Representations of Bayesian Networks by Low-Rank Models

Petr Tichavský TICHAVSK@UTIA.CAS.CZ
Jiří Vomlel VOMLEL@UTIA.CAS.CZ
Institute of Information Theory and Automation, Czech Academy of Sciences

Abstract

Conditional probability tables (CPTs) of discrete valued random variables may achieve high dimensions and Bayesian networks defined as the product of these CPTs may become intractable by conventional methods of BN inference because of their dimensionality. In many cases, however, these probability tables constitute tensors of relatively low rank. Such tensors can be written in the so-called Kruskal form as a sum of rank-one components. Such representation would be equivalent to adding one artificial parent to all random variables and deleting all edges between the variables. The most difficult task is to find such a representation given a set of marginals or CPTs of the random variables under consideration. In the former case, it is a problem of joint canonical polyadic (CP) decomposition of a set of tensors. The latter fitting problem can be solved in a similar manner. We apply a recently proposed alternating direction method of multipliers (ADMM), which assures that the model has a probabilistic interpretation, i.e., that all elements of all factor matrices are nonnegative. We perform experiments with several well-known Bayesian networks.

Keywords: canonical polyadic tensor decomposition; conditional probability tables; marginal probability tables; alternating direction method of multipliers.

1. Introduction

As Bayesian networks (BNs) are becoming more and more popular frameworks for reasoning under uncertainty, larger and larger BN models are constructed by domain experts or learned from extensive datasets that are common in many areas of human activities nowadays. The construction of large BNs is difficult but, often, a more critical task is an efficient reasoning within these large models.

In this paper, we study representations of whole BN models by their so-called Kruskal forms. Each Kruskal form is a sum of rank-one components. This extends our previous work in this area (Savicky and Vomlel, 2007; Vomlel and Tichavský, 2014) in the sense that in our previous works we aimed at decomposing conditional probability tables (CPTs) independently and then we combined the decomposed tables together using the standard BN inference approaches such as, for example, the Junction Tree Algorithm (Lauritzen and Spiegelhalter, 1988; Jensen et al., 1990; Shenoy and Shafer, 1990)]. The work presented in this paper is, in a sense, more ambitious since we aim at decomposing the whole BN at once. This requires new decomposition algorithms that are capable of working with the whole BN. On the other hand, if we succeed with the decomposition, we can perform reasoning much more efficiently than with the aid of the standard methods since the computational complexity is proportional to the rank of the decomposition. This is in contrast with the standard methods where the complexity is proportional to the treewidth of the triangulated moral graph of the BN. Of course, the key question is: can BNs from real applications be decomposed using reasonably low ranks (say, at most in the orders of hundreds) so that the decom-

posed form well approximates the original BNs. To answer this question, we used BNs from a BN repository (Scutari, 2009).

In this paper, we address this task using three approaches. The first approach is based on the fact that the basic building blocks of each BN are CPTs that define the joint probability so that the resulting joint probability distributions have their CPTs equal to the input CPTs. The standard way how these CPTs are combined is derived from the conditional independence (CI) relations defined by the directed acyclic graph of the BN. It is well known that the joint probability distribution satisfying the above properties (i.e., having the input CPTs and satisfying the CI relations) is the product of the input CPTs. We also aim at a joint probability distribution having CPTs equal (or at least close) to the input CPTs but we relax the CI requirements and require a low rank instead. We propose a novel method for fitting the given set of the CPT's by a low-rank CP decomposition model.

In the second approach, instead of CPTs, we use marginal probability tables (MPTs) defined by the BN for each variable by its family[1]. In this approach, we aim at a joint probability distribution having its MPTs equal (or at least close) to the input MPTs and, again, we relax the CI requirements and require a low rank instead. In this case, we apply the fitting method proposed by Kargas et al. (2017).

The third approach is similar to the second one but instead of marginals in families of model variables we use a relatively large number of marginals defined on randomly selected sets of model variables. The cardinality of these sets is restricted. In the experiments we used sets of cardinality three. A motivation for this approach is that, contrary to the second approach, by having marginals defined on sets that differ from families of model variables, we can enforce CI relations into the model (e.g., if a marginal contains variables that are conditionally independent). This can be supported by recent results (Kargas et al., 2017, Theorem 1) implying that a joint probability distribution with rank restricted by a relatively mild condition is identifiable from the MPTs of three variables.

The rest of the paper is organized as follows. In Section 2 we introduce the basic concepts. In Section 3 we present the ADMM method of minimizing the criterion (6) under the constraint that all elements of factor matrices $A_1, \ldots, A_N$ are nonnegative. Section 4 is devoted to computational experiments. We report the results of experiments performed with six large BNs from a BN repository (Scutari, 2009). In particular, we have measured the dependence of the approximation error on the rank of the resulting form. Section 5 concludes the paper.

