its rank is at most 4.

The tensors studied in this paper are symmetric.

Definition 1. *Let* $\mathbb{X}$ *be either* $\mathbb{R}$ *or* $\mathbb{C}$*. Tensor* $\mathcal{A} : \{0,1\}^k \to \mathbb{X}$ *is symmetric if for* $(i_1, \ldots, i_k) \in \{0,1\}^k$ *it holds that*

$$\mathcal{A}_{i_1,\ldots,i_k} = \mathcal{A}_{i_{\sigma(1)},\ldots,i_{\sigma(k)}} \ ,$$

for any permutation σ *of* $\{1, \ldots, k\}$*.*

Example 3. The tensor $\mathcal{A}$ from Example 1 is symmetric.

Definition 2. *Let* $\mathbb{X}$ *be either* $\mathbb{R}$ *or* $\mathbb{C}$*. The symmetric rank* $srank_{\mathbb{X}}(\mathcal{A})$ *of a tensor* $\mathcal{A}$ *is the minimum number of symmetric rank-one tensors taking values from* $\mathbb{X}$ *such that their linear combination is equal to* $\mathcal{A}$*, i.e.,*

$$\mathcal{A} = \sum_{i=1}^{r} b_i \cdot \boldsymbol{a}_i^{\otimes k} \ , \tag{3}$$

where $\boldsymbol{a}_i, i = 1, \ldots, r$ *are vectors of length equal to dimensions of* $\mathcal{A}$ *taking values form* $\mathbb{X}$*,* $b_i \in \mathbb{X}, i = 1, \ldots, r$*, and* $\boldsymbol{a}_i^{\otimes k}$ *is used to denote* $\underbrace{\boldsymbol{a}_i \otimes \ldots \otimes \boldsymbol{a}_i}_{k \ copies}$*.*

As we will discuss later some tensors $\mathcal{A}$ can be approximated with arbitrarily small error by tensors of lower rank than their rank. This can be formalized using the notion of border rank.

Definition 3. *The border rank of* $\mathcal{A} : \{0,1\}^k \to \mathbb{R}$ *is*

$$brank(\mathcal{A}) = \min\{r : \forall \varepsilon > 0 \ \ \exists \mathcal{E} : \{0,1\}^k \to \mathbb{R}, ||\mathcal{E}|| < \varepsilon, rank(\mathcal{A} + \mathcal{E}) = r\} \ ,$$

where $|| \cdot ||$ *is any norm.*

Next we give an example of a tensor that has its border rank at most two. The example is a specialization of Example 4.2 from [10].

Example 4. Let $k = 4$. Then for $q > 0$ tensor

$$\begin{aligned}\mathcal{B}(q) = \quad & \frac{1}{2q} \cdot (1,q) \otimes (1,q) \otimes (1,q) \otimes (1,q) \\ & -\frac{1}{2q} \cdot (1,-q) \otimes (1,-q) \otimes (1,-q) \otimes (1,-q)\end{aligned}$$

has rank at most two. Note that

$$\lim_{q \to 0} \mathcal{B}(q) = \mathcal{A}$$

where $\mathcal{A}$ is the tensor from Example 1. This implies that $brank(\mathcal{A}) \leq 2$.

A class of tensors that appear in the real applications are tensors representing functions. In this paper we pay special attention to tensors representing the exact ℓ-out-of-k functions, i.e. a Boolean function taking value 1 if and only if exactly ℓ of its k inputs have value 1.

Definition 4. *Tensor $\mathcal{S}(\ell, k) : \{0,1\}^k \to \{0,1\}$ represents an exact ℓ-out-of-k function if it holds for $(i_1, \ldots, i_k) \in \{0,1\}^k$:*

$$\mathcal{S}_{i_1,\ldots,i_k}(\ell, k) = \delta(i_1 + \ldots + i_k = \ell)$$
$$\delta(i = \ell) = \begin{cases} 1 \text{ if } i = \ell \\ 0 \text{ otherwise.} \end{cases}$$

Example 5. The tensor $\mathcal{A}$ presented in Examples 1– 4 is tensor $\mathcal{S}(1, 4)$. It follows from Example 2 it has rank at most 4 and border rank at most 2 (Example 4).

