KYBERNETIKA — VOLUME 47 (2011), NUMBER 3, PAGES 317–336

RANK OF TENSORS OF ℓ-OUT-OF-k FUNCTIONS: AN APPLICATION IN PROBABILISTIC INFERENCE

Jiří Vomlel

Bayesian networks are a popular model for reasoning under uncertainty. We study the problem of efficient probabilistic inference with these models when some of the conditional probability tables represent deterministic or noisy ℓ-out-of-k functions. These tables appear naturally in real-world applications when we observe a state of a variable that depends on its parents via an addition or noisy addition relation. We provide a lower bound of the rank and an upper bound for the symmetric border rank of tensors representing ℓ-out-of-k functions. We propose an approximation of tensors representing noisy ℓ-out-of-k functions by a sum of r tensors of rank one, where r is an upper bound of the symmetric border rank of the approximated tensor. We applied the suggested approximation to probabilistic inference in probabilistic graphical models. Numerical experiments reveal that we can get a gain in the order of two magnitudes but at the expense of a certain loss of precision.

Keywords: Bayesian networks, probabilistic inference, uncertainty in artificial intelligence, tensor rank, symmetric rank, border rank

Classification: 68T37, 62E15, 15A69

1. INTRODUCTION

Often, in real world applications of Bayesian networks many of the conditional probability tables have a certain local structure. The local structure can be exploited to speed-up computations performed with these networks. In this paper we study conditional probability tables representing deterministic ℓ-out-of-k functions and their noisy counterparts. 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 and otherwise the function value is zero. The noisy version of an ℓ-out-of-k function allows noise at the inputs of the function. The conditional probability tables of these functions appear naturally when we observe a state of a variable that depends on its parents via an addition or noisy addition relation. These relations are quite common in practice since in many domains we want to represent the model where several causes have an additive impact on a dependent variable. A good example are medical applications where, for example, several causes can have an additive effect on a symptom , e. g., on the body temperature [18, page 557].

Each conditional probability table can be viewed as a multidimensional array, i. e., as a tensor. We prove that the border rank of the tensors corresponding to

ℓ-out-of-k functions is at most $\min\{\ell+1, k-\ell+1\}$. The same holds for their noisy counterparts – tensors of ℓ-out-of-k functions with a noise on their inputs.

In this paper we empirically evaluate the computational gain achieved by rank-one decomposition in a generalization of the game of Minesweeper. Specifically, we compare the computations of the probability of a mine at all uncovered fields using the lazy propagation in the junction tree method [9] performed by the standard technique and by use of the rank-one decomposition of the conditional probability tables representing the noisy ℓ-out-of-k function. We observed a gain in the order of two magnitudes. However, this gain is achieved at the expense of a certain loss of precision. We observed that in most cases the precision is quite high but we also observed several cases where the conditional probabilities given evidence were substantially different for the exact and approximate computations. Thus, it remains to be a task for a future research to find rank-one decompositions of these tables that would avoid the loss of precision possibly with a lower computational gain.

This paper is organized as follows. In Section 2 we introduce basic tensor notion. In Section 3 we introduce tensors of ℓ-out-of-k functions and their basic properties. In Section 4 we prove the main theoretical result of this paper about the upper bound of the border rank of tensors representing the ℓ-out-of-k functions. In Section 7 we introduce our generalization of the game of Minesweeper and the task we will solve using a probabilistic inference method. In Section 8 we compare the results of numerical computations performed by the standard lazy propagation in the junction tree and by the same method after the rank-one decomposition. In Section 9 we present the conclusions.

2. PRELIMINARIES

Tensor is a mapping $\boldsymbol{A} : \mathbb{I} \rightarrow \mathbb{R}$, where $\mathbb{I} = I_1 \times \ldots \times I_d$, d is a natural number called the order of tensor $\boldsymbol{A}$, and $I_j, j = 1, \ldots, d$ are index sets. Typically, I_j are sets of integers $\{v_j, \ldots, v_j + n_j\}$. Then we can say that tensor $\boldsymbol{A}$ has dimensions $n_1 + 1, n_2 + 1, \ldots, n_k + 1$.

Definition 2.1. Tensor $\boldsymbol{A}$ has rank one if it can be written as a tensor product[1] of vectors, i. e.,

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

where $\boldsymbol{a}_j, j = 1, \ldots, d$ are real valued vectors of length $|I_j|$.

The rank $rank(\boldsymbol{B})$ of a tensor $\boldsymbol{B}$ is the minimum number of rank-one tensors that sum up to $\boldsymbol{B}$.

The decomposition of a tensor $\boldsymbol{A}$ to tensors of rank one that sum up to $\boldsymbol{A}$ is called rank-one decomposition. In literature it is also known as canonical decomposition (CANDECOMP) [2] or parallel factor analysis (PARAFAC) [5].

A special class of tensors that appear in the problems that motivated our research in this area [13, 16] are tensors representing functions.

[1] We use the symbol $\otimes$ to denote the tensor (outer) product of two tensors. Similarly to standard multiplication of scalars the tensor product is associative.

Definition 2.2. Tensor $\boldsymbol{A}(f) : I_1 \times \ldots \times I_d \leftarrow \{0,1\}$ represents a function $f : I_1 \times \ldots \times I_{d-1} \rightarrow I_d$ if it holds for $(i_1, \ldots, i_d) \in \mathbb{I}$ that

$$\boldsymbol{A}_{i_1,\ldots,i_d}(f) \quad = \quad \begin{cases} 1 & \text{if } i_d = f(i_1, \ldots, i_{d-1}) \\ 0 & \text{otherwise.} \end{cases}$$

3. TENSORS REPRESENTING THE ℓ-OUT-OF-k FUNCTIONS

In this paper we will pay special attention to tensors representing the sum function $f(i_1, \ldots, i_{d-1}) = i_1 + \ldots + i_{d-1}$ with $I_j = \{0,1\}$ for $j = 1, \ldots, d-1$ and $I_d = \{0, 1, \ldots, d\}$. Particularly, we will deal with subtensors $\boldsymbol{A}(\ell, k)$, $k = d-1$ of these tensors obtained by restricting their dth dimension to one particular value $\ell \in \{0, 1, \ldots, k\}$:

Definition 3.1. Tensor $\boldsymbol{A}(\ell, k) : \{0,1\}^k \rightarrow \{0,1\}$ represents an ℓ-out-of-k function if it holds for $(i_1, \ldots, i_k) \in \{0,1\}^k$ that

$$\boldsymbol{A}_{i_1,\ldots,i_k}(\ell, k) \quad = \quad \delta(i_1 + \ldots + i_k, \ell) \ ,$$

where $\delta(x, y)$ is the Dirac function, i. e., $\delta(x, y) = 1$ if $x = y$, otherwise $\delta(x, y) = 0$.