2. Preliminaries

Let $\{X_1, \ldots, X_N\}$ be a set of discrete random variables and assume that X_n can achieve I_n values, $1, \ldots, I_n$ for all $n = 1, \ldots, N$. The probability distribution of $\{X_1, \ldots, X_N\}$ is determined by the probability table (tensor) $\mathcal{T}$ of the size $I_1 \times \ldots \times I_N$ which has elements

$$\mathcal{T}_{i_1,\ldots,i_N} \quad = \quad P(X_1 = i_1, \ldots, X_N = i_N) \ . \tag{1}$$

The number of the elements of the tensor is $I_1 I_2 \ldots I_N$ and it can obviously be very large. The integer N will be called the order of the tensor.

Assume the existence of an integer R (to be called a rank) and of N real- or complex-valued matrices $A_1, \ldots, A_N$ (to be called factor matrices) such that A_n has the size $I_n \times R$ and elements

1. Family of a variable is the variable plus its parents.

$A_{n,j,r}$, $n = 1, \ldots, N$, $j = 1, \ldots, I_n$, $r = 1, \ldots, R$, and a vector $\lambda = (\lambda_1, \ldots, \lambda_R)$ such that

$$\mathfrak{T}_{i_1,\ldots,i_N} \quad = \quad \sum_{r=1}^{R} \lambda_r A_{1,i_1,r} A_{2,i_2,r} \ldots A_{N,i_N,r} \quad . \tag{2}$$

for all $i_n = 1, \ldots, I_n$, $n = 1, \ldots, N$.

The number of elements of the factor matrices is $R(I_1 + \ldots + I_N)$. If R is small, this number might be much lower than the number of the elements in the tensor $\mathfrak{T}$. The representation of the tensor $\mathfrak{T}$ by the factor matrices $A_1, \ldots, A_N$ is called the Kruskal representation. We use the notation (Kolda and Bader, 2009)

$$\mathfrak{T} \quad = \quad [[\lambda, A_1, \ldots, A_N]] \; . \tag{3}$$

The Kruskal representation may serve as a convenient tool for storing large-dimensional tensors with many elements. The decomposition can be either general, or nonnegative. In the latter case, we assume that all factor matrices are composed of nonnegative elements only. The nonnegative Kruskal representation can be interpreted by introducing an artificial random variable Y, such that the random variables $X_1, \ldots, X_N$ are conditionally independent given Y. In this interpretation, $A_{n,j,r}$ is the probability[2] that $X_n = j$ given $Y = r$, and λ_r is the probability that $Y = r$. Note that $A_{n,j,r} \geq 0$ for all n, j, r, $\sum_j A_{n,j,r} = 1$ for all n and r, $\lambda_r \geq 0$ for all r, and $\sum_r \lambda_r = 1$. For computational purposes, we simplify this model by assuming $\lambda_1 = \ldots = \lambda_R$ and absorbing them in the factor matrices, so that they do not necessarily sum up their columns to one. We shall write simply

$$\mathfrak{T} \quad = \quad [[A_1, \ldots, A_N]] \quad . \tag{4}$$

In Bayesian networks, we are given a set of conditional probability tables, where each conditional probability table is given for a random variable given its parents. A conceptually easier case is if the given tensors represent marginal probability tables (MPTs). For each variable and its parents it is possible to compute the MPTs, so we start with this case first.

Assume we are given M marginal probability tables $\mathfrak{T}_1, \ldots, \mathfrak{T}_M$ such that each tensor $\mathfrak{T}_m$ represents a marginal probability distribution of variables $X_{i_1}, \ldots, X_{i_{N_m}}$. Then, it holds

$$\mathfrak{T}_m \quad = \quad [[\lambda, A_{i_1}, \ldots, A_{i_{N_m}}]] \quad . \tag{5}$$

This means that only factor matrices of variables from $\mathfrak{T}_m$ are needed to compute $\mathfrak{T}_m$. It is claimed by Kargas et al. (2017) that, if the true model of the network is of a low rank and the set $\mathfrak{T}_1, \ldots, \mathfrak{T}_M$ contains sufficient numbers of marginal probability tables of order at least three, then the decomposition of the whole tensor (4) is uniquely determined. The CP decomposition of the tensors $\mathfrak{T}_1, \ldots, \mathfrak{T}_M$ in (5) is joint in the sense that if two tensors $\mathfrak{T}_m$, $\mathfrak{T}_n$ share a variable $i \in S_m \cap S_n$, their CP decomposition shares the corresponding factor matrix A_i. Note that the ranks R_m of tensors $\mathfrak{T}_m$ (the minimum rank-one components in a CP decomposition of $\mathfrak{T}_m$) can be strictly lower than the rank R of the the decomposition (3). On the other hand, R must be greater than or equal to the maximum of $\{R_m\}$.