3 Approximate Tensor Decompositions

Tensors $\mathcal{S}(\ell, k)$ were studied in [9]. It was shown that their symmetric rank in the real domain is equal to k for all integer k, ℓ [9, Proposition 1 and Proposition 3], except for the trivial cases $\ell \in \{0, k\}$, where the rank is one [9, Lemma 2], symbolically:

$$srank_{\mathbb{R}}(\mathcal{S}(\ell, k)) = \begin{cases} k & \text{for } 1 \leq \ell \leq (k-1) \\ 1 & \text{for } \ell \in \{0, k\}. \end{cases} \quad (4)$$

In the complex domain the tensor rank is slightly smaller for ℓ in vicinity of $k/2$:

$$srank_{\mathbb{C}}(\mathcal{S}(\ell, k)) = \max\{\ell + 1, k - \ell + 1\} \quad \text{for } 1 \leq \ell \leq (k-1), \quad (5)$$

see [9, Proposition 3]. The proofs in [9] are constructive.

In practical applications, the tensors with ℓ near zero, $\ell = 1, 2, 3, 4$ and with ℓ near k, i.e. $\ell = k-1, k-2, k-3, k-4$, seem to be more common that those with ℓ around $k/2$. For example, in Section 5 we discuss an application to the computer game of Minesweeper where CPTs with values of ℓ around $k/2$ appear rarely. For ℓ near $k/2$ we recommend decomposition in the complex domain [9], which has the rank specified in formula (5).

Earlier it was shown in [10, Theorem 4.3] that the symmetric border rank of the tensor $\mathcal{S}(\ell, k)$ can be bounded as

$$brank(\mathcal{S}(\ell, k)) \leq \min\{\ell + 1, k - \ell + 1\} \; .$$

It means that the tensor can be expressed as a limit of a series of tensors having the displayed rank. Unfortunately, the CP decomposition of the approximating tensors are such that some elements of the factor matrices converge to zero and some other converge to infinity. For practical applications it is indeed possible to work with an inaccurate decomposition, provided that the approximation error is sufficiently low, and the corresponding factor matrices do not have too large Frobenius norm, so that there are no serious numerical issues with these factors.

The paper [10, Section 6] contains a general construction of series of tensors of rank $\min\{\ell+1, k-\ell+1\}$ that converge to $\mathcal{S}(\ell, k)$ for a general pair (k, ℓ). Convergence of the series is relatively slow with respect to the Frobenius norm of the factor matrices, except for the special case $\ell = 1$.

1-out-of-*k*

The tensor $\mathcal{S}(1, k)$ can be written as a limit

$$\mathcal{S}(1, k) = \lim_{x \to \infty} \mathcal{S}_a(1, k, x)$$

where

$$\begin{aligned} \mathcal{S}_a(1, k, x) &= (x, y)^{\otimes k} - (x, -y)^{\otimes k} \\ y = y(x, k) &= \frac{1}{2x^k} \ . \end{aligned}$$

Obviously, rank of $\mathcal{S}_a(1, k, x)$ is 2. The error of the approximation is

$$E(1, k, x) = \|\mathcal{S}(1, k) - \mathcal{S}_a(1, k, x)\|_\infty \;=\; \frac{1}{4x^{2k}} \ .$$

Note that Example 4 represents a special case for $k = 4$.

A Method for Tensor Approximations

In this paper we extend the above result for the cases $\ell = 2, 3, 4$ and a general k. In other words we write the tensor $\mathcal{S}(\ell, k)$ as a limit of an appropriately parameterized tensor $\mathcal{S}_a(\ell, k, x)$ of a low rank,

$$\mathcal{S}(\ell, k) = \lim_{x \to \infty} \mathcal{S}_a(\ell, k, x) \ .$$

We have conducted a series of numerical experiments attempting to decompose the tensors numerically, using Levenberg-Marquardt method [11] starting from different random starting points, which allowed us to guess a functional form of suitable approximations.

Once the functional form of the approximation was found, we used symbolic matlab tool to evaluate the assumed tensor decomposition as a function of 2 to 5 designed parameters. These parameters were selected to approximate the exact tensor of interest to the maximum possible extent, with a single parameter left. This parameter allows one to control the quality of the approximation, possibly at the expense of numerical stability.