Example 3.2. Tensor $\boldsymbol{A}(\ell, k)$ representing an ℓ-out-of-k function for $k = 3$ and $l = 1$:

$$\boldsymbol{A}(1,3) \quad = \quad \begin{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{pmatrix} .$$

Tensors representing ℓ-out-of-k functions have certain nice properties, e. g., they are symmetric.

Definition 3.3. Tensor $\boldsymbol{A} : \{0,1\}^k \rightarrow \mathbb{R}$ is symmetric if for $(i_1, \ldots, i_k) \in \{0,1\}^k$ it holds that

$$\boldsymbol{A}_{i_1,\ldots,i_k} \quad = \quad a_{i_1+\ldots+i_k} \ ,$$

where $\boldsymbol{a} = (a_0, \ldots, a_k)$ is a vector of real numbers.

It can be easily shown that each symmetric tensor $\boldsymbol{A} : \{0,1\}^k \rightarrow \mathbb{R}$ of rank one can be written as

$$\boldsymbol{A} \quad = \quad \begin{cases} \lambda \cdot \underbrace{(0,1) \otimes \ldots \otimes (0,1)}_{k \text{ copies}} & \text{if } A_{0,\ldots,0} = 0 \\ \\ \lambda \cdot \underbrace{(1,\alpha) \otimes \ldots \otimes (1,\alpha)}_{k \text{ copies}} & \text{otherwise,} \end{cases} \tag{1}$$

where α and λ are real numbers. Note that

$$\lim_{\alpha\to\infty} \frac{\lambda}{\alpha^k} \cdot \underbrace{(1,\alpha)\otimes\ldots\otimes(1,\alpha)}_{k \text{ copies}} \quad = \quad \lambda \cdot \underbrace{(0,1)\otimes\ldots\otimes(0,1)}_{k \text{ copies}} \quad .$$

Therefore the first outer product in formula (1) can be approximated with arbitrarily small nonzero error by the second one.

Lemma 3.4. For $\ell \in \{0,k\}$ the rank of tensors $\boldsymbol{A}(\ell,k)$ representing the respective ℓ-out-of-k function is one.

Proof. Note that

$$\begin{aligned} \boldsymbol{A}(0,k) &= \underbrace{(1,0)\otimes\ldots\otimes(1,0)}_{k \text{ copies}} \\ \boldsymbol{A}(k,k) &= \underbrace{(0,1)\otimes\ldots\otimes(0,1)}_{k \text{ copies}} \quad . \end{aligned}$$

□

If to the definition of $rank(\boldsymbol{B})$ we add the restriction that the tensors that sum up to $\boldsymbol{B}$ are symmetric we get the definition of symmetric rank.

Definition 3.5. The symmetric rank $srank(\boldsymbol{B})$ of a tensor $\boldsymbol{B}$ is the minimum number of symmetric rank-one tensors that sum up to $\boldsymbol{B}$.

Remark 3.6. It is not known whether it holds for symmetric tensors $\boldsymbol{B}$ that $rank(\boldsymbol{B}) = srank(\boldsymbol{B})$.

It follows from formula 1 that in order to find symmetric rank of tensors $\boldsymbol{A}(\ell,k)$ we need to find the minimum number of sequences of length $k+1$ of types (a) or (b) (note that the upper index of α_j is an exponent):

(a) $(\alpha_j^0,\ldots,\alpha_j^k)$, $\alpha_j \in \mathbb{R}$

(b) $(\underbrace{0,\ldots,0}_{k},1)$

such that their linear combination is equal to the sequence

$$\boldsymbol{y}(\ell,k) \quad = \quad (\underbrace{0,\ldots,0}_{\ell},1,\underbrace{0,\ldots,0}_{k-\ell}) \quad .$$

The following lemma gives a lower bound on the minimum number of such sequences.

Lemma 3.7. The minimum number of sequences of length $k+1$ of types (a) or (b) such that their linear combination is equal to the sequence $\boldsymbol{y}(\ell,k)$, $\ell < k$ is at least $\ell + 1$.

Proof. Assume $\boldsymbol{y}(\ell, k)$ is a linear combination of n sequences of type (a) and (b). Take the subsequence of $\boldsymbol{y}(\ell, k)$ consisting of its first ℓ elements, which are all zero. Then this subsequence is also a linear combination of subsequences of those n sequences consisting of their first ℓ elements. Furthermore, we can exclude from the linear combination the subsequence of the type (b) sequence (if there is any) since for $\ell < k$ all its elements are zero. Writing this in a matrix form we get $(m \leq n)$:

$$\begin{pmatrix} \alpha_1^0 & \dots & \alpha_m^0 \\ \vdots & & \vdots \\ \alpha_1^{\ell-1} & \dots & \alpha_m^{\ell-1} \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} .$$

If $m \leq \ell$ then it would imply that a Vandermonde matrix $m \times m$ is singular, which is not possible. Therefore, $n \geq m \geq \ell + 1$. □

Lemma 3.8. The minimum number of sequences of length $k+1$ of types (a) or (b) such that their linear combination is equal to the sequence $\boldsymbol{y}(\ell, k)$, $\ell > 0$ is at least $k - \ell$.

Proof. The idea of the proof is similar to the proof of Lemma 3.7. Assume $\boldsymbol{y}(\ell, k)$ is a linear combination of n sequences of type (a) and (b). Take the subsequence of $\boldsymbol{y}(\ell, k)$ consisting of its last but one $k - \ell - 1$ elemements, which are all zero. Then this subsequence is also a linear combination of subsequences of those n sequences consisting of their last but one $k - \ell - 1$ elemements. Furthermore, we can exclude from the linear combination the subsequence of the type (b) sequence (if there is any) since for $\ell < k$ all its elements are zero. Writing this in a matrix form we get $(m \leq n)$:

$$\begin{pmatrix} \alpha_1^0 & \dots & \alpha_m^0 \\ \vdots & & \vdots \\ \alpha_1^{k-\ell-2} & \dots & \alpha_m^{k-\ell-2} \end{pmatrix} \cdot \begin{pmatrix} \alpha_1^{\ell+1} \cdot \lambda_1 \\ \vdots \\ \alpha_m^{\ell+1} \cdot \lambda_m \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} .$$

If $m \leq k - \ell - 1$ then it would imply that a Vandermonde matrix $m \times m$ is singular, which is not possible. Therefore, $n \geq m \geq k - \ell$. □

A consequence of Lemma 3.7 and Lemma 3.8 is a somewhat negative result giving a quite high lower bound of the symmetric rank of tensors representing an ℓ-out-of-k function for $\ell \notin \{0, k\}$.