2. The matrices and the vector λ are normalized so that all matrix row sums and the vector sum are equal to one.

In practice, we do not know the ideal rank R. It would be a design variable. For a given R, we seek for the factor matrices A_n of sizes $I_n \times R$, $n = 1, \ldots, N$ such that the identities (5) are fulfilled at least approximately, by minimizing the criterion

$$\varepsilon(A_1, \ldots A_N) \quad = \quad \sum_{m=1}^{M} w_m \|\mathfrak{T}_m - [[\lambda, A_{i_1}, \ldots, A_{i_{N_m}}]]\|_F^2 \tag{6}$$

where w_m is a nonnegative weight assigned to the $m-$th tensor, and $\| \cdot \|_F$ is the Frobenius norm. The weights w_m can reflect the number of elements in $\mathfrak{T}_m$, or express a measure of confidence that we have in the tensor $\mathfrak{T}_m$[3].

Example 1 (A BN with six variables) *Assume a BN with the structure given by the directed acyclic graph presented in Figure 1 and with CPTs defined in Table 1. Using the notation defined above, we have $N = 6$. In the first approach, the input tensors correspond to CPTs. This means $\mathfrak{T}_1 = P(X_1)$, $\mathfrak{T}_2 = P(X_2|X_1)$, $\mathfrak{T}_3 = P(X_3|X_1)$, $\mathfrak{T}_4 = P(X_4)$, $\mathfrak{T}_5 = P(X_5|X_1, X_2, X_3, X_4)$, and $\mathfrak{T}_6 = P(X_5|X_1, X_3, X_4)$. The nonnegative Kruskal representation of this BN corresponds to a Naive Bayes model that has the structure given in Figure 2. The variable Y can be understood as a hidden variable whose number of states is equal to the rank of the nonnegative Kruskal representation. The task is to find matrices $A_1, \ldots, A_6$ that correspond to conditional probability tables $Q(X_1|Y), \ldots, Q(X_6|Y)$, and vector λ corresponding to $Q(Y)$.*

In order to compute a CPT, say $Q(X_6|X_1, X_3, X_4)$, from the nonnegative Kruskal representation of this BN, which corresponds to the Naive Bayes model, we need to consider only matrices $A_{i_1}, \ldots, A_{i_{N_6}}$ that for variables from the family[4] of X_6. The variables from the family of X_6 are X_1, X_3, X_4 and X_6 which means we have to consider only matrices A_1, A_3, A_4, and A_6. Therefore we compute

$$\mathfrak{T}_6 \quad = \quad Q(X_6|X_1, X_3, X_4) \quad = \quad \frac{Q(X_1, X_3, X_4, X_6)}{\sum_{X_6} Q(X_1, X_3, X_4, X_6)} \tag{7}$$

where

$$Q(X_1, X_3, X_4, X_6) \quad = \quad [[A_1, A_3, A_4, A_6]] \tag{8}$$

$$\quad = \quad \sum_{Y} Q(Y) \cdot Q(X_1|Y) \cdot Q(X_3|Y) \cdot Q(X_4|Y) \cdot Q(X_6|Y) \ . \tag{9}$$

3. Joint CP decomposition

3.1 Arbitrary Joint CP decomposition

First, we explain the alternating least squares (ALS) algorithm for the joint CP decomposition of the tensors without the non-negativity constraint.

3. If the primary goal is to approximate a BN it is desirable to monitor the error of the decomposition of the whole tensor, $\|\mathfrak{T} - [[A_1, \ldots, A_N]]\|_F$, but it is often difficult, because the number of elements of $\mathfrak{T}$ is too large to be enumerated. However, we can compute certain numbers of elements on both sides of $\mathfrak{T} \approx [[A_1, \ldots, A_N]]$ and check how they differ from each other, to check whether the approximation is good or not.
4. The remaining variables are barren, see, e.g. Jensen and Nielsen (2007).