2-out-of-k

For $\ell = 2$ we get

$$\mathcal{S}_a(2,k,x) = (x,y)^{\otimes k} + (x,-y)^{\otimes k} - 2x^k(1,0)^{\otimes k}$$
$$y = y(x,k) \;=\; \frac{1}{\sqrt{2}x^{(k-2)/2}} \;.$$

The error of the approximation is

$$E(2,k,x) = \|\mathcal{S}(2,k) - \mathcal{S}_a(2,k,x)\|_\infty = \frac{1}{2x^k} \;.$$

3-out-of-k

For $\ell = 3$ we get

$$\mathcal{S}_a(3,k,x) = (x,y)^{\otimes k} - (x,-y)^{\otimes k} - (z,w)^{\otimes k} + (z,-w)^{\otimes k}$$

with

$$\begin{aligned} y = y(x,k) &= -\frac{1}{2}x^{1-k/3} \\ z = z(x,y,k) &= \left(\frac{2x^{3k-3}y^3}{2x^{k-3}y^3 - 1}\right)^{1/(2k)} \\ w = w(x,y,z,k) &= y\left(\frac{x}{z}\right)^{k-1} \;. \end{aligned}$$

The error of the approximation is

$$E(3,k,x) = \|\mathcal{S}(3,k) - \mathcal{S}_a(3,k,x)\|_\infty \;=\; \frac{3}{2x^{2k/3}} \;.$$

4-out-of-k

Finally, for $\ell = 4$ we get

$$\mathcal{S}_a(4,k,x) = (x,y)^{\otimes k} + (x,-y)^{\otimes k} - (z,w)^{\otimes k} - (z,-w)^{\otimes k} - 2(x^k - z^k)(1,0)^{\otimes k}$$

with

$$\begin{aligned} y = y(x,k) &= x^{1-k/4} \\ z = z(x,y,k) &= \left(\frac{2x^{2k-4}y^4}{2x^{k-4}y^4 - 1}\right)^{1/k} \\ w = w(x,y,z,k) &= y\left(\frac{x}{z}\right)^{k/2-1} \;. \end{aligned}$$

The error of the approximation is

$$E(4,k,x) = \|\mathcal{S}(4,k) - \mathcal{S}_a(4,k,x)\|_\infty \;=\; \frac{3}{2x^{k/2}} \;.$$

$(k-\ell)$-out-of-k

Approximations for $\ell = k-1, k-2, k-3, k-4$ can be constructed from $\ell = 1, 2, 3, 4$, respectively, by swapping values of all vectors in the CP decompositions, i.e., from

$$\mathcal{S}_a(\ell, k) = \sum_{i=1}^{r} b_i \cdot \boldsymbol{a}_i^{\otimes k} \ ,$$

we get

$$\mathcal{S}_a(k-\ell, k) = \sum_{i=1}^{r} b_i \cdot \overline{\boldsymbol{a}}_i^{\otimes k} \ ,$$

where vector $\overline{\boldsymbol{a}}_i = (y_i, x_i)$ is obtained from $\boldsymbol{a}_i = (x_i, y_i)$ by swapping its values.

Approximate Decompositions of Threshold Tensors

Similar functional forms can be derived also for approximate decompositions of threshold tensors discussed in [9]. For tensors $\mathcal{T}(\ell, k)$ with ℓ near zero ($\ell = 1, 2, 3, 4$) and with ℓ near k ($\ell = k-1, k-2, k-3, k-4$) we can use already derived expressions for $\mathcal{S}(\ell, k)$ and combine them using the following identity:

$$\mathcal{T}(\ell, k) = \begin{cases} \sum_{m=\ell}^{k} \mathcal{S}(m, k) & \text{for } \ell = k-1, k-2, \ldots \\ (1,1)^{\otimes k} - \sum_{m=1}^{\ell-1} \mathcal{S}(m, k) & \text{for } \ell = 1, 2, \ldots. \end{cases}$$

Complex Valued Decompositions

It is worth noting that if complex-valued factors in the decomposition are allowed, the approximation is possible with higher accuracy for the same variable x. We consider the same functional form as in the real-valued decomposition.