Theorem 3.9 (Lower bound of symmetric rank)**.** The symmetric rank of a tensor $\boldsymbol{A}(\ell, k)$, for $0 < \ell < k$ representing an ℓ-out-of-k function is at least $\max\{\ell+1, k-\ell\}$.

Proof. The theorem follows from Lemma 3.7, Lemma 3.8, and formula 1. In order to find symmetric rank of a tensor $\boldsymbol{A}(\ell, k)$ we need to find the minimum number of sequences of length $k + 1$ of types (a) or (b) such that their linear combination is equal to the sequence $\boldsymbol{y}(\ell, k)$. □

4. BORDER RANK OF TENSORS REPRESENTING THE ℓ-OUT-OF-k FUNCTIONS

In this section we describe a construction of a tensor $\boldsymbol{B}(q)$ that in limit $q \to 0$ converges to the tensor $\boldsymbol{A}(\ell, k)$ representing an ℓ-out-of-k function and has rank at most $\min\{\ell+1, k-\ell+1\}$. This, contrary to Theorem 3.9, allows construction of tensors that have relatively low rank and approximate tensor $\boldsymbol{A}(\ell, k)$ with arbitrarily small error. We start with the definition of border rank introduced in [1].

Definition 4.1 (Border rank)**.** The border rank of $\boldsymbol{A} : \{0,1\}^k \to \mathbb{R}$ is

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

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

Example 4.2. Let $k > 0$ be an even number. Then for $q > 0$

$$\boldsymbol{B}(q, \ell) \quad = \quad \frac{1}{2q^\ell} \cdot \underbrace{(1,q) \otimes \ldots \otimes (1,q)}_{k \text{ copies}} \quad - \frac{1}{2q^\ell} \cdot \underbrace{(1,-q) \otimes \ldots \otimes (1,-q)}_{k \text{ copies}}$$

has rank at most two. Since

$$\lim_{q \to 0} \boldsymbol{B}(q, 1) \quad = \quad \boldsymbol{A}(1, k)$$

tensor $\boldsymbol{A}(1, k)$ has border rank at most two. Also, since

$$\lim_{q \to \infty} \boldsymbol{B}(q, k-1) \quad = \quad \boldsymbol{A}(k-1, k)$$

tensor $\boldsymbol{A}(k-1, k)$ has border rank at most two.

Next, we will present the main theoretical result of this paper giving an upper bound on $brank(\boldsymbol{A}(\ell, k))$.

Theorem 4.3 (Upper bound of the border rank)**.** The border rank of a tensor $\boldsymbol{A}(\ell, k)$ representing an ℓ-out-of-k function is at most $\min\{\ell+1, k-\ell+1\}$.

P r o o f. For a given ℓ-out-of-k function we solve a system of linear equations

$$\boldsymbol{V}\boldsymbol{\beta} \quad = \quad \boldsymbol{v} \tag{2}$$

where vector $\boldsymbol{v} = (0, \ldots, 0, 1)^T$ has length $\ell + 1$ and $\boldsymbol{V}$ is a Vandermonde $(\ell+1) \times (\ell+1)$ matrix (note that the upper index of α_i is an exponent)

$$\boldsymbol{V} \quad = \quad \begin{pmatrix} \alpha_1^0 & \alpha_2^0 & \ldots & \alpha_{\ell+1}^0 \\ \alpha_1^1 & \alpha_2^1 & \ldots & \alpha_{\ell+1}^1 \\ \vdots & \vdots & & \vdots \\ \alpha_1^\ell & \alpha_2^\ell & \ldots & \alpha_{\ell+1}^\ell \end{pmatrix}$$

where for $i, j \in \{1, 2, \dots, \ell+1\}$ such that $j \neq i$ it holds that $\alpha_i \neq \alpha_j$. This definition of $\alpha_i, i = 1, 2, \dots, \ell+1$ guarantees that the solution $\boldsymbol{\beta}$ of system (2) exists. From $\boldsymbol{V}$ and $0 < q < 1$ we construct a $(k+1) \times (\ell+1)$ matrix $\boldsymbol{W}(q)$ defined as

$$\boldsymbol{W}(q) \quad = \quad \frac{1}{q^\ell} \begin{pmatrix} q^0 \cdot \alpha_1^0 & q^0 \cdot \alpha_2^0 & \dots & q^0 \cdot \alpha_{\ell+1}^0 \\ q^1 \cdot \alpha_1^1 & q^1 \cdot \alpha_2^1 & \dots & q^1 \cdot \alpha_{\ell+1}^1 \\ \vdots & \vdots & & \vdots \\ q^k \cdot \alpha_1^k & q^k \cdot \alpha_2^k & \dots & q^k \cdot \alpha_{\ell+1}^k \end{pmatrix} .$$

We use matrix $\boldsymbol{W}(q)$ and vector $\boldsymbol{\beta}$ to define a vector $\boldsymbol{w}(q)$ of length k as

$$\boldsymbol{w}(q) \quad = \quad \boldsymbol{W}(q)\,\boldsymbol{\beta} \ . \tag{3}$$

Note that components $w_i(q), i = 1, 2, \dots, k$ of $\boldsymbol{w}(q)$ are

$$w_i(q) \quad = \quad \begin{cases} 0 & \text{for } i \leq \ell \\ 1 & \text{for } i = \ell+1 \\ h(\alpha_1, \alpha_2, \dots, \alpha_\ell) \cdot q^{i-\ell} & \text{for } i \geq \ell+2 \end{cases}$$

where h is a function of $\alpha_1, \alpha_2, \dots, \alpha_\ell$ that is constant with respect to q. Now, we can see that

$$\lim_{q \to 0} w_i(q) \quad = \quad \begin{cases} 1 & \text{if } i = \ell+1 \\ 0 & \text{otherwise.} \end{cases}$$