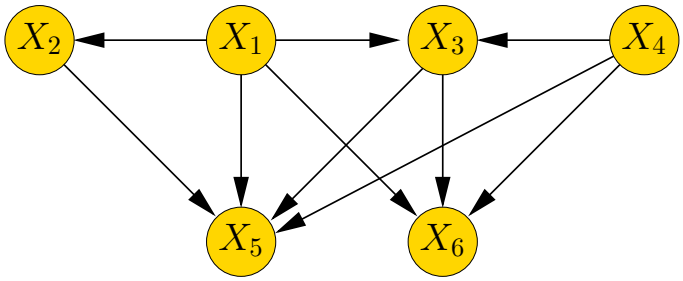

Figure 1: The structure of the BN from Example 1

Table 1: The CPTs of BN from Example 1

$P(X_1)$	
X_1	p
0	0.2
1	0.8

$P(X_2\|X_1)$		
X_2	X_1	p
0	0	0.9
1	0	0.1
0	1	0.3
1	1	0.7

$P(X_3\|X_1, X_4)$			
X_3	X_1	X_4	p
0	0	0	0.15
1	0	0	0.85
0	1	0	0.25
1	1	0	0.75
0	0	1	0.4
1	0	1	0.6
0	1	1	0.1
1	1	1	0.9

$P(X_4)$	
X_4	p
0	0.4
1	0.6

$P(X_5\|X_1, X_2, X_3, X_4 = 0)$				
X_5	X_1	X_2	X_3	p
0	0	0	0	0.9
1	0	0	0	0.1
0	1	0	0	0.09
1	1	0	0	0.91
0	0	1	0	0.18
1	0	1	0	0.82
0	1	1	0	0.018
1	1	1	0	0.982
0	0	0	1	0.27
1	0	0	1	0.73
0	1	0	1	0.027
1	1	0	1	0.973
0	0	1	1	0.054
1	0	1	1	0.946
0	1	1	1	0.0054
1	1	1	1	0.9946

$P(X_5\|X_1, X_2, X_3, X_4 = 1)$				
X_5	X_1	X_2	X_3	p
0	0	0	0	0.36
1	0	0	0	0.64
0	1	0	0	0.036
1	1	0	0	0.964
0	0	1	0	0.072
1	0	1	0	0.928
0	1	1	0	0.0072
1	1	1	0	0.9928
0	0	0	1	0.108
1	0	0	1	0.892
0	1	0	1	0.0108
1	1	0	1	0.9892
0	0	1	1	0.0216
1	0	1	1	0.9784
0	1	1	1	0.00216
1	1	1	1	0.99784

$P(X_6\|X_1, X_3, X_4)$				
X_6	X_1	X_3	X_4	p
0	0	0	0	0.75
1	0	0	0	0.25
0	1	0	0	0.2625
1	1	0	0	0.7375
0	0	1	0	0.3375
1	0	1	0	0.6625
0	1	1	0	0.118125
1	1	1	0	0.881875
0	0	0	1	0.1875
1	0	0	1	0.8125
0	1	0	1	0.065625
1	1	0	1	0.934375
0	0	1	1	0.084375
1	0	1	1	0.915625
0	1	1	1	0.029531
1	1	1	1	0.970469

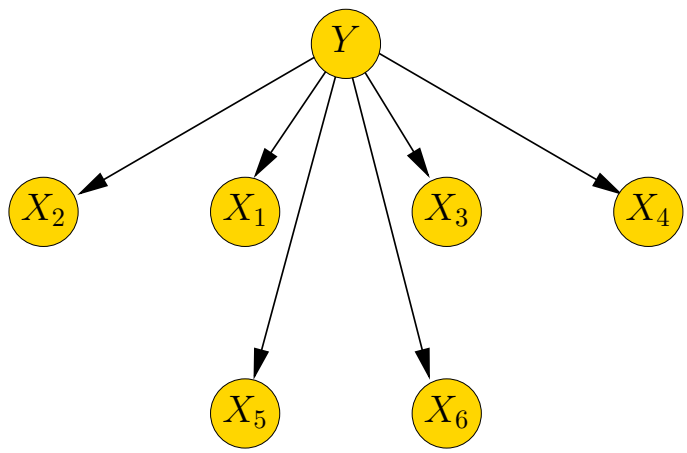

Figure 2: The structure of the Naive Bayes model that corresponds to the nonnegative Kruskal representation of BN from Example 1

The ALS method alternates the optimization of one factor matrix, say A_i, while keeping the other factor matrices fixed. The index i can be updated cyclically in a loop, or according to another policy. Assume that all other factor matrices A_j are known and fixed and derive an update formula for A_i. The iterations start with randomly chosen matrices. We can observe that criterion (6) is a quadratic function of A_i, and therefore it can be minimized in a closed form.

First of all, note that in the summation in (6), only those marginal tensors $\mathfrak{T}_m$ play a role for which $i \in S_m$. Assume that the variable i is involved in $\mathfrak{T}_m$. Let $\mathfrak{T}_m^{(i)}$ be a transposition[5] of the tensor so that the dimension i is placed at the first position, say

$$\mathfrak{T}_m^{(i)} \quad = \quad [[\lambda, A_i, A_{j_1}, \ldots, A_{j_{N_m-1}}]] \tag{10}$$

where $\{j_1, \ldots, j_{N_m-1}\} = \{i_1, \ldots, i_{N_m}\} - \{i\}$. Let $T_m^{(i)}$ be a matricization[6] of $\mathfrak{T}_m^{(i)}$ along its first dimension. It is a matrix of the size $I_i \times (\prod_k I_{j_k})$. Then, the CP decomposition of the tensor can be written as

