

DECOMPOSITION OF PROBABILITY TABLES REPRESENTING BOOLEAN FUNCTIONS

Jiří Vomlel

Institute of Information Theory and Automation
Academy of Science of the Czech Republic
vomlel@utia.cas.cz

Abstract

We apply tensor rank-one decomposition (Savicky and Vomlel, 2005) to conditional probability tables representing Boolean functions. We present a numerical algorithm that can be used to find a minimal tensor rank-one decomposition together with the results of the experiments performed using the proposed algorithm.

We will pay special attention to a family of Boolean functions that are common in probabilistic models from practice - monotone and symmetric Boolean functions. We will show that these functions can be better decomposed than general Boolean functions, specifically, rank of their corresponding tensor is lower than average rank of a general Boolean function.

1 Introduction

Probabilistic graphical models are a powerful family of models for reasoning under uncertainty. In these models conditional independence relations between model variables are exploited so that the computations can be performed efficiently. Diverse methods were developed for efficient computations in these models. Probably the best known is in the junction tree propagation (Jensen et al., 1990).

The efficiency of the computations can be further improved if internal structure of conditional probability tables (CPTs) is taken into account. Again, different techniques for exploiting the internal structure were proposed. Some of these techniques concentrate on special class of models, for example, independence of causal influence (or causal independence) models (Heckerman and Breese, 1994; Takikawa and D'Ambrosio, 1999; Zhang and Poole, 1996; Lucas, 2005), context specific independence (Boutilier et al., 1996), logical constraints (Darwiche, 2002), and functional dependence (Vomlel, 2002).

In this paper we apply the tensor rank-one decomposition (Savicky and Vomlel, 2005) to CPTs representing Boolean functions. The basic idea is to decompose a probability table into a series of tables, such that the table that is

the sum of the series is equal to the original table. Each table in the series has the same domain as the original table but can be expressed as a product of one-dimensional tables. Entries in tables are allowed to be any real number, i.e., they can be also negative numbers. The possibility of having negative numbers, in contrary to a multiplicative decomposition, opens new possibilities for compact representation of probability tables. In order to have the decomposition as compact as possible we search for a shortest series.

It is convenient to formally specify the task using the tensor terminology. Assume variables X_i , $i \in N \subset \mathbb{N}$ each variable X_i taking values (a value of X_i will be denoted x_i) from a finite set \mathcal{X}_i . Let for any $A \subseteq N$ the symbol x_A denotes a vector of the values $(x_i)_{i \in A}$, where for all $i \in A$: x_i is a value from \mathcal{X}_i .

Definition 1 Tensor

Let $A \subset N$. *Tensor* ψ over A is a mapping

$$\times_{i \in A} \mathcal{X}_i \mapsto \mathbb{R} .$$

The cardinality $|A|$ of the set A is called *tensor dimension*.

Note that every probability table can be looked upon as a tensor. Tensor ψ over A is an (unconditional) probability table if for every x^A it holds that $0 \leq \psi(x^A) \leq 1$ and $\sum_{x^A} \psi(x^A) = 1$. Tensor ψ is a conditional probability table (CPT) if for every x^A it holds that $0 \leq \psi(x^A) \leq 1$ and if there exists $B \subset A$ such that for every x_B it holds $\sum_{x_{A \setminus B}} \psi(x_B, x_{A \setminus B}) = 1$.

Next, we will recall the basic tensor notion. If $|A| = 1$ then tensor is a vector. If $|A| = 2$ then tensor is a matrix. The *outer product* $\psi \otimes \varphi$ of two tensors $\psi : \times_{i \in A} \mathcal{X}_i \mapsto \mathbb{R}$ and $\varphi : \times_{i \in B} \mathcal{X}_i \mapsto \mathbb{R}$, $A \cap B = \emptyset$ is a tensor $\xi : \times_{i \in A \cup B} \mathcal{X}_i \mapsto \mathbb{R}$ defined for all $x_{A \cup B}$ as

$$\xi(x_{A \cup B}) = \psi(x_A) \cdot \varphi(x_B) .$$

Now, let ψ and φ are defined on the same domain $\times_{i \in A} \mathcal{X}_i$. The sum $\psi + \varphi$ of two tensors is tensor $\xi : \times_{i \in A} \mathcal{X}_i \mapsto \mathbb{R}$ such that

$$\xi(x_A) = \psi(x_A) + \varphi(x_A) .$$

Definition 2 Tensor rank (Håstad, 1990)

