Rank of tensors of l-out-of-k functions: an application in probabilistic inference

Jiří Vomlel

Institute of Information Theory and Automation (ÚTIA)
Academy of Sciences of the Czech Republic

Contents

Contents

- The computer game of Minesweeper
- Probabilistic reasoning given evidence
  (using a simple example)

Contents

- The computer game of Minesweeper
- Probabilistic reasoning given evidence (using a simple example)
- Improving the computational efficiency

Contents

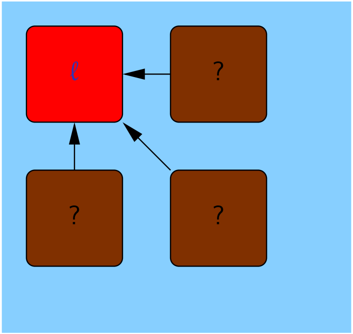
- The computer game of Minesweeper
- Probabilistic reasoning given evidence (using a simple example)
- Improving the computational efficiency
- Rank-one decomposition of probability tables representing addition

Contents

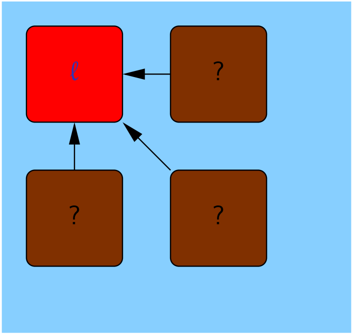
- The computer game of Minesweeper
- Probabilistic reasoning given evidence (using a simple example)
- Improving the computational efficiency
- Rank-one decomposition of probability tables representing addition
- Results of experiments

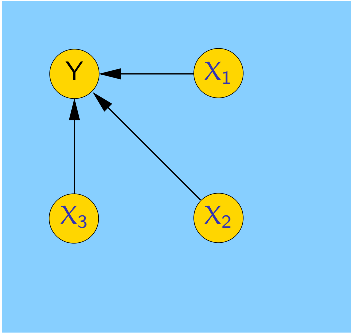
The game of Minesweeper

Bayesian network for the game of Minesweeper

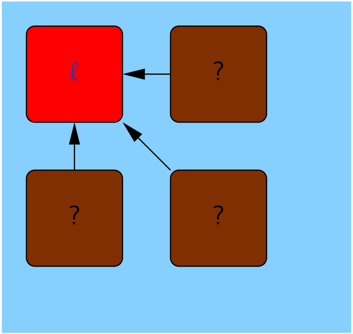


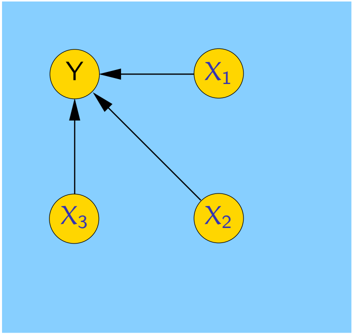
Bayesian network for the game of Minesweeper





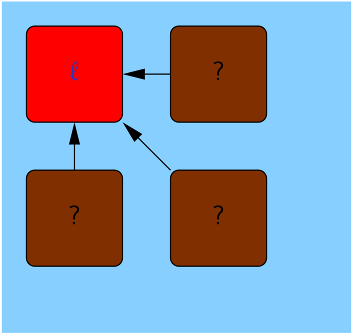
Bayesian network for the game of Minesweeper

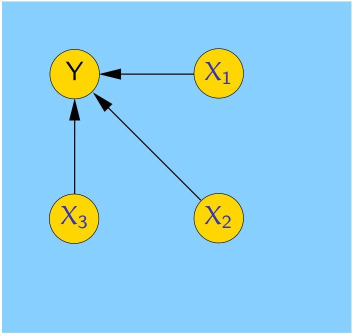




$$P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3) = \begin{cases} 1 & \text{if } \ell = x_1 + x_2 + x_3 \\ 0 & \text{otherwise.} \end{cases}$$

Bayesian network for the game of Minesweeper





$$P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3) = \begin{cases} 1 & \text{if } \ell = x_1 + x_2 + x_3 \\ 0 & \text{otherwise.} \end{cases}$$

$$P(X_i) = \frac{r}{s \cdot t - o}$$

r is the number of mines, o is the number of observations
s, t are the dimensions of the game grid.

Bayes rule for updating probabilities

- Assume we observe $Y = \ell$.