$$T_m^{(i)} \quad = \quad A_i \mathrm{diag}(\lambda)(A_{j_{N_m-1}} \odot \ldots \odot A_{j_1})^T \quad \triangleq \quad A_i B_{im}^T \tag{11}$$

where $\odot$ is the Khatri-Rao product and ^{T} is the transposition. The Khatri-Rao product is defined for matrices having the same number of columns. For example, for two matrices F, G with R columns it is defined as

$$F \odot G \quad = \quad [F_1 \otimes G_1, \ldots, F_R \otimes G_R] \tag{12}$$

where F_r, G_r is the $r-$th column of F and G, respectively, and $\otimes$ is the Kronecker (tensor) product.

The criterion (6) can be rewritten as

$$\varepsilon(\lambda, A_1, \ldots A_N) \quad = \quad \sum_{m:i \in S_m} w_m \|T_m^{(i)} - A_i B_{im}^T\|_F^2 + \text{const} \tag{13}$$

5. In Matlab, the transposition is done through the function "permute".
6. In Matlab, the matricization is done through the function "reshape".

The closed-form minimizer of (13) with respect to A_i is then

$$A_i = \left(\sum_{m:i \in S_m} w_m T_m^{(i)} B_{im} \right) \left(\sum_{m:i \in S_m} w_m B_{im}^T B_{im} \right)^{-1} \tag{14}$$

In this way, we would update A_1, A_2, ..., A_N cyclically until convergence is achieved. The vector λ is updated in order to maintain normalization of columns of the matrices A_i.

3.2 Nonnegative Joint CP decomposition

The use of the ADMM technique for a nonnegative tensor factorization was proposed in (Liavas and Sidiropoulos, 2015). In this subsection we adapt this technique to joint CP decomposition of several tensors. Instead of minimizing the criterion $\varepsilon(A_1, \ldots A_N)$ in (6) subject to the condition $A_n \geq 0$ for $n = 1, \ldots, N$, we consider an equivalent minimization with respect to $\{A_n, \widetilde{A}_n\}$,

$$\min \varepsilon(\lambda, A_1, \ldots A_N) + \sum_{n=1}^{N} g(\widetilde{A}_n) \tag{15}$$

subject to $\widetilde{A}_n - A_n = \mathbf{0}$ for $n = 1, \ldots, N$, where

$$g(M) = \begin{cases} 0, & \text{if } M \geq \mathbf{0} \\ \infty, & \text{otherwise .} \end{cases} \tag{16}$$

We introduce dual variables Y_n of size $I_n \times R$ for $n = 1, \ldots, N$ and the vector of penalty terms $\rho = [\rho_1, \ldots, \rho_N]$. Then, the ADMM optimization method iterates

$$\{A_m^{k+1}\} = \text{argmin}_{\{A_m\}} \varepsilon(A_1, \ldots A_N) + \sum_{n=1}^{N} Y_n * A_n + \frac{\rho_n}{2} \|A_n - \widetilde{A}_n^k\|^2 \tag{17}$$

$$\widetilde{A}_m^{k+1} = \left(A_m^{k+1} + \frac{1}{\rho_m} Y_m \right)_+ \tag{18}$$

$$Y_m^{k+1} = Y_m^k + \rho_m \left(A_m^{k+1} - \widetilde{A}_m^{k+1} \right) . \tag{19}$$

where k is the iteration index, $Y_n * A_n$ is the scalar product of Y_n and A_n, and $(X)_+$ is the nonnegative part of X. The minimization in (17) is, indeed, not quadratic, and is replaced by a series of updates

$$A_i^{k+1} = \left(\sum_{m:i \in S_m} w_m T_m^{(i)} B_{im} + \rho_i \widetilde{A}_i^k - Y_i^k \right) \left(\sum_{m:i \in S_m} w_m B_{im}^T B_{im} + \rho_i I_R \right)^{-1} \tag{20}$$

for $i = 1, \ldots, N$, where B_{im} is computed from the latest available estimates of A_j, $j \neq i$, and I_R is the $R \times R$ identity matrix. For simplicity, we skip technical details of the derivation of these updates.