In particular, for $\ell = 3$ the dependence of the variable y on x can be taken as $y = (2^{-2/3} - 1/x^2)\, x^{1-k/3}$, and for $\ell = 4$ we propose the choice $y = (2^{-1/2} - 1/x^2)\, x^{1-k/4}$. With these choices, the variables z and w become complex-valued, but the decomposition remains valid and the total approximation error is reduced.

Approximation Errors

In Table 1 we present maximum[2] approximation error for approximate decompositions of ℓ-out-of-k tensors. These errors were obtained for variable $x = 10$. With higher variable x, the approximation errors could be still lower, but for the price of a risk of numerical issues.

[2] The maximum is taken over all absolute values of differences of all corresponding pairs of tensor values.

The first two column present errors of the complex-valued and real-valued decomposition described above. For $\ell = 1$ and $\ell = 2$ there is no difference, no increase of accuracy can be attained in the complex domain. For $\ell = 3$ and $\ell = 4$ the former decomposition is more accurate, but for the price of involving arithmetic with complex numbers. The third column contains the error obtained by a rank-k approximation in the real domain suggested in [9]. The error is effectively zero.

Table 1. Maximum approximation error for approximate and exact decompositions of ℓ-out-of-k tensors. The errors for decompositions of $(k-\ell)$-out-of-k tensors are the same as of ℓ-out-of-k by their construction.

CPT	complex approx.	real approx.	real exact
1-out-of-4	2.5e-09	2.5e-09	1.465e-14
2-out-of-4	5,00e-05	5,00e-05	1.908e-15
1-out-of-5	2.5e-11	2.5e-11	1.399e-14
2-out-of-5	5,00e-06	5,00e-06	5.995e-15
1-out-of-6	2.5e-13	2.5e-13	1.654e-14
2-out-of-6	5,00e-07	5,00e-07	6.573e-14
3-out-of-6	3.78e-06	6.3e-05	1.248e-13
1-out-of-7	2.5e-15	2.5e-15	7.472e-14
2-out-of-7	5,00e-08	5,00e-08	2.315e-12
3-out-of-7	8.144e-07	4.454e-05	6.625e-14
1-out-of-8	1.11e-16	1.11e-16	1.798e-12
2-out-of-8	1.192e-07	1.192e-07	1.396e-12
3-out-of-8	1.755e-07	2.18e-05	2.376e-12
4-out-of-8	5.698e-06	0.00015	1.239e-12

Remark 1. For the three cases 2-out-of-4, 3-out-of-6 and 4-out-of-8 presented in Table 1 we can get exact CP complex decomposition of the same symmetric rank as the approximate one – see formula (5).

4 A Comparison with the Parent Divorcing Method

A different transformation that can be applied to CPTs of ℓ-out-of-k functions is the parent-divorcing method [12].

On the right hand side of Figure 1 we present the graph after parent divorcing and consequent moralization for $k = 5$ and $\ell = 1$. First, we add $k-2$ auxiliary variables and connect each of them with two parents. The CPT of each auxiliary node is for $\ell \leq k/2$, $j = 2, \ldots, k-1$ and $y = 0, 1, \ldots, \min\{j, \ell+1\}$ defined as

$$P(Y_j = y | Y_{j-1} = y', X_j = x) = \begin{cases} 1 & \text{if either } y = y' + x \text{ or} \\ & y' + x \geq \ell \text{ and } y = \ell + 1 \\ 0 & \text{otherwise,} \end{cases}$$

where for $j = 2$ variable Y_{j-1} is replaced by X_1. Note that we need not consider the values of Y_j greater than $\ell + 1$ since in these cases the exact ℓ-out-of-k function is already ensured to be zero and by adding the values of the remaining variables $X_j, \ldots, X_k$ the sum cannot decrease. For $\ell > k/2$ the CPTs are defined similarly but with values of y swapped.