Bayes rule for updating probabilities

- Assume we observe $Y = \ell$.
- We compute by Bayes rule

$$P(X_1 = x_1, X_2 = x_2, X_3 = x_3 | Y = \ell)$$

Bayes rule for updating probabilities

- Assume we observe $Y = \ell$.
- We compute by Bayes rule

$$
\begin{aligned}
& P(X_1 = x_1, X_2 = x_2, X_3 = x_3 | Y = \ell) \\
= \quad & \frac{P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3) \cdot \prod_{i=1}^{3} P(X_i = x_i)}{P(Y = \ell)}
\end{aligned}
$$

Bayes rule for updating probabilities

- Assume we observe $Y = \ell$.
- We compute by Bayes rule

$$
\begin{aligned}
& P(X_1 = x_1, X_2 = x_2, X_3 = x_3 | Y = \ell) \\
= \quad & \frac{P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3) \cdot \prod_{i=1}^{3} P(X_i = x_i)}{P(Y = \ell)} \\
\propto \quad & P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3)
\end{aligned}
$$

Bayes rule for updating probabilities

- Assume we observe $Y = \ell$.
- We compute by Bayes rule

$$
\begin{aligned}
&P(X_1 = x_1, X_2 = x_2, X_3 = x_3 | Y = \ell) \\
&= \frac{P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3) \cdot \prod_{i=1}^{3} P(X_i = x_i)}{P(Y = \ell)} \\
&\propto P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3)
\end{aligned}
$$

- This is a probability table over 3 binary variables X_1, X_2, X_3:

$$
\begin{aligned}
&P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3) \\
&= \begin{cases} 1 & \text{if } x_1 + x_2 + x_3 = \ell \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

Bayes rule for updating probabilities

- Assume we observe $Y = \ell$.
- We compute by Bayes rule

$$
\begin{aligned}
& P(X_1 = x_1, X_2 = x_2, X_3 = x_3 | Y = \ell) \\
= \quad & \frac{P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3) \cdot \prod_{i=1}^{3} P(X_i = x_i)}{P(Y = \ell)} \\
\propto \quad & P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3)
\end{aligned}
$$

- This is a probability table over 3 binary variables X_1, X_2, X_3:

$$
\begin{aligned}
& P(Y = \ell | X_1 = x_1, X_2 = x_2, X_3 = x_3) \\
= \quad & \begin{cases} 1 & \text{if } x_1 + x_2 + x_3 = \ell \\ 0 & \text{otherwise.} \end{cases} \\
= \quad & \psi(X_1 = x_1, X_2 = x_2, X_3 = x_3) \ .
\end{aligned}
$$

Tensors of ℓ-out-of-k functions

We can visualize probability table ψ as a tensor (for $\ell = 1$):

$$\begin{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{pmatrix}$$

In this talk all tensors are functions from $\{0,1\}^k$ to real numbers.

Tensors of ℓ-out-of-k functions

We can visualize probability table ψ as a tensor (for $\ell = 1$):

$$\begin{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{pmatrix}$$

In this talk all tensors are functions from $\{0, 1\}^k$ to real numbers.
We are interested in tensors of ℓ-out-of-k functions $f_\ell(x_1, \ldots, x_k)$, where:

- ℓ is the observed state of Y and
- k is the number of binary variables - parents of Y.

Tensors of ℓ-out-of-k functions

We can visualize probability table ψ as a tensor (for $\ell = 1$):

$$\begin{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{pmatrix}$$

In this talk all tensors are functions from $\{0,1\}^k$ to real numbers.
We are interested in tensors of ℓ-out-of-k functions $f_\ell(x_1, \ldots, x_k)$, where:

- ℓ is the observed state of Y and
- k is the number of binary variables - parents of Y.

$$f_\ell(x_1, \ldots, x_k) = \begin{cases} 1 & \text{if } \ell = \sum_{i=1}^{k} x_i \\ 0 & \text{otherwise.} \end{cases}$$

Tensors of ℓ-out-of-k functions

We can visualize probability table ψ as a tensor (for $\ell = 1$):

$$\begin{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{pmatrix}$$

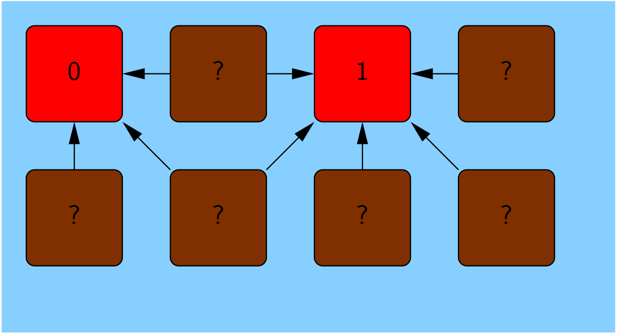
In this talk all tensors are functions from $\{0,1\}^k$ to real numbers.
We are interested in tensors of ℓ-out-of-k functions $f_\ell(x_1, \ldots, x_k)$, where:

- ℓ is the observed state of Y and
- k is the number of binary variables - parents of Y.