During the iteration process, $k \to \infty$, the Frobenius norm of the difference $\|A_n^k - \widetilde{A}_n^k\|$ should converge to zero for all $n = 1, \ldots, N$. We also found it useful to monitor the fitting error for all input tensors

$$e_m(\lambda, A_1, \ldots, A_N) = \|\mathfrak{T}_m - [[\lambda, A_{i_1}, \ldots, A_{i_{N_m}}]]\|_F . \tag{21}$$

3.3 Fitting Conditional Probability Tables

Assume now that the given set of tensors $\{\mathfrak{T}_m\}$ does not represent marginal probability tables of $X_{i_1}, \ldots, X_{i_m}$ but conditional probability tables of X_{i_1} given $X_{i_2}, \ldots, X_{i_m}$. Instead of the errors in (21) we shall consider the errors

$$\tilde{e}_m \quad = \quad \|[[\lambda, A_{i_1}, \ldots, A_{i_{N_m}}]] - \mathfrak{T}_m \star [[\lambda, \mathbf{1}_{I_1,R}, A_{i_2} \ldots, A_{i_{N_m}}]]\|_F \tag{22}$$

where $\star$ is the elementwise product and $\mathbf{1}_{I_1,R}$ is a matrix of the size $I_1 \times R$ filled with ones, and I_1 is the number of states of X_{i_1}. We note that $(\tilde{e}_m)^2$ is a quadratic function of $A_{i_1}, \ldots, A_{i_{N_m}}$. However, we need to avoid a trivial solution when some of the factor matrices A_{i_j} is zero. It can be done in several ways. One of them is to add the constraint that the sum of the elements of $[[\lambda, A_{i_1}, \ldots, A_{i_{N_m}}]]$ should be one,

$$\|[[\lambda, A_{i_1}, \ldots, A_{i_{N_m}}]]\|_1 \quad = \quad 1 \ , \tag{23}$$

where $\|\cdot\|_1$ is the L_1 norm. The total criterion to replace (6) is

$$\tilde{\varepsilon}(\lambda, A_1, \ldots, A_N) \quad = \quad \sum_{m=1}^{M} w_m[((\tilde{e}_m)^2 + (\|[[\lambda, A_{i_1}, \ldots, A_{i_{N_m}}]]\|_1 - 1)^2] \ . \tag{24}$$

The criterion is quadratic like the former one, and it can be optimized by an ADMM alternating least square technique similar to that described in the previous subsection.

Example 2 (Nonnegative Kruskal form of rank three for a BN with six variables) *Assume again the BN from Example 1 whose structure is given in Figure 1. Now, we will use the ADMM technique for the nonnegative factorization of tensor $\mathfrak{T} = P(X_1, \ldots, X_6)$ with the input tensors $\mathfrak{T}_1, \ldots, \mathfrak{T}_6$ fulfilling one of the following conditions:*

- *CPTs of the BN (Section 3.3),*
- *MPTs computed from the BN for the families of all variables (Section 3.2),*
- *all MPTs of cardinality three computed from the BN. There are twenty different subsets of cardinality three in this example, or*
- *the single tensor of the joint probability table of the size $2 \times 2 \times 2 \times 2 \times 2 \times 2$.*

We will consider Kruskal forms of rank three in this example. We aim at minimizing the criterion (6) *or* (24)*, respectively, with uniform weights, $w_1 = \ldots = w_M = 1$. We run the algorithm several times from different starting points and let the algorithm run for 1000 iterations.*

In Table 3 we present few statistics from the experiment. First, we present the approximation error(6) *or*(13)*, respectively, achieved in the optimization. We can see that the approximation error can be quite low even at a small rank, $R = 3$. Second, the sum of absolute errors of each approximation with respect to the tensor of the original Bayesian network model is presented. Third, we report the Kendall correlation coefficient between the tensor of the original Bayesian network (arranged in one long vector) and the approximation. We can see that although the fitting error is quite low even for the low model rank, the difference from the original tensor is quite large, both in terms of the sum of absolute errors and also in the terms of the Kendall correlation coefficient.*

Table 2: Statistics of approximations of BN from Example 1

	fitting error	overall error	Kendall correlation
CPT's	5.047e-4	0.4367	0.5010
MPT's of the same size	1.735e-4	0.1003	0.7579
MPT's of the size 3	6.0e-3	0.2005	0.6781
The whole distribution	4.62e-4	0.0923	0.7520

Table 3: The CPTs of the Naive Bayes model

$P(X_1\|Y)$		
X_1	Y	p
0	0	0
1	0	1
0	1	0
1	1	1
0	2	0.8853
1	2	0.1147

$P(X_2\|Y)$		
X_2	Y	p
0	0	0.2595
1	0	0.7405
0	1	0.2876
1	1	0.7124
0	2	0.8976
1	2	0.1024

$P(X_3\|Y)$		
X_3	Y	p
0	0	0.4203
1	0	0.5797
0	1	0.0609
1	1	0.9391
0	2	0.3024
1	2	0.6976

$P(X_4\|Y)$		
X_4	Y	p
0	0	0.7738
1	0	0.2262
0	1	0.2622
1	1	0.7378
0	2	0.4126
1	2	0.5874

$P(X_5\|Y)$		
X_5	Y	p
0	0	0.0157
1	0	0.9843
0	1	0
1	1	1
0	2	0.2151
1	2	0.7849

$P(X_6\|Y)$		
X_6	Y	p
0	0	0.2832
1	0	0.7168
0	1	0.0069
1	1	0.9931
0	2	0.1991
1	2	0.8009

$P(X_1)$	
X_1	p
0	0.2049
1	0.5700
2	0.2251

The best approximation is achieved by fitting the whole distribution and then by fitting the marginal probability tables of the nodes' families within the Bayesian model.