Therefore for

$$\boldsymbol{B}(q) \quad = \quad \sum_{i=1}^{\ell+1} \frac{\beta_i}{q^\ell} \cdot \underbrace{(1, q \cdot \alpha_i) \otimes \ldots \otimes (1, q \cdot \alpha_i)}_{k \text{ copies}}$$

it holds that

$$\lim_{q \to 0} \boldsymbol{B}(q) \quad = \quad \boldsymbol{A}(\ell, k) \ .$$

If $\ell > \frac{k}{2}$ then we find the solution β_i' of system (2) for $\ell' = k - \ell$. From this solution we define

$$\boldsymbol{B}'(q) \quad = \quad \sum_{i=1}^{\ell'+1} \frac{\beta_i'}{q^{\ell'}} \cdot \underbrace{(q \cdot \alpha_i, 1) \otimes \ldots \otimes (q \cdot \alpha_i, 1)}_{k \text{ copies}} \ .$$

Now we get again that

$$\lim_{q \to 0} \boldsymbol{B}'(q) \quad = \quad \boldsymbol{A}(\ell, k) \ .$$

$\square$

Remark 4.4. Assume $\alpha_1 = 0$. Then for $\ell = 0$ it holds that $\boldsymbol{B}(q) = \boldsymbol{A}(0, k)$ and, similarly, for $\ell = k$ it holds that $\boldsymbol{B}'(q) = \boldsymbol{A}(k, k)$, which is in accordance with Lemma 3.4.

5. TENSORS FOR NOISY INPUTS

In the real world there is usually a noise that modifies functional relations between variables. Therefore, in this section we generalize tensors of ℓ-out-of-k functions to tensors of noisy ℓ-out-of-k functions.

Definition 5.1. Tensor $\boldsymbol{N}(\ell, k, p, q)$ represents an ℓ-out-of-k function with noisy inputs if it holds for $(i_1, \ldots, i_k) \in \{0, 1\}^k$ that

$$\boldsymbol{N}(\ell, k, p, q)_{i_1, i_2, \ldots, i_k} = \sum_{(j_1, j_2, \ldots, j_k) \in \{0,1\}^k} \boldsymbol{A}_{j_1, j_2, \ldots, j_k}(\ell, k) \cdot \prod_{n=1}^{k} \boldsymbol{M}_{i_n, j_n}(p, q) \quad , \tag{4}$$

where $A_{j_1, j_2, \ldots, j_k}(\ell, k)$ are elements of tensor $\boldsymbol{A}(\ell, k)$ representing the (exact) ℓ-out-of-k function and $M_{i_n, j_n}(p, q)$ are elements of matrix $\boldsymbol{M}(p, q)$ defined by

$$M_{i_n, j_n}(p, q) = \begin{cases} q & \text{if } j_n = 0 \text{ and } i_n = 0 \\ 1 - q & \text{if } j_n = 1 \text{ and } i_n = 0 \\ 1 - p & \text{if } j_n = 0 \text{ and } i_n = 1 \\ p & \text{if } j_n = 1 \text{ and } i_n = 1 \end{cases} \tag{5}$$

with $0 < p \leq 1$ and $0 < q \leq 1$ being the parameters of the input noise.

Remark 5.2. Note that $\boldsymbol{N}(\ell, k, 1, 1) = \boldsymbol{A}(\ell, k)$.

Example 5.3. Tensor $\boldsymbol{N}(\ell, k, p, q)$ representing an ℓ-out-of-k function with noisy inputs for $k = 3$, $l = 1$, and $p = q = 0.9$:

$$\boldsymbol{N}(1, 3, 0.9, 0.9) = \left(\begin{array}{cc} \begin{pmatrix} 0.243 \\ 0.747 \end{pmatrix} & \begin{pmatrix} 0.747 \\ 0.163 \end{pmatrix} \\ \begin{pmatrix} 0.747 \\ 0.163 \end{pmatrix} & \begin{pmatrix} 0.163 \\ 0.027 \end{pmatrix} \end{array} \right).$$

Lemma 5.4. For tensor $\boldsymbol{N}(\ell, k, p, q)$ that represents an ℓ-out-of-k function with noisy inputs it holds for $(i_1, \ldots, i_k) \in \{0, 1\}^k$ that

$$\begin{aligned} & \boldsymbol{N}(\ell, k, p, q)_{i_1, \ldots, i_k} \\ & = \sum_{i=0}^{\ell} \binom{j}{\ell - i} \cdot p^{\ell - i} \cdot (1 - p)^{j - \ell + i} \cdot \binom{k - j}{i} \cdot q^{k - j - i} \cdot (1 - q)^i \quad , \end{aligned}$$

where $j = \sum_{n=1}^{k} i_n$.

Proof. Starting with Definition 5.1, for each combination $i_1, i_2, \ldots, i_k$, we have to sum over all compatible combinations of $j_1, j_2, \ldots, j_k$. □

Remark 5.5. From Lemma 5.4 we can easily see that $\boldsymbol{N}(\ell, k, p, q)$ is a symmetric tensor.

Corollary 5.6. The border rank of a tensor $\boldsymbol{N}(\ell, k, p, q)$ representing an ℓ-out-of-k function with noisy inputs is at most $\min\{\ell + 1, k - \ell + 1\}$.

Proof. This is a direct consequence of Theorem 4.3 presented in this paper and Theorem 6 from [13]. □

6. APPROXIMATIONS OF TENSORS

Given a symmetric tensor $\boldsymbol{N}(\ell,k,p,q)$ representing an ℓ-out-of-k function with noisy inputs our goal is to find another symmetric tensor $\boldsymbol{A}$ of the same order and the same dimensions having symmetric rank at most $r = \min\{\ell+1, k-\ell+1\}$ that is a good approximation of the original tensor. We will assume all r tensors that sum up to $\boldsymbol{A}$ to be also symmetric and to have a non-zero value at the $(0,\ldots,0)$ coordinate. Then, using (1) we can write

$$\boldsymbol{A} \quad = \quad \sum_{j=1}^{r} \lambda_j \cdot \underbrace{(1,\alpha_j) \otimes \ldots \otimes (1,\alpha_j)}_{k \text{ copies}} \ , \tag{6}$$

where α_j and λ_j, $j = 1,\ldots,r$ are real numbers. We reparametrize tensor $\boldsymbol{A}$ by taking

$$\begin{aligned} \alpha_j &= \exp(\beta_j) \\ \lambda_j &= \exp(-\ell \cdot \beta_j) \cdot \mu_j \end{aligned}$$

for $j = 1,\ldots,r$. Using this we rewrite (6) as

$$\boldsymbol{A} \quad = \quad \sum_{j=1}^{r} \frac{\mu_j}{\exp(\ell \cdot \beta_j)} \cdot \underbrace{(1,\exp(\beta_j)) \otimes \ldots \otimes (1,\exp(\beta_j))}_{k \text{ copies}} \ . \tag{7}$$