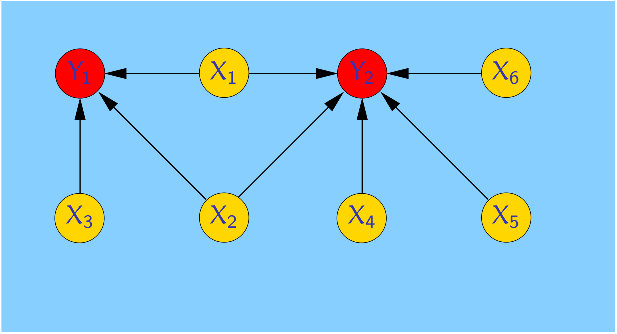
$$f_\ell(x_1, \ldots, x_k) = \begin{cases} 1 & \text{if } \ell = \sum_{i=1}^{k} x_i \\ 0 & \text{otherwise.} \end{cases}$$

In our example $\ell = 1$ and $k = 3$.

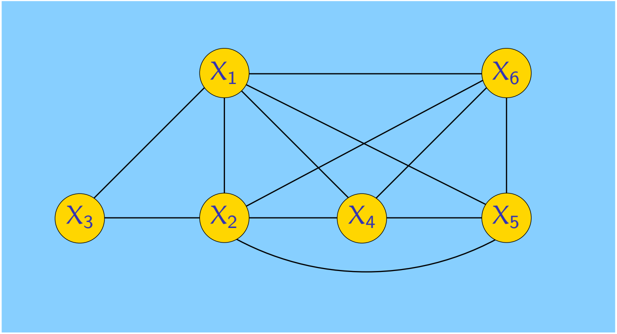
Combining information



Combining information

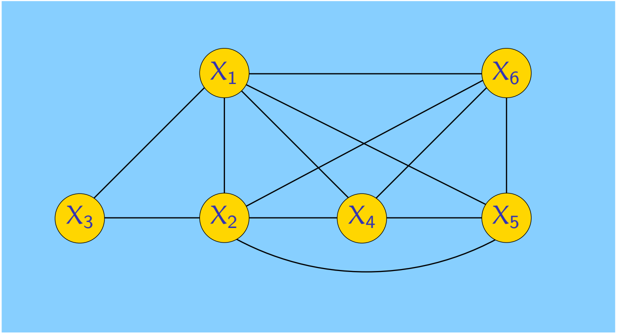


Combining information



$$\xi(X_1, \ldots, X_6) \quad = \quad \psi(X_1, \ldots, X_3) \cdot \varphi(X_1, X_2, X_4, \ldots, X_6)$$

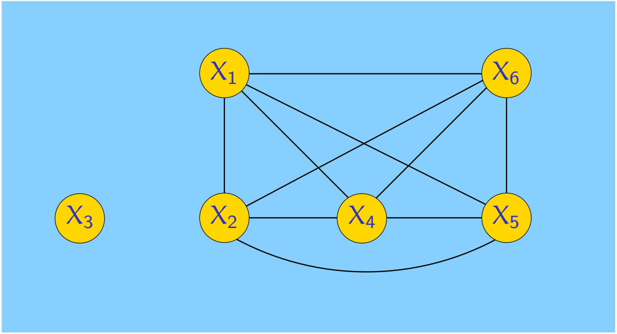
Combining information



$$\xi(X_1, \ldots, X_6) \quad = \quad \psi(X_1, \ldots, X_3) \cdot \varphi(X_1, X_2, X_4, \ldots, X_6)$$

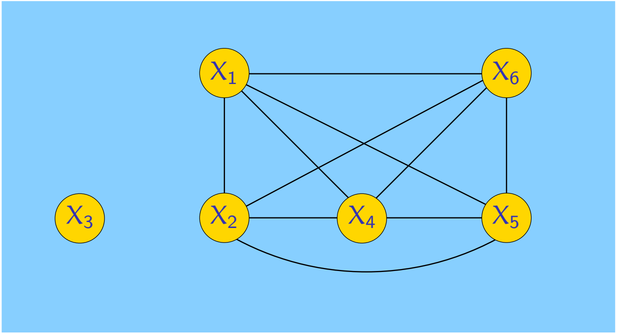
Total table size is $2^3 + 2^5 = 8 + 32 = 40$.