Tensor of dimension $|A|$ has *rank* one if it is an outer product of $|A|$ vectors. *Rank* of tensor ψ is the minimal number of tensors of rank one that sum to ψ . *Rank* of tensor ψ will be denoted as $rank(\psi)$.

Remark Note that standard matrix rank is a special case of tensor rank (for $|A| = 2$).

Now, we are ready to formalize the task of decomposition of a probability table into a shortest series of tables that are product of one-dimensional tables.

Definition 3 Tensor rank-one decomposition

Assume a tensor ψ over A . A series of tensors $\{\varrho_b\}_{b=1}^r$ such that

- for $b = 1, \dots, r$: $\text{rank}(\varrho_b) = 1$, i.e.,

$$\varrho_b = \otimes_{i \in A} \varphi_{b,i} ,$$

where $\varphi_{b,i}, i \in A$ are vectors and

- $\psi = \sum_{b=1}^r \varrho_b$

is called *tensor rank-one decomposition* of ψ .

Note that from the definition of tensor rank it follows that for $r \geq \text{rank}(\psi)$ such a series always exists. The decomposition is minimal if there is no shorter series satisfying two conditions of Definition 3.

Example 1 Let $\psi : \{0, 1\} \times \{0, 1\} \times \{0, 1\} \mapsto \mathbb{R}$ be

$$\begin{pmatrix} (1, 2)^T & (2, 4)^T \\ (2, 4)^T & (4, 9)^T \end{pmatrix} .$$

This tensor has rank two since

$$\psi = (1, 2) \otimes (1, 2) \otimes (1, 2) + (0, 1) \otimes (0, 1) \otimes (0, 1)$$

and there are no three vectors whose outer product is equal to ψ . \diamond

2 Symmetric Boolean functions

We will pay a special attention to a family of Boolean functions that are common in probabilistic models from practice - monotone and symmetric Boolean functions. In Section 4 we will see that Boolean functions from this family can be better decomposed than general Boolean functions, i.e. the rank of their corresponding tensors is lower than the average rank of a general Boolean function.

Let $\mathcal{X} = \{0, 1\}^m$ and x be a vector from \mathcal{X} . By $|x|$ we will denote the number of non-zero values of vector x . We define ordering of elements of \mathcal{X} by a binary relation \leq defined for every $x, y \in \mathcal{X}$ as $x \leq y$ if $|x| \leq |y|$.

Definition 4 Boolean function $f : \mathcal{X} \mapsto \{0, 1\}$ is monotone, if for all $x, y \in \mathcal{X}$ it holds that if $x \leq y$ then $f(x) \leq f(y)$.

Remark In the literature there exist different definitions of monotonicity due to the fact that one may define the binary relation \leq differently. For example, \leq can be defined for all $x = (x_1, \dots, x_m) \in \mathcal{X}$ and $y = (y_1, \dots, y_m) \in \mathcal{X}$ as: $x \leq y$ if $i \in \{1, \dots, m\} : x_i \leq y_i$, which is called the canonical ordering (Wegener, 1987).

Definition 5 Boolean function $f : \mathcal{X} \mapsto \{0, 1\}$ is symmetric, if for all $x \in \mathcal{X}$: $f(x) = c_{|x|}$.

Definition 6 Boolean threshold function is a Boolean function f such that

$$f(x) = \begin{cases} 0 & \text{if } |x| < t \\ 1 & \text{if } |x| \geq t. \end{cases}$$

where $t \in \{0, \dots, m+1\}$ is the threshold value.

Note that if $t = 0$ then it corresponds to the *true* function, if $t = m+1$ it is the *false* function, if $t = 1$ then it is the *or* function, and if $t = m$ then it is the *and* function.