We have also tried to approximate the whole probability table with tensors of a higher rank. For rank $R = 10$ we receive the Kendall correlation coefficient value of 0.9266. It looks like this Bayesian network is not easy to be represented by a low-rank model.

If we compare the total table size of the junction tree of the original model, which is 48, with the total tables size of the Naive Bayes model (representing the nonnegative Kruskal form), which is 36, we get a 25% saving in the total table size. This saving can help us reduce the complexity of inference, which is proportional to the total table size.

4. Numerical Experiments

For the numerical experiments, we have selected the method that uses MPTs computed from the BN for the families of each variable, since this method provides the best fit of the original BNs. We let the algorithm iterate until the convergence is reached, which sometimes requires a number of iterations in the order of tens of thousands. We have also started the algorithm from several different starting points.

In Figure 3 we present results of the numerical experiments on six large Bayesian networks from a Bayesian network repository (Scutari, 2009). In Table 4 we summarize the basic characteristics of these BNs[7]. Each point in the graph corresponds to a tensor decomposition of a Bayesian

7. We report the total table size of the junction tree representation computed by Hugin (Hugin, 2015) using its (in most cases optimal) triangulation.

Table 4: Characteristics of BNs used in the experiments

name	number of variables	total table size
Pigs	441	709,830
Hepar 2	70	2,617
Pathfinder	109	182,641
Link	724	37,870,762
Insurance	27	46,872
Diabetes	413	9,989,707

network. Its horizontal coordinate is the compression ratio of the total table size (tts) of the tensor decomposition with respect to the optimal tts of the Bayesian network computed by (Hugin, 2015). A value smaller than 1.0 means we get a saving in tts. The horizontal coordinate corresponds to the average absolute error of the tensor decomposition computed for all values of marginal probability tables and input marginal probability tables of the original Bayesian network. The lower the value the better approximation we get. Each point is labeled by the value of the rank of the corresponding tensor decomposition.

In Figure 3 we can see that we can control the approximation error by the rank. For some networks (Pathfinder, Diabetes) we get a low error rate (0.0002 and 0.0004, respectively) with a favorable compression ratio (0.03 and 0.01, respectively). For some other networks (Pigs, Link, and Insurance) the error rate is higher (0.003, 0.003, and 0.0015, respectively) but one of them (Link) has a very high compression (the ratio is about 0.001). The Hepar 2 network has a low tts already for the original Bayesian network (tts=2617), which is the main reason that it is hard to get any higher compression rate with a small error rate using the suggested tensor decomposition.

5. Conclusions

In our experiments, we have observed that the marginal probability tables of all six tested models can be approximated by a low-rank Kruskal model with a quite low average error. Similarly, we can approximate the corresponding CPTs. However, already in the example of six binary variables we have observed that the obtained low-rank models differ from each other. They also differ from the Bayesian network model in the sense that their joint probability distributions differ, which implies they give different inference results. At first sight, this may be a surprise since the MPTs and CPTs are fitted well. But when we realize that BNs and low rank models combine the input CPTs and MPTs differently, we see there is no reason why they should arrive at the same joint probability distributions.

Indeed, it should be possible to combine information from the CPTs and from the marginals or introduce more sophisticated weighting and do many more experiments. It seems that the low-rank models cannot compete with the Bayesian product models if the latter model is considered to be the ground truth. However, if the ground truth is not available and CPTs need to be estimated from data, then the low-rank models can be appropriate. At least the low-rank models have the advantage of a simple inference.