A more efficient way of combining information



$$
\begin{aligned}
\xi(X_1, \ldots, X_6) &= \psi_1(X_1) \cdot \ldots \cdot \psi_3(X_3) \\
&\quad \cdot \varphi_1(X_1, X_2, X_4, \ldots, X_6)
\end{aligned}
$$

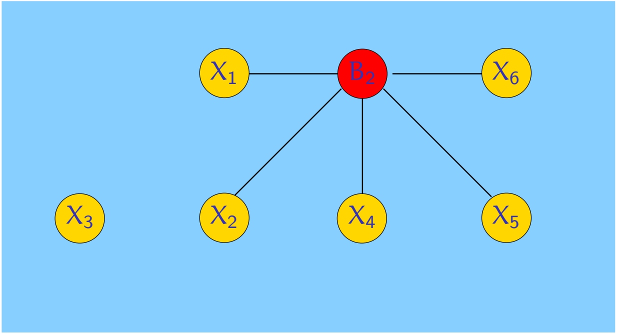
A more efficient way of combining information



$$\xi(X_1, \ldots, X_6) = \psi_1(X_1) \cdot \ldots \cdot \psi_3(X_3) \cdot \varphi_1(X_1, X_2, X_4, \ldots, X_6)$$

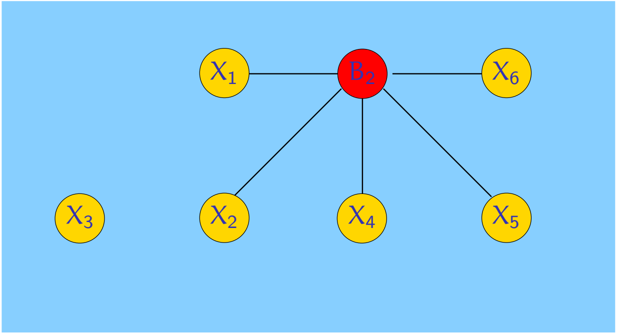
Total table size is $3 \cdot 2 + 2^5 = 6 + 32 = 38$.

An even more efficient way of combining information



$$\xi(X_1, \ldots, X_6) = \sum_{B_2} \psi_1(X_1) \cdot \ldots \cdot \psi_3(X_3)$$
$$\cdot\, \varphi_1(B_2, X_1) \cdot \varphi_2(B_2, X_2) \cdot \varphi_4(B_2, X_4) \ldots \varphi_6(B_2, X_6)$$

An even more efficient way of combining information



$$\xi(X_1, \ldots, X_6) = \sum_{B_2} \psi_1(X_1) \cdot \ldots \cdot \psi_3(X_3) \cdot \varphi_1(B_2, X_1) \cdot \varphi_2(B_2, X_2) \cdot \varphi_4(B_2, X_4) \ldots \varphi_6(B_2, X_6)$$

Since B is binary the total table size is $3 \cdot 2 + 5 \cdot 2^2 = 6 + 20 = 26$.

Tensor rank

We have just seen that

$$
\begin{aligned}
&\varphi_1(X_1, X_2, X_4, \ldots, X_6) \\
&\quad = \sum_{B_2} \varphi_1(B_2, X_1) \cdot \varphi_2(B_2, X_2) \cdot \varphi_4(B_2, X_4) \ldots \varphi_6(B_2, X_6) \enspace .
\end{aligned}
$$

Tensor rank

We have just seen that

$$\varphi_1(X_1, X_2, X_4, \ldots, X_6)$$
$$= \sum_{B_2} \varphi_1(B_2, X_1) \cdot \varphi_2(B_2, X_2) \cdot \varphi_4(B_2, X_4) \ldots \varphi_6(B_2, X_6) \ .$$

But there is no way we can write

$$\varphi_1(X_1, X_2, X_4, \ldots, X_6) \ = \ \varphi_1(X_1) \cdot \varphi_2(X_2) \cdot \varphi_4(X_4) \ldots \varphi_6(X_6)$$