Proposition 1 Boolean function f is monotone and symmetric if and only if it is a Boolean threshold function.

Proof

(1) monotonicity & symmetry \Rightarrow threshold

If $\forall x \in \mathcal{X} : f(x) = 0$ or if $\forall x \in \mathcal{X} : f(x) = 1$ then f corresponds to the threshold function with $t = m+1$ and $t = 0$, respectively. Otherwise, for any symmetric monotone Boolean function we can find $x_t \in \mathcal{X}$ such that

$$f(x) = \begin{cases} 0 & \text{if } x < x_t \\ 1 & \text{if } x \geq x_t. \end{cases}$$

Since $x < x_t \iff |x| < |x_t|$ and $x \geq x_t \iff |x| \geq |x_t|$ we can define $t = |x_t|$. Observe that the Boolean threshold function with this threshold is equivalent to the original Boolean function f .

(2) threshold \Rightarrow monotonicity & symmetry

It is easy to check that every threshold function is symmetric and monotone. \square

Note that not all monotone Boolean function are symmetric. Next we present an example of such a function.

Example 2 Let f_{t+1} be the Boolean threshold function with threshold $t+1$ and f_{t+2} be the Boolean threshold function with threshold $t+2$. Then for any $t \in \{1, m-1\}$ the function $g_t(x) = f_{t+1}(x) - f_{t+2}(x)$ is an example of symmetric function that is not monotone. Since this function indicates whether exactly t values of x are non-zero we will refer to this function as exactly- t function. \diamond

3 Numerical algorithm

Definition 7 Tensor rank-one approximation

Assume a tensor ψ and an integer $s \geq 1$. A *tensor rank-one approximation* of length s is a series $\{\varrho_b\}_{b=1}^s$ of rank-one tensors ϱ_b that is a tensor rank-one decomposition of a tensor $\hat{\psi}$ with $\text{rank}(\hat{\psi}) = s$.

If $\hat{\psi}$ minimizes $\sum_x (\psi(x) - \hat{\psi}(x))^2$ we say that it is a *best tensor rank-one approximation of length s* .

Note that if $s = \text{rank}(\psi)$ then the minimal value of $\sum_x (\psi(x) - \hat{\psi}(x))^2$ is zero and a best tensor rank-one approximation of length s is also a minimal tensor rank-one decomposition of ψ . Therefore, we can search numerically for a minimal tensor rank-one decomposition by solving the task from Definition 7 starting with $s = 1$ and then incrementing s by one until $\sum_x (\psi(x) - \hat{\psi}(x))^2$ is sufficiently close to zero.

We performed tests with several gradient methods. The best performance was achieved with Polak-Ribière conjugate gradient method that used the Newton method in one dimension. We performed initial experiments for tensors corresponding to the exclusive-or and or functions of two parent binary variables. For these functions we know the tensor rank is two (Savicky and Vomlel, 2005) therefore we could verify whether for $s = 2$ the algorithm found a tensor rank-one decomposition of these tensors.

The initial values of the algorithm were random numbers from interval $[-1, +1]$. We started the algorithm from ten different starting points to avoid getting stuck in a local minima. Figures 1 and 2 illustrate the convergence using three sample runs. The displayed values are values of $\sum_x (\psi(x) - \hat{\psi}^{(i)}(x))^2, i = 1, 2, \dots$ as they change with the progress of the algorithm. We can see rapid convergence to the global minima in this example.