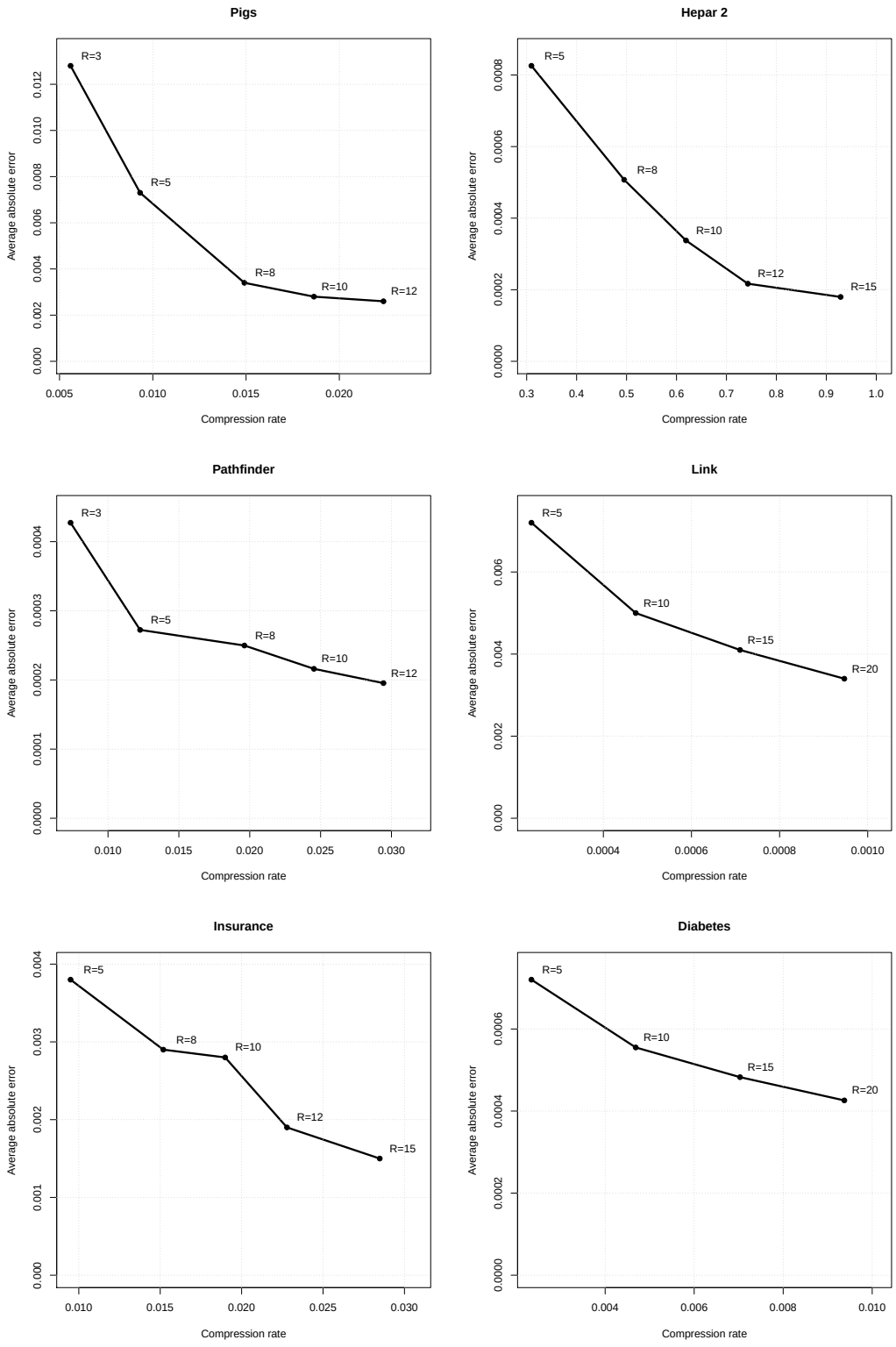

Figure 3: Results of experiments

It would be interesting to compare BN models with low-rank Kruskal models when both were learned from real data, e.g., data from a machine learning repository. If they were both learned from the same training dataset and tested on a testing dataset, we could see which model better represents data. But we leave this as a topic for our future research.

Acknowledgements

This work was supported by the Czech Science Foundation (Project 17-00902S).

References

Hugin. API Reference Manual, version 8.2, 2015. http://www.hugin.com/.

F. Jensen and T. Nielsen. *Bayesian Networks and Decision Graphs*. Springer, 2nd edition edition, 2007.

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.

N. Kargas, N. D. Sidiropoulos, and X. Fu. Tensors, learning, and 'Kolmogorov extension' for finite-alphabet random vectors, 2017. arXiv:1712.00205 [eess.SP].

T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51:455–500, 2009.

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.

A. P. Liavas and N. D. Sidiropoulos. Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 63(20): 5450–5463, Oct 2015. ISSN 1053-587X. http://dx.doi.org/10.1109/TSP.2015.2454476.

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

M. Scutari. Bayesian Network Repository, 2009. http://www.bnlearn.com/bnrepository/.

P. P. Shenoy and G. Shafer. Axioms for probability and belief-function propagation. In R. D. Shachter, T. S. Lewitt, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, volume 9 of *Machine Intelligence and Pattern Recognition*, pages 169 – 198. North-Holland, 1990. https://doi.org/10.1016/B978-0-444-88650-7.50019-6.

J. Vomlel and P. Tichavský. Probabilistic inference with noisy-threshold models based on a CP tensor decomposition. *International Journal of Approximate Reasoning*, 55:1072—1092, 2014. http://dx.doi.org/10.1016/j.ijar.2013.12.002.