Tensor rank

We have just seen that

$$\varphi_1(X_1, X_2, X_4, \ldots, X_6) = \sum_{B_2} \varphi_1(B_2, X_1) \cdot \varphi_2(B_2, X_2) \cdot \varphi_4(B_2, X_4) \ldots \varphi_6(B_2, X_6) \; .$$

But there is no way we can write

$$\varphi_1(X_1, X_2, X_4, \ldots, X_6) = \varphi_1(X_1) \cdot \varphi_2(X_2) \cdot \varphi_4(X_4) \ldots \varphi_6(X_6)$$

What is the minimal number of states of a variable B so that it holds that

$$\psi(X_1, \ldots, X_k) = \sum_{B} \prod_{i=1}^{k} \psi_i(B, X_i) \; ?$$

Tensor rank

We have just seen that

$$\varphi_1(X_1, X_2, X_4, \ldots, X_6) = \sum_{B_2} \varphi_1(B_2, X_1) \cdot \varphi_2(B_2, X_2) \cdot \varphi_4(B_2, X_4) \ldots \varphi_6(B_2, X_6) \ .$$

But there is no way we can write

$$\varphi_1(X_1, X_2, X_4, \ldots, X_6) = \varphi_1(X_1) \cdot \varphi_2(X_2) \cdot \varphi_4(X_4) \ldots \varphi_6(X_6)$$

What is the minimal number of states of a variable B so that it holds that

$$\psi(X_1, \ldots, X_k) = \sum_{B} \prod_{i=1}^{k} \psi_i(B, X_i) \ ?$$

This number is called the rank of tensor ψ.

Symmetric rank of tensors of ℓ-out-of-k functions

- Generally, finding the rank of a tensor is NP-hard.

Symmetric rank of tensors of ℓ-out-of-k functions

- Generally, finding the rank of a tensor is NP-hard.
- However, tensors of ℓ-out-of-k functions define a restricted class of tensors.

Symmetric rank of tensors of ℓ-out-of-k functions

- Generally, finding the rank of a tensor is NP-hard.
- However, tensors of ℓ-out-of-k functions define a restricted class of tensors.
- These tensors are all symmetric. A tensor ψ is symmetric if $\psi(X_1 = x_1, \ldots, X_k = x_k) = a_{x_1 + \ldots + x_k}$ where $\mathbf{a} = (a_0, \ldots, a_k)$ is a vector of real numbers.

Symmetric rank of tensors of ℓ-out-of-k functions

- Generally, finding the rank of a tensor is NP-hard.
- However, tensors of ℓ-out-of-k functions define a restricted class of tensors.
- These tensors are all symmetric. A tensor ψ is symmetric if $\psi(X_1 = x_1, \ldots, X_k = x_k) = a_{x_1 + \ldots + x_k}$ where $\mathbf{a} = (a_0, \ldots, a_k)$ is a vector of real numbers.
- The symmetric rank of tensor ψ is the minimum number of symmetric tensors of rank one that sum up to ψ.

Symmetric rank of tensors of ℓ-out-of-k functions

- Generally, finding the rank of a tensor is NP-hard.
- However, tensors of ℓ-out-of-k functions define a restricted class of tensors.
- These tensors are all symmetric. A tensor ψ is symmetric if $\psi(X_1 = x_1, \ldots, X_k = x_k) = a_{x_1+\ldots+x_k}$ where $\mathbf{a} = (a_0, \ldots, a_k)$ is a vector of real numbers.
- The symmetric rank of tensor ψ is the minimum number of symmetric tensors of rank one that sum up to ψ.

Theorem

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

Border rank of tensors of ℓ-out-of-k functions

Definition (Border rank)

The border rank of a tensor $\boldsymbol{A}$ is

$$\min\{r : \forall \varepsilon > 0 \ \ \exists \text{ tensor } \mathbf{E} : \|E\| < \varepsilon, \mathrm{rank}(\boldsymbol{A} + \mathbf{E}) = r\} \ ,$$

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

Border rank of tensors of ℓ-out-of-k functions

Definition (Border rank)

The border rank of a tensor $\boldsymbol{A}$ is

$$\min\{r : \forall \varepsilon > 0 \;\; \exists \text{ tensor } \mathbf{E} : ||E|| < \varepsilon, \mathrm{rank}(\boldsymbol{A} + \mathbf{E}) = r\} \;,$$

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

Theorem (Upper bound of the border rank)

The border rank of a tensor $\mathbf{A}(\ell, k)$ representing an ℓ-out-of-k function is at most $\min\{\ell + 1, k - \ell + 1\}$.