Let $\boldsymbol{\mu} = (\mu_1,\ldots,\mu_r)$ and $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_r)$. Further define

$$\boldsymbol{a}(\boldsymbol{\mu},\boldsymbol{\beta}) \quad = \quad (a_0(\boldsymbol{\mu},\boldsymbol{\beta}),\ldots,a_k(\boldsymbol{\mu},\boldsymbol{\beta})) \ ,$$

where for $i = 0,\ldots,k$

$$a_i(\boldsymbol{\mu},\boldsymbol{\beta}) \quad = \quad \sum_{j=1}^{r} \mu_j (\beta_j)^i \ .$$

Then for $(i_1,\ldots,i_k) \in \{0,1\}^k$ it holds that

$$\boldsymbol{A}_{i_1,\ldots,i_k}(\boldsymbol{\mu},\boldsymbol{\beta}) \quad = \quad a_{i_1+\ldots+i_k}(\boldsymbol{\mu},\boldsymbol{\beta}) \ .$$

The criteria f we want to minimize will be defined in terms of $\boldsymbol{\mu}$ and $\boldsymbol{\beta}$ as

$$f(\boldsymbol{\mu},\boldsymbol{\beta}) \quad = \quad ||\boldsymbol{y}(\ell,k) - \boldsymbol{a}(\boldsymbol{\mu},\boldsymbol{\beta}|| \ , \tag{8}$$

where $||\cdot||$ is the Euclidean norm and $\boldsymbol{y}(\ell,k)$ was defined in Section 3 as

$$\boldsymbol{y}(\ell,k) \quad = \quad (\underbrace{0,\ldots,0}_{\ell},1,\underbrace{0,\ldots,0}_{k-\ell}) \ .$$

To find a good approximation with a small value of the above criteria we use a kind of stochastic hill climbing[2]. The algorithm iterates over two steps. In the first

[2] We tried also different gradient search methods but they either had strong tendency towards a local minima or lead to numerically unstable solutions with very large and very small numbers.

Table 1. Values of parameters.

stage s	σ_s	t_s
1	7.00	20 000
2	1.00	10 000
3	0.10	5 000
4	0.01	3 000

step we set randomly the value of $\boldsymbol{\beta}$. In the second step we find $\boldsymbol{\mu}$ that minimize (8) for $\boldsymbol{\beta}$ selected in the first step. If the value of criteria is the minimum from all values computed up to that moment we keep the values of $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$. The minimization of criteria f for a given $\boldsymbol{\beta}$ is the standard linear least squares problem that has a closed-form solution

$$\boldsymbol{\mu} \quad = \quad \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{y}(\ell, k) \ .$$

In our implementation the algorithm consisted of four stages $s = 1, 2, 3, 4$. In each stage s we sampled a vector $\boldsymbol{g}$ of r random numbers from Gaussian distribution with zero mean and standard deviation σ_s. In the first stage we selected the value $\boldsymbol{\beta} = \boldsymbol{g}$ in remaining stages $\boldsymbol{g}$ was added to currently best $\boldsymbol{\beta}$. Each stage terminated after there was no improvement of criteria f for certain number of iterations t_s. In the experiments reported in Section 7 we used the values of parameters given in Table 1.

In Figure 1 we present the decadic logarithm of the maximum distance

$$||\ |\boldsymbol{y}(\ell, k) - \boldsymbol{a}(\boldsymbol{\mu}, \boldsymbol{\beta})|\ ||_\infty$$

of the tensors $\boldsymbol{N}(\ell, k, p, q)$ representing ℓ-out-of-k functions with noisy inputs found by the algorithm described above for $k = 4, 5, \ldots, 8$ and $\ell = 1, \ldots, k-1$ from their approximations. Note that for $\ell \in \{0, k\}$ we have exact solutions of rank one — see the proof of Lemma 3.4.

7. THE GAME OF MINESWEEPER

Our main motivation for the rank-one decomposition of tensors representing ℓ-out-of-k functions with noisy inputs is the application of their rank-one decomposition in probabilistic inference – see [13] for details. To evaluate the approach presented in previous section we selected the task of computing the probability of a mine at all uncovered grid positions in the noisy modification of the computer game of Minesweeper.

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

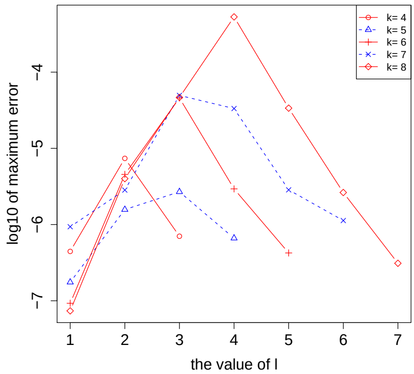

Fig. 1. Decadic logarithm of the maximum errors of the approximations for $k = 4, 5, \ldots, 8$ and $\ell = 1, \ldots, k-1$.

without clicking on any of them. In Figure 2 we present two screenshots from the game. For more information on Minesweeper see, e. g., Wikipedia [19].

7.1. A Bayesian network model for Minesweeper

Assume a game of Minesweeper on a $n \times m$ grid. The Bayesian network (BN) [6] 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[3] variables from $\mathcal{X}$ that are on the neighboring positions in the grid, i. e.

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

In the standard Minesweeper game each observed variable $Y_{i,j}$ from $\mathcal{Y}$ conveys the number of its neighbors with a mine. Therefore the number of states of $Y_{i,j}$ is the number of its parents plus one. The conditional probability table (CPT) is defined by the addition function for all combinations of states $\mathbf{x} = (x_1,\ldots,x_k)$ of its parents $pa(Y_{i,j})$ and for all states ℓ of $Y_{i,j}$ as

$$P(Y_{i,j} = \ell \mid pa(Y_{i,j}) = \mathbf{x}) \quad = \quad \left\{ \begin{array}{ll} 1 & \text{if } \ell = \sum_{u=1}^{k} x_u \\ 0 & \text{otherwise.} \end{array} \right. \tag{9}$$

[3]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.