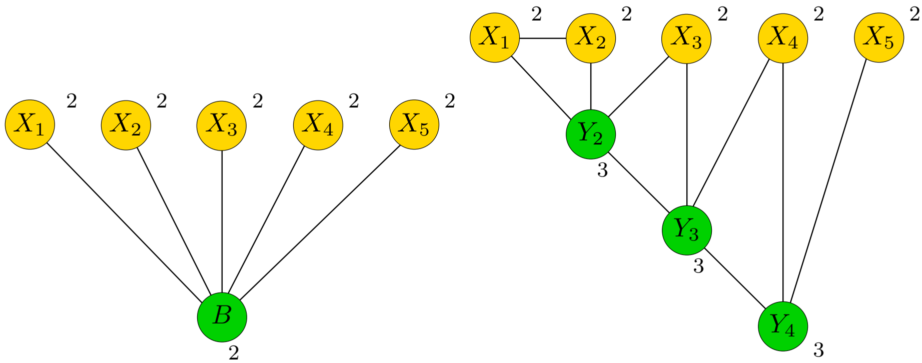


Fig. 1. The graph after the CP decomposition (left) and the parent divorcing method with consequent moralization (right) for $k = 5$ and $\ell = 1$. The small numbers attached to nodes represent number of states of corresponding variables.

In the moralization step all parents of each node are pairwise connected by an undirected edge and directions of edges are removed. Finally, the last auxiliary node is connected to the last parent node by an undirected edge. The table corresponding to clique $\{Y_{k-1}, X_k\}$ is for the observed value y of Y defined as

$$P(Y = y | Y_{j-1} = y', X_j = x) = \begin{cases} 1 & \text{if either } \ell = y' + x \text{ and } y = 1 \text{ or} \\ & \ell \neq y' + x \text{ and } y = 0 \\ 0 & \text{otherwise.} \end{cases}$$

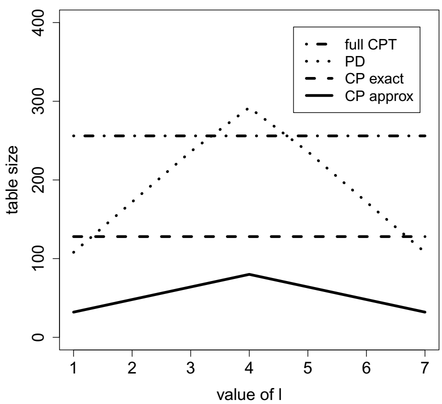
Table Size

The table size is the number of numerical values (memory units) that are needed to represent all tables of a CPT. In Table 2 we compare the table size of CPT after the real approximate (ts_{CPa}) and real exact CP decompositions (ts_{CPe}) of ℓ-out-of-k tensors compared with the parent divorcing (PD) method (ts_{PD}) and the full table size (ts_f). The table sizes were for given $k \geq 4$ and $(k-1) \geq \ell \geq 1$ computed by following formulas:

$$
\begin{aligned}
ts_{CPa} &= 2k \min\{k-\ell+1, \ell+1\} \\
ts_{CPe} &= 2k^2 \\
ts_{PD} &= \sum_{j=2}^{k-1} 2\min\{j, k-\ell+2, \ell+2\} \cdot \min\{j+1, k-\ell+2, \ell+2\} \\
&\quad +2\min\{k-\ell+2, \ell+2\} \\
ts_f &= 2^k
\end{aligned}
$$

Table 2. Table size for the real approximate and real exact CP decompositions compared with the parent divorcing (PD) method and the full table size. On the right we present a plot of table sizes for $k = 8$. The table sizes for decompositions of $(k-\ell)$-out-of-k tensors are the same as of ℓ-out-of-k by their construction.

CPT	ts_{CPa}	ts_{CPe}	ts_{PD}	ts_f
1-out-of-4	16	32	36	16
2-out-of-4	24	32	44	16
1-out-of-5	20	50	54	32
2-out-of-5	30	50	76	32
1-out-of-6	24	72	72	64
2-out-of-6	36	72	108	64
3-out-of-6	48	72	136	64
1-out-of-7	28	98	90	128
2-out-of-7	42	98	140	128
3-out-of-7	56	98	186	128
1-out-of-8	32	128	108	256
2-out-of-8	48	128	172	256
3-out-of-8	64	128	236	256
4-out-of-8	80	128	292	256



5 The Game of Minesweeper

In [13] the computer game of Minesweeper was used to illustrate a few modeling tricks utilized when applying Bayesian networks in real applications. In [10] this game was used to illustrate the benefits of CP tensor decompositions of CPTs of noisy exact ℓ-out-of-k functions. In this paper we will use the Bayesian network model of this game to compare exact and approximate CP tensor decompositions of deterministic CPTs of exact ℓ-out-of-k functions.