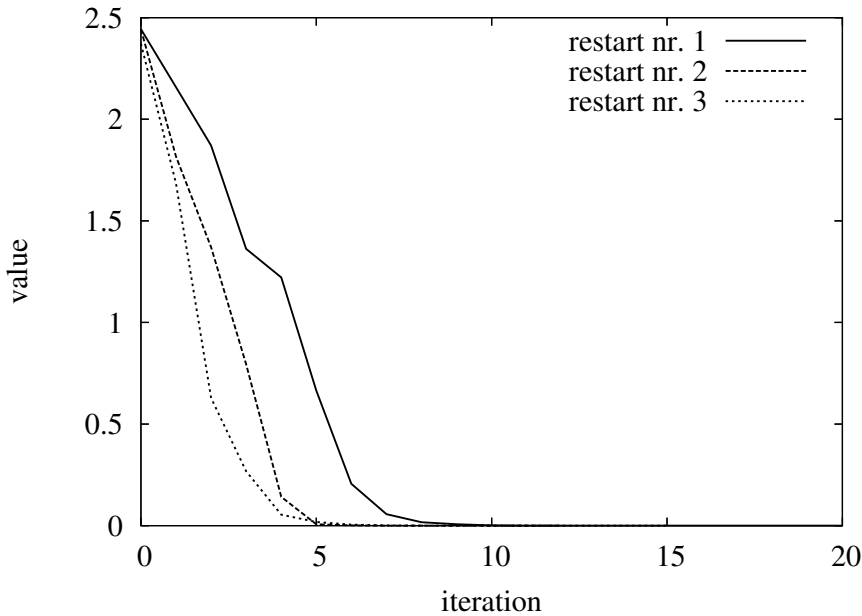


Figure 1: The values $\sum_x (\psi(x) - \hat{\psi}^{(i)}(x))^2$ for the xor function.

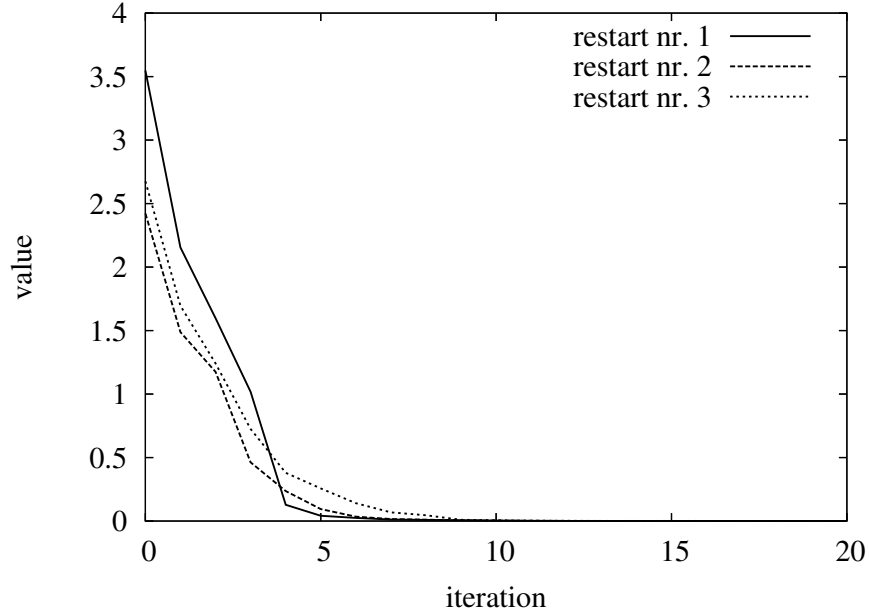


Figure 2: The values $\sum_x (\psi(x) - \hat{\psi}^{(i)}(x))^2$ for the or function.

4 Results

The final experiments aimed at a numerical computation of rank of tensors corresponding to the threshold function, the exactly- t function, and randomly generated general Boolean functions.

We used the Polak-Ribière conjugate gradient method described in Section 3. For each $s = 1, 2, \dots$ we searched for the best tensor rank-one approximation of length s until we found an approximation that was sufficiently closed to the tensor of a given Boolean function (we again started the algorithm from ten different randomly generated starting points). As a stopping criteria we used the condition

$$\sum_x (\psi(x) - \hat{\psi}^{(i)}(x))^2 < 10^{-5} .$$

The lowest value of s for which the above condition was met we regarded to be the rank of the tensor of the given Boolean function.