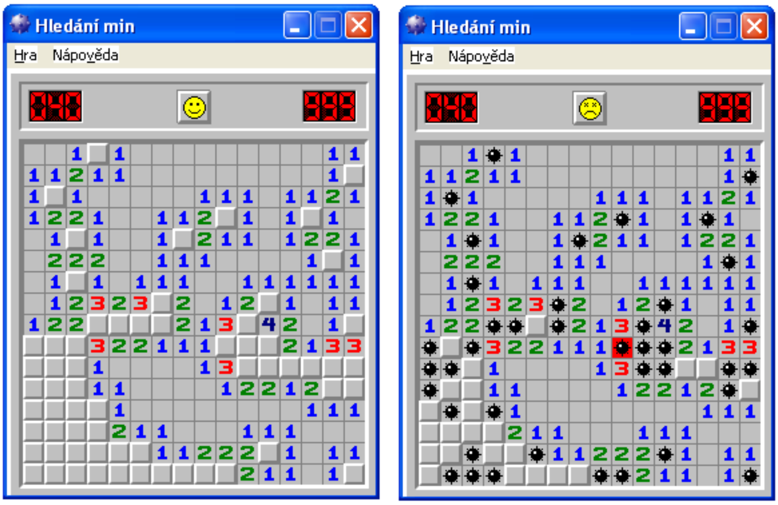

Fig. 2. Two screenshots from the game of Minesweeper. The screenshot on the right hand side is taken after the player stepped on a mine. It shows the actual position of mines.

In the noisy modification of the game we assume that there can be a noise at each argument of the sum — in the same way as it is defined by equations (4) and (5) taking $p = q = 0.9$, i. e.,

$$P(Y_{i,j} = \ell \mid pa(Y_{i,j}) = \mathbf{x}) \quad = \quad \boldsymbol{N}(\ell, k, p, q)_{x_1,\dots,x_k} \ . \tag{10}$$

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. To see this note that 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. This is a standard trick used in computations with BNs known as removing barren variables — see [6, 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 (i, j) of the grid. Each node of a graph corresponding to a grid position (i, j) will be indexed by a unique number $g = j + (i - 1) \cdot m$.

7.2. A standard approach for inference in BNs

In a standard inference technique such as [7, 8], and [9] 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 3. If $Y = \ell, \ell \in \{0, k\}$ we can propagate this evidence to each parent independently therefore we need not perform the moralization step.

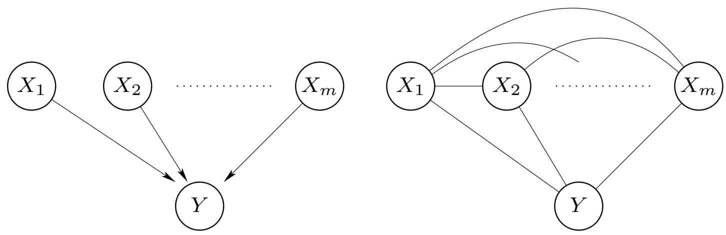

Fig. 3. The original subgraph of the DAG induced by $\{Y, X_1, \ldots, X_m\}$ and its counterpart after moralization.

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. The triangulation was performed using minweight heuristics [10]. In case of all variables being binary minweight heuristics is equivalent to the minwidth heuristics [12].

See Figure 4 and Figure 5 for an example of the moral graph and triangulated graph, respectively, after 44 observations in the game of Minesweeper on a 10×10 grid. The filled (green — if you can see it in color) circles correspond to the fields with evidence, the empty circles to the fields with an unknown status. There is a (red) cross at a circle if it corresponds to a field with mine — but note that the player and, consequently, the inference algorithm does not know where the mines are.

7.3. Tensor rank-one decomposition

In the BN model of Minesweeper the CPTs of all observations have a special local structure. For a given value ℓ of an observation its CPT correspond to a tensor representing an ℓ-out-of-k function with noisy inputs as it was defined by formula (10). Therefore, we can approximate each CPT by use of tensor rank-one decomposition proposed in Section 6. 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

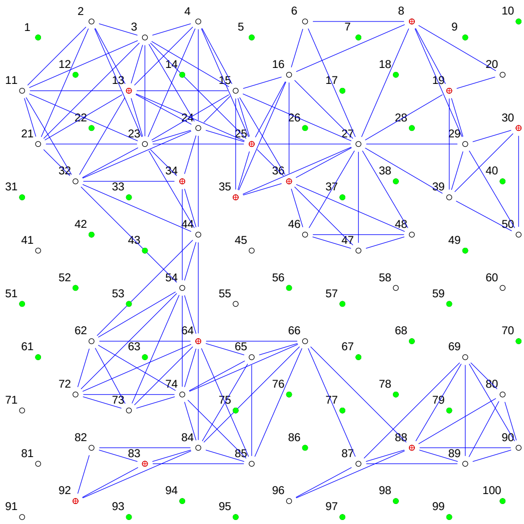

Fig. 4. An example of the moral graph of the BN after 44 observations in the game of Minesweeper on a 10×10 grid for the standard method.

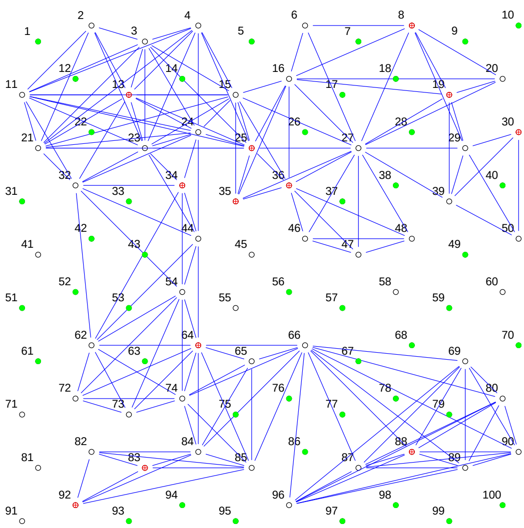

Fig. 5. The triangulated graph of the moral graph from Figure 4.

a variable B connected by undirected edges to all variables from $\{X_1, \ldots, X_m\}$, see Figure 6. The resulting graph is in [14] called the BROD graph. The number of states of B is the number of summands in (6), i. e., it is r.

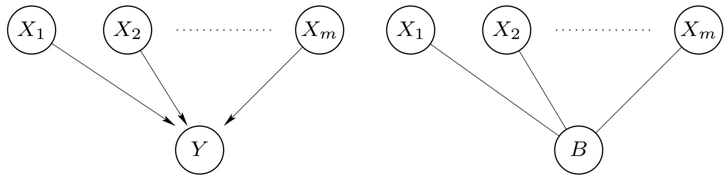

Fig. 6. The original subgraph of the DAG induced by $\{Y, X_1, \ldots, X_m\}$ and its counterpart after tensor rank-one decomposition.

Similarly, as in the standard approach, if $Y = \ell, \ell \in \{0, k\}$ we can propagate this evidence to each parent independently therefore we can remove node Y without any need for an approximation. It means that we neither include node B nor add any edges. This implies there is at most one node in the graph for the corresponding field of the grid. The second graph transformation is again triangulation, but this time applied to the undirected graph after the tensor rank-one decomposition.