Tensor approximations

Given a symmetric tensor representing an ℓ-out-of-k function our goal is to find another symmetric tensor:

- of the same order and the same dimensions

Tensor approximations

Given a symmetric tensor representing an ℓ-out-of-k function our goal is to find another symmetric tensor:

- of the same order and the same dimensions
- having symmetric rank at most $r = \min\{\ell + 1, k - \ell + 1\}$

Tensor approximations

Given a symmetric tensor representing an ℓ-out-of-k function our goal is to find another symmetric tensor:

- 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.

Tensor approximations

Given a symmetric tensor representing an ℓ-out-of-k function our goal is to find another symmetric tensor:

- 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 used a kind of stochastic hill-clibing algorithm.

Tensor approximations - example

The tensor for

$$\left(\begin{array}{cc} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{array} \right)$$

Tensor approximations - example

The tensor for

$$\begin{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{pmatrix}$$

$$\sim \quad \frac{-0.19}{\exp(-15.75)}(1, \exp(-15.75)) \otimes \ldots \otimes (1, \exp(-15.75))$$

Tensor approximations - example

The tensor for

$$
\begin{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{pmatrix}
$$

$$
\begin{aligned}
\sim \quad & \frac{-0.19}{\exp(-15.75)} (1, \exp(-15.75)) \otimes \ldots \otimes (1, \exp(-15.75)) \\
& + \frac{1.19}{\exp(-13.90)} (1, \exp(-13.90)) \otimes \ldots \otimes (1, \exp(-13.90))
\end{aligned}
$$

Tensor approximations - example

The tensor for

$$\begin{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{pmatrix}$$

$$\begin{aligned} \sim \quad & \frac{-0.19}{\exp(-15.75)}(1, \exp(-15.75)) \otimes \ldots \otimes (1, \exp(-15.75)) \\ & + \frac{1.19}{\exp(-13.90)}(1, \exp(-13.90)) \otimes \ldots \otimes (1, \exp(-13.90)) \end{aligned}$$

$$= \begin{pmatrix} \begin{pmatrix} 2.33 \cdot 10^{-10} \\ 1.0 \end{pmatrix} & \begin{pmatrix} 1.0 \\ 1.07 \cdot 10^{-6} \end{pmatrix} \\ \begin{pmatrix} 1.0 \\ 1.07 \cdot 10^{-6} \end{pmatrix} & \begin{pmatrix} 1.07 \cdot 10^{-6} \\ 9.96 \cdot 10^{-13} \end{pmatrix} \end{pmatrix}$$

Tensor with noisy inputs

In the real world there is usually a noise that modifies functional relations between variables.

Tensor with noisy inputs

In the real world there is usually a noise that modifies functional relations between variables.

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

Tensor with noisy inputs

In the real world there is usually a noise that modifies functional relations between variables.

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

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

Tensor with noisy inputs

In the real world there is usually a noise that modifies functional relations between variables.

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

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

where $A_{j_1, j_2, \ldots, j_k}(\ell, k)$ represents the (exact) ℓ-out-of-k function,

Tensor with noisy inputs

In the real world there is usually a noise that modifies functional relations between variables.

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

$$\mathbf{N}(\ell, k, p, q)_{i_1, i_2, \ldots, i_k}$$

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

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

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

$0 < p \leqslant 1$, $0 < q \leqslant 1$ are parameters of the input noise.

Experiments

We performed experiments with the game of Minesweeper for the 20×20 grid size.

Experiments

We performed experiments with the game of Minesweeper for the 20×20 grid size.
We used a random selection of fields to be played and we assumed we never hit any of fifty mines during the game.

Experiments

We performed experiments with the game of Minesweeper for the 20×20 grid size.
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

1. the standard method consisting of moralization and triangulation steps and

Experiments

We performed experiments with the game of Minesweeper for the 20×20 grid size.
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

1. the standard method consisting of moralization and triangulation steps 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) followed by the triangulation step.

Experiments

We performed experiments with the game of Minesweeper for the 20×20 grid size.
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

1. the standard method consisting of moralization and triangulation steps 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) followed by the triangulation step.

In both networks we then used the lazy propagation method of Madsen and Jensen with the computations performed with lists of tables over the junction trees.

Results of experiments

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. See Figure.