Minesweeper is a one-player game. 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 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 find all mines without clicking on them. In Figure 2 two

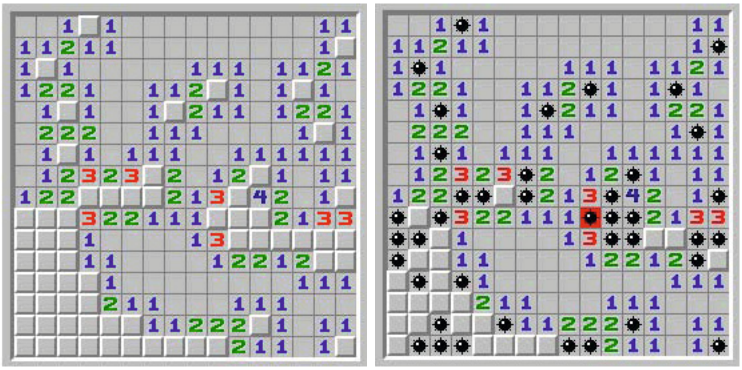


Fig. 2. Two screenshots from the game of Minesweeper. The screenshot on the right hand side is taken after the player stepped on a mine and it shows the actual position of mines.

screenshots from the game are presented. More information about Minesweeper can be found at Wikipedia [14].

The Bayesian network of Minesweeper contains two variables for each field on the game grid. One variable is binary and corresponds to the (originally unknown) state of each field of the game grid. It has state 1 if there is a mine on this field and state 0 otherwise. The second variable corresponds to the observation made during the game. It has state variables on the neighboring positions in the grid as its parents. It conveys the number of its neighbors with a mine. Thus, its number of states is the number of its parents plus one. Its CPT is defined by the addition function. Whenever an observation is made the corresponding state variable can be removed from the BN since its state is known. If its state is 1 the game is over, otherwise its state is 0. When evidence from an observation is distributed to its neighbors the node corresponding to the observation can be removed. By entering evidence to a CPT of addition a table of exact ℓ-out-of-k function is created. Variables from the second set that were not observed are not included in the BN model since they are barren variables [2, Section 5.5.1]. The above considerations implies that in every moment of the game we will have at most one node for each field of the grid and all tables in the BN are either one dimensional priors that are the same for each position or tables of exact ℓ-out-of-k function. Thus, the BN of Minesweeper represent a good test bed for inference algorithms exploiting the local structure of tables of ℓ-out-of-k functions. This paragraph is a digest of a more detailed description of the BN of Minesweeper in [10, Section 7.1].

In Figure 3 we present an example of the game grid after 175 random observations of fields without a mine from the point of view of a game oracle. The players do not see the positions of mines.

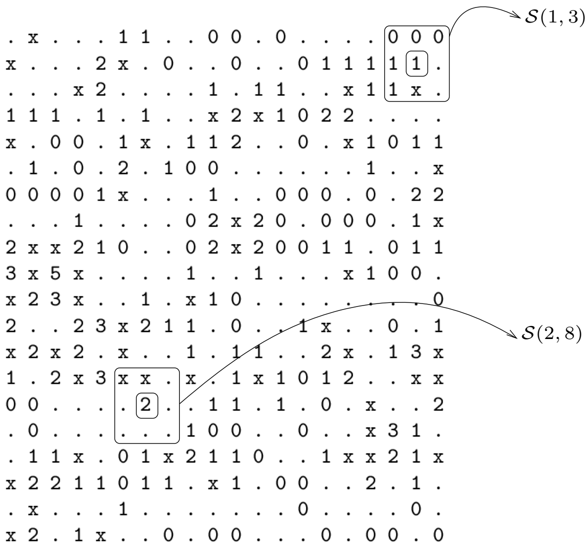


Fig. 3. The game grid after 175 random observations. The points ". " represent covered fields without a mine, crosses "x" represent covered fields with a mine. The numbers correspond to uncovered fields and give the number of mines in the neighborhood. The neighborhoods of 2 out of 175 observed fields are denoted by rectangles. In the corresponding steps of the game the CPTs $\mathcal{S}(1, 3)$ and $\mathcal{S}(2, 8)$ are added to the Bayesian network. Note that nodes of observed fields are connected to uncovered fields only – therefore we add CPT $\mathcal{S}(1, 3)$ instead of $\mathcal{S}(1, 8)$.