See Figure 7 for an example of the graph for the model after tensor rank-one decomposition and after 44 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 remains two, for the variables corresponding to the observed fields of the game grid the number of states is r, which can be higher than two (but it is at most five). For the triangulated graph of the graph from Figure 7 see Figure 8.

8. NUMERICAL EXPERIMENTS

We performed experiments with the game of Minesweeper for the 20×20 grid size. All computations were implemented in the R language [11]. For simplicity we used a random selection of fields to be played and we assumed we never hit any of fifty mines during the game. At each of 350 steps of the game we created a Bayesian network as described in Section 7.1. We performed network transformations by:

1. the standard method consisting of moralization and triangulation steps (Section 7.2) and

2. the tensor rank-one decomposition applied to CPTs with number of parents higher than three (for CPTs with less than four parents we used the moralization) (Section 7.3) followed by the triangulation step.

In both networks we then used the lazy propagation method [9] with the computations performed with lists of tables over the junction trees. At each step of the game we recorded:

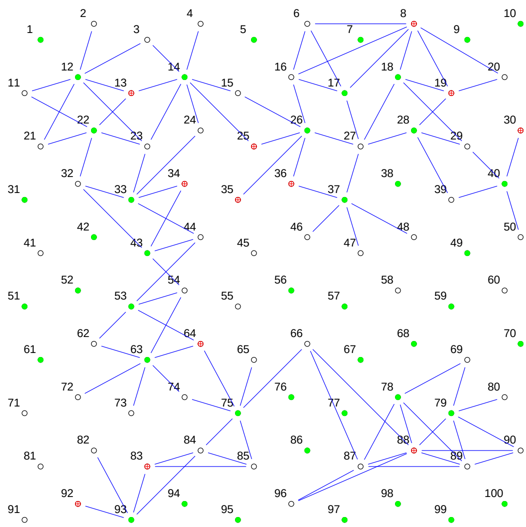

Fig. 7. The BROD graph of the BN after the same 44 observations as in Figure 4 in the game of Minesweeper on a 10×10 grid.

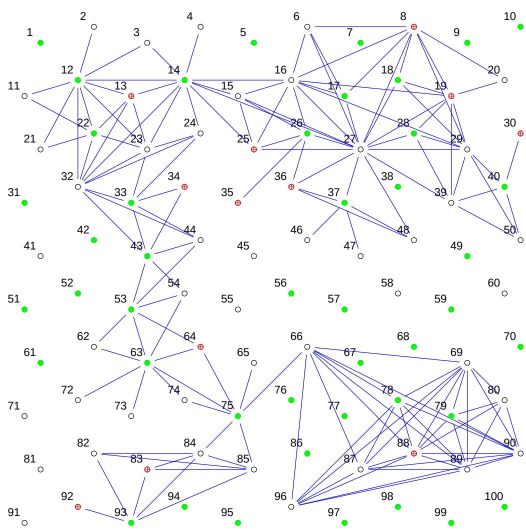

Fig. 8. Triangulated graph of the BROD graph from Figure 7.

(a) decadic logarithm of the total table size of all cliques[4] of the triangulated graphs[5],

(b) decadic logarithm of the total size of all tables before the lazy propagation was performed,

(c) decadic logarithm of the size of the largest table created during the lazy propagation,

(d) the total time (in seconds) spent during the lazy propagation,

(e) the median of the approximation error $err(i,j)$ of the probability of a mine given evidence at all unobserved fields (i,j) of the grid

$$err(i,j) \quad = \quad \left| P(X_{i,j}=0|e) - \tilde{P}(X_{i,j}=0|e) \right| \ ,$$

where P is the exact probability computed by the standard method and $\tilde{P}$ the approximate probability computed after the approximations by rank-one decomposition, and

(f) the maximum of the approximation error $err(i,j)$ over all unobserved fields (i,j) of the grid.

For the results of experiments see Figure 9. In plots (a)—(c), (e) all values represent the average over ten different games, in plot (d) the values are total sums over ten different games, and in plot (f) the values are maxima over ten different games.

From the results of experiments we can see that by using the rank-one decomposition we decrease the total table size and the size of the largest table by the order of two magnitudes. Also, we avoid high peaks in computational time. However, this gain is achieved at the expense of a certain loss of precision. From the plot (e) of Figure 9 we can see that the median of the error is not very high, but, certainly, there exist variables whose conditional probabilities given evidence were substantially different for the exact and approximate computations — see plot (f) of the same figure.

9. CONCLUSIONS

In this paper we provided a lower bound of the rank and an upper bound for the symmetric border rank of tensors representing ℓ-out-of-k functions. It is an interesting observation that they are in a way antagonistic — for ℓ with low symmetric border rank the rank is high and vice versa.

Based on our result for symmetric border rank we proposed an approximation of tensors representing ℓ-out-of-k functions with noisy inputs by r tensors of rank one, where r is an upper bound of the symmetric border rank of the approximated tensor. We applied the suggested approximation to probabilistic inference in probabilistic

[4]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.

[5]Table size of a clique is the product of numbers of states of all variables of that clique.