In Table 1 we compare the rank of the threshold Boolean functions with threshold t compared with the average rank and the rank interval of randomly generated Boolean functions. We performed the comparisons for different number of variables m . We computed average tensor rank value of fifty randomly generated Boolean functions. The values in brackets define the interval containing all rank values. Note that this does not mean that there are no Boolean

functions with lower or higher rank, it only means that they were not present in our random sample. In Table 2 we present similar comparisons for the exactly- t function.

Table 1: Rank of the threshold Boolean function with threshold t compared with the average rank and the rank interval of randomly generated Boolean functions (rnd).

t	0	1	2	3	4	5	6	rnd
$m = 2$	1	2	2	1	1	1	1	1.92 [1, 2]
$m = 3$	1	2	3	2	1	1	1	2.78 [2, 3]
$m = 4$	1	2	3	3	2	1	1	3.94 [3, 5]
$m = 5$	1	2	3	4	3	2	1	6.62 [5, 8]

Table 2: Rank of the exactly- t function compared with the average rank and the rank interval of randomly generated Boolean functions (rnd).

t	0	1	2	3	4	5	6	rnd
$m = 2$	2	2	2	1	1	1	1	1.92 [1, 2]
$m = 3$	2	3	3	2	1	1	1	2.78 [2, 3]
$m = 4$	2	3	4	3	2	1	1	3.94 [3, 5]
$m = 5$	2	3	4	4	3	2	1	6.62 [5, 8]

From the tables we can see that the rank of the threshold Boolean functions (for $m > 3$) and also the exactly- t Boolean functions (for $m > 4$) is lower than rank of a randomly generated Boolean functions. The lower the rank the more compact is the tensor rank-one decomposition (as defined in Section 1), which implies that the computations in these models can be performed more efficiently, see (Vomlel, 2002; Savicky and Vomlel, 2005).

5 Conclusions

We applied the tensor rank-one decomposition to the conditional probability tables representing Boolean functions. Using a numerical algorithm we have shown that the conditional probability tables of some commonly used Boolean function (threshold, exactly- t) can be nicely decomposed since their rank is lower than the average rank of a random Boolean function. This allows for more efficient computations when the probabilistic model contains conditional probability tables representing these functions.

Acknowledgments

This work was supported by the Ministry of Education of the Czech Republic under the project nr. 1M6798555601 Data, Algorithms, and Decision-Making and by the Grant Agency of the Czech Republic under the grant project nr. 201/04/0393.

References

- Boutilier C, Friedman N, Goldszmidt M, and Koller D. Context-specific independence in Bayesian networks. In *Proceedings of the 12th Annual Conference on Uncertainty in AI (UAI)*, pp. 115–123 (1996).
- Darwiche A. A logical approach to factoring belief networks. In *Proceedings of International Conference on Knowledge Representation and Reasoning*, pp. 409–420 (2002).
- Heckerman D and Breese JS. A new look at causal independence. In RL de Mantaras and D Poole, eds., *Proc. of the Tenth Conf. on Uncertainty in AI*, pp. 286–292 (1994).
- Håstad J. Tensor rank is NP-complete. *Journal of Algorithms*, 11:pp. 644–654 (1990).
- Jensen FV, Lauritzen SL, and Olesen KG. Bayesian updating in recursive graphical models by local computation. *Computational Statistics Quarterly*, 4:pp. 269–282 (1990).
- Lucas PJF. Bayesian network modelling through qualitative patterns. *Artificial Intelligence*, 163:pp. 233–263 (2005).
- Savicky P and Vomlel J. Tensor rank-one decomposition of probability tables (2005). Unpublished manuscript.
- Takikawa M and D’Ambrosio B. Multiplicative factorization of noisy-max. In K Laskey and H Prade, eds., *Proc. of the Fifteenth Conf. on Uncertainty in AI*, pp. 622–630 (1999).
- Vomlel J. Exploiting functional dependence in Bayesian network inference. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 528–535. Morgan Kaufmann Publishers (2002).
- Wegener I. *The Complexity of Boolean Functions*. Wiley-Teubner (1987).
- Zhang NL and Poole D. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:pp. 301–328 (1996).