6 Numerical Experiments

We performed experiments with Minesweeper of 20×20 grid size. We implemented all algorithms in the R language [15]. In each of 350 steps of the game the oracle randomly selected a field to be observed from those 350 not containing a mine and we created a Bayesian network corresponding to that step. We compared two transformations:

– the standard method consisting of moralization and triangulation steps and
– the CP tensor decomposition applied to CPTs with higher number of parents[3] – for other CPTs we used the moralization followed by the triangulation step.

In both networks we used the lazy propagation method [16] which is junction tree based methods where the computations are performed with messages that are kept as long as possible as lists of tables.

[3] We applied CP tensor decomposition only when the total size of created tables was less than the size of the table after moralization. Roughly speaking, this happened when the number of parents was higher than three for the approximate methods and higher than six for the exact method.

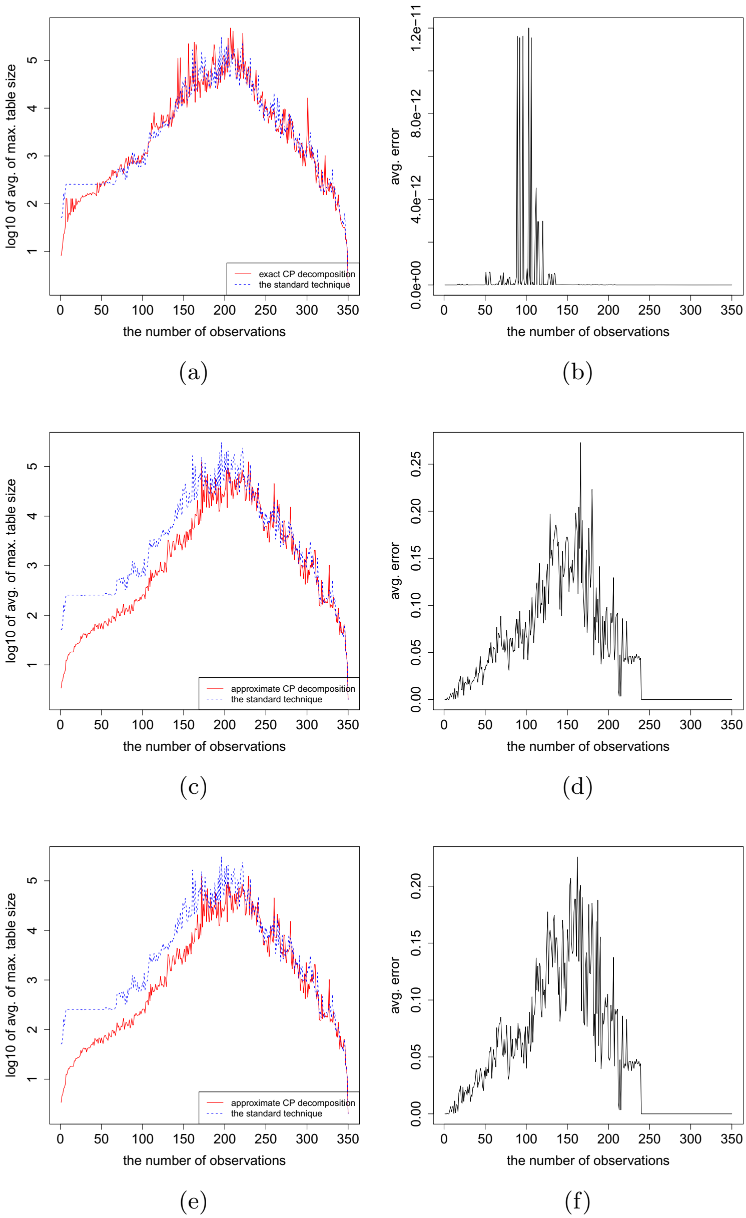


Fig. 4. Results of the experiments for the real exact decompositions – (a) and (b), the real approximate decompositions – (c) and (d), and the complex approximate decompositions – (e) and (f).