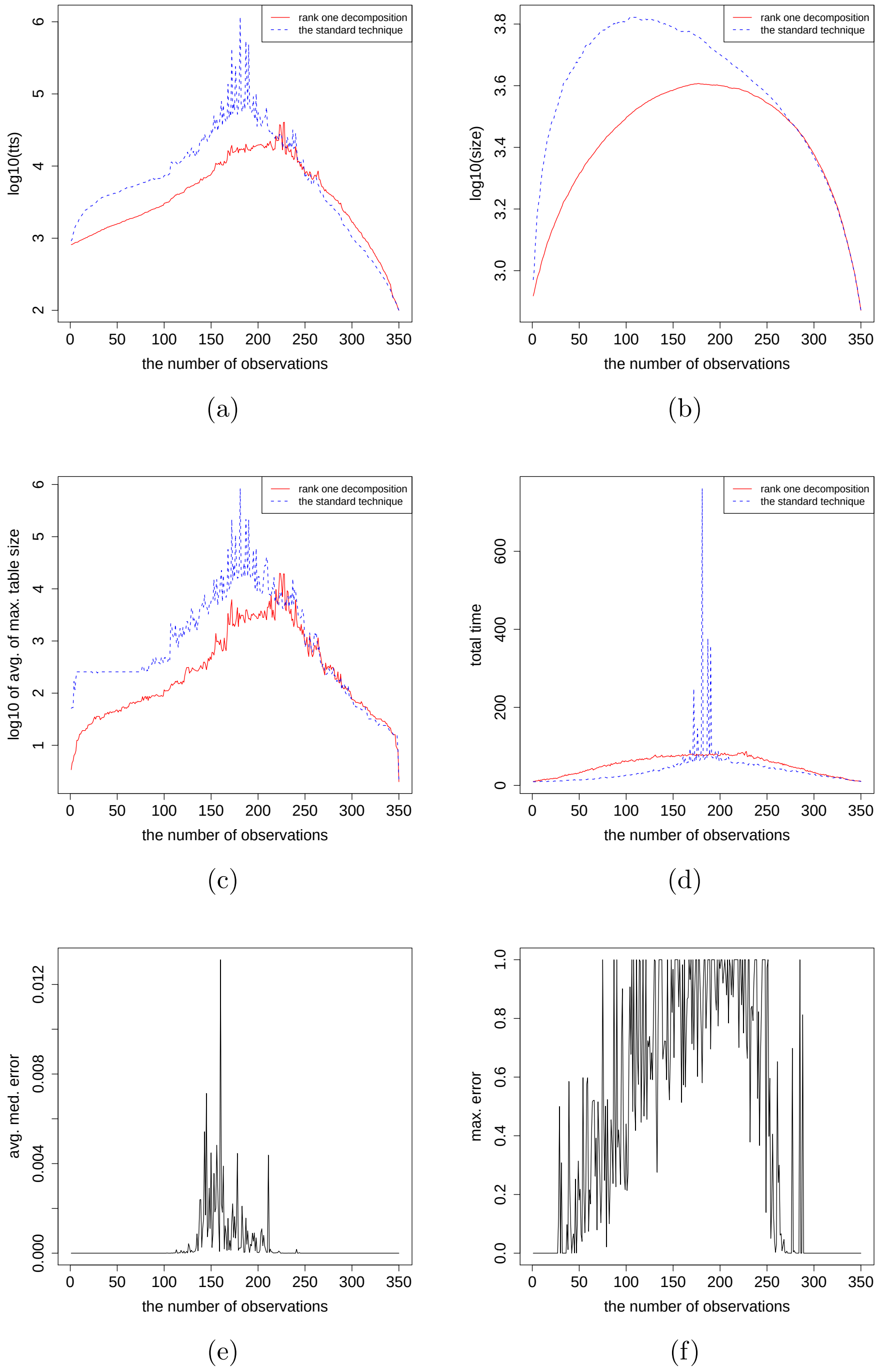

Fig. 9. Results of the experiments.

graphical models. To evaluate the proposed method we used Bayesian network models constructed for a modification of the game of Minesweeper. We believe that this model is a good example from a class of models used in real-world applications of Bayesian networks. Numerical experiments reveal that we can get a gain in the order of two magnitudes but at the expense of a certain loss of precision. A possible task for a future research is to find another type of rank-one decompositions of tensors representing ℓ-out-of-k functions that would avoid the loss of precision possibly at the expense of a lower computational gain.

ACKNOWLEDGMENT

This paper largely benefited from personal communication with Petr Savický. Namely, the idea of the proof of Lemma 3.7 is due to him.

This work was done within the project 1M0572 granted by the Ministry of Education of the Czech Republic. The author was also supported by the Grant Agency of the Czech Republic through project 201/09/1891, by the Eurocores LogICCC Project LcpR (ICC/08/E010), and by project 2C06019 granted by the Ministry of Education of the Czech Republic.

(Received August 27, 2010)

REFERENCES

[1] D. Bini and G. Lotti, and F. Romani: Approximate solutions for the bilinear form computational problem. SIAM J. Comput. *9* (1980), 692–697.

[2] J. D. Carroll and J. J. Chang: Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart–Young decomposition. Psychometrika *35* (1970), 283–319.

[3] L. De Lathauwer and B. De Moor: From matrix to tensor: multilinear algebra and signal processing. In: 4th Internat. Conf. on Mathematics in Signal Processing, Part I, Warwick 1996, IMA Conf. Series, pp. 1–11.

[4] F. J. Díez and S. F. Galán: An efficient factorization for the noisy MAX. Internat. J. Intell. Systems *18* (2003), 2, 165–177.

[5] R. A. Harshman: Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-mode factor analysis. UCLA Working Papers in Phonetics *16* (1970), 1–84.

[6] F. V. Jensen: Bayesian Networks and Decision Graphs. Springer-Verlag, New York 2001.

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

[8] S. L. Lauritzen and D. J. Spiegelhalter: Local computations with probabilities on graphical structures and their application to expert systems (with discussion). J. Roy. Statist. Soc., Ser. B *50* (1988), 157–224.

[9] A. L. Madsen and F. V. Jensen: Lazy propagation: A junction tree inference algorithm based on lazy evaluation. Artificial Intelligence *113 (1999)*, 1–2, 203–245.

[10] S. M. Olmsted: On Representing and Solving Decision Problems. PhD. Thesis, Stanford University 1983.

[11] R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna 2008.

[12] D. J. Rose: A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. Graph Theory Comput. (1972), 183–217.

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

[14] P. Savický and J. Vomlel: Triangulation heuristics for BN2O networks. In: Tenth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2009, Lecture Notes in Comput. Sci. *5590*, Springer, Berlin – Heidelberg 2009, pp. 566–577.

[15] G. R. Shafer and P. P. Shenoy: Probability propagation. Ann. Math. Artif. Intell. *2* (1990), 1–4, 327–351.

[16] J. Vomlel: Exploiting functional dependence in Bayesian network inference. In: Proc. 18th Conference on Uncertainty in AI (UAI), Morgan Kaufmann Publ. 2002, pp. 528–535.

[17] J. Vomlel and P. Savický: Arithmetic circuits of the noisy-or models. In: Prof. Fourth European Workshop on Probabilistic Graphical Models (PGM'08), Hirtshals 2005, pp. 297–304.

[18] W. Wiegerinck, B. Kappen, and W. Burgers: Bayesian Networks for Expert Systems: Theory and Practical Applications. In: Interactive Collaborative Information Systems – Studies in Computational Intelligence (R. Babuška and F. C. A. Groen, eds.), Springer-Verlag, Berlin – Heidelberg 2010, pp. 547–578.

[19] Wikipedia: Minesweeper (Computer game).
http://en.wikipedia.org/wiki/Minesweeper_(computer_game)

Jiří Vomlel, Institute of Information Theory and Automation – Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8. Czech Republic.
e-mail: vomlel@utia.cas.cz