At each step of the game we recorded (1) decadic logarithm of the size of the largest table created during the lazy propagation and (2) the conditional marginal probabilities given current observations computed by (1) the standard method and three versions of methods exploiting the CP tensor decomposition: (2) the real exact decompositions, (3) the real approximate decompositions, and (4) the complex approximate decompositions. The value of parameter x was set to 10.

For the results of experiments see Figure 4. All values represent the average over ten different games. The plots in the first column present the decadic logarithm of the size of the largest table created at each step of the game. The plots in the second column present the average error (measured by the absolute value of difference) of the conditional marginal probabilities.

We can see that when using exact CP tensor decomposition the size of largest tables are not reduced but there is no approximation error (as expected). When the real or complex approximate CP tensor decomposition is used at some stages of the game the size of largest tables is reduced by an order of magnitude. But this is achieved at an expense of a certain loss of the accuracy. The loss is lower for the complex CP tensor decomposition (ranging from 0 to 0.2) than for the real CP tensor decomposition (ranging from 0 to 0.25). Unfortunately, for some particular configurations the approximation error is high. The numerical stability of probabilistic inference seems to be an important issue here – however, we did not study this issue in depth and leave it as a topic for our future research.

7 Conclusions

The reduction of maximal table size is important for applications where large tables imply memory requirements that forbid using standard probabilistic inference schemes based on moralization and triangulation. The game of Minesweeper is a prototype application, where the tensor decomposition approach can be really useful for reducing computational load of the inference mechanism. For this particular game, the computational savings are low, if any, because the maximum number of parents (order of the probability tables) is at most 8. However, we can imagine more complex situations, where the number of parents is higher. In such cases the computational advantage would be more apparent.

We can consider, for example, a 3D generalization of Minesweeper, where each field has not only 8 but 26 neighbors. The complexity of such problem would grow significantly. We believe that the tensor decompositions presented in this paper might be very suitable for the probabilistic inference in Bayesian networks of such a type.

References

1. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
2. Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer, New York (2001)

3. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs, 2nd edn. Springer (2007)
4. Díez, F.J., Druzdzel, M.J.: Canonical probabilistic models for knowledge engineering.Technical Report CISIAD-06-01, UNED, Madrid, Spain (2006)
5. Díez, F.J., Galán, S.F.: An efficient factorization for the noisy MAX. International Journal of Intelligent Systems 18, 165–177 (2003)
6. Savicky, P., Vomlel, J.: Exploiting tensor rank-one decomposition in probabilistic inference. Kybernetika 43(5), 747–764 (2007)
7. Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition. Psychometrika 35, 283–319 (1970)
8. Harshman, R.A.: Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-mode factor analysis. UCLA Working Papers in Phonetics 16, 1–84 (1970)
9. Vomlel, J., Tichavský, P.: Probabilistic inference with noisy-threshold models based on a CP tensor decomposition. International Journal of Approximate Reasoning 55, 1072–1092 (2014)
10. Vomlel, J.: Rank of tensors of l-out-of-k functions: an application in probabilistic inference. Kybernetika 47(3), 317–336 (2011)
11. Phan, A.H., Tichavský, P., Cichocki, A.: Low complexity damped Gauss-Newton algorithms for CANDECOMP/PARAFAC. SIAM Journal on Matrix Analysis and Applications 34, 126–147 (2013)
12. Olesen, K.G., Kjærulff, U., Jensen, F., Jensen, F.V., Falck, B., Andreassen, S., Andersen, S.K.: A MUNIN network for the median nerve — a case study on loops. Applied Artificial Intelligence 3, 384–403 (1989); Special issue: Towards Causal AI Models in Practice
13. Vomlelová, M., Vomlel, J.: Applying Bayesian networks in the game of Minesweeper. In: Proceedings of the Twelfth Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty, pp. 153–162 (2009)
14. Wikipedia: Minesweeper (computer game), `http://en.wikipedia.org/wiki/Minesweeper_(computer_game)`
15. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008) ISBN 3-900051-07-0
16. Madsen, A.L., Jensen, F.V.: Lazy propagation: A junction tree inference algorithm based on lazy evaluation. Artificial Intelligence 113(1–2), 203–245 (1999)