Computationally efficient methods for partition function estimation in graphical models in TT-format

D. Vetrov

Lomonosov Moscow State University

MIT

Notation

Tensor — multidimensional array (function of its indices)

$$\mathbf{A}(x) = \mathbf{A}(x_1, \ldots, x_n), \quad x_i \in \{1, \ldots, d_i\}$$

Terminology:

- n — *dimensionality*;
- x_i — *indices*.

TT-format

A tensor $\mathbf{A}$ is said to be represented in the TT-format if

$$\mathbf{A}(x_1, \ldots, x_n) = G_1^{\mathbf{A}}[x_1] G_2^{\mathbf{A}}[x_2] \cdots G_n^{\mathbf{A}}[x_n],$$

where $G_i^{\mathbf{A}}[x_i]$ is a matrix of size $\mathrm{r}_{i-1}(\mathbf{A}) \times \mathrm{r}_i(\mathbf{A})$, $\mathrm{r}_0(\mathbf{A}) = \mathrm{r}_n(\mathbf{A}) = 1$.
Terminology:

- $G_i^{\mathbf{A}}$ — TT-cores;
- $\mathrm{r}_i(\mathbf{A})$ — TT-ranks;
- $r(\mathbf{A}) = \max\limits_{i=0,\ldots,n} r_i(\mathbf{A})$ — maximal TT-rank.

The TT-format uses $O\left(ndr^2(\mathbf{A})\right)$ memory to store $O\left(d^n\right)$ elements ($d = \max\limits_{i=1,\ldots,n} d_i$).

Operations on tensors in the TT-format

Operation	Output rank
$\mathbf{C} = c \cdot \mathbf{A}$	$\mathrm{r}(\mathbf{C}) = \mathrm{r}(\mathbf{A})$
$\mathbf{C} = \mathbf{A} + c$	$\mathrm{r}(\mathbf{C}) = \mathrm{r}(\mathbf{A}) + 1$
$\mathbf{C} = \mathbf{A} + \mathbf{B}$	$\mathrm{r}(\mathbf{C}) \leq \mathrm{r}(\mathbf{A}) + \mathrm{r}(\mathbf{B})$
$\mathbf{C} = \mathbf{A} \odot \mathbf{B}$	$\mathrm{r}(\mathbf{C}) \leq \mathrm{r}(\mathbf{A})\,\mathrm{r}(\mathbf{B})$
$\boldsymbol{c} = M\boldsymbol{b}$	$\mathrm{r}(\boldsymbol{c}) \leq \mathrm{r}(M)\,\mathrm{r}(\boldsymbol{b})$
$\mathbf{C} = \mathrm{round}(\mathbf{A}, \varepsilon)$	$\mathrm{r}(\mathbf{C}) \leq \mathrm{r}(\mathbf{A})$
sum $\mathbf{A}$	–
$\|\mathbf{A}\|_F$	–

TT-rounding

Suppose that you have a TT-decomposition $\mathbf{A}(x) = G_1^{\mathbf{A}}[x_1] \cdots G_n^{\mathbf{A}}[x_n]$ with non-optimal TT-ranks.

The TT-rounding procedure (Oseledets 2011)

$$\widehat{\mathbf{A}} = \text{round}(\mathbf{A}, \varepsilon), \quad \varepsilon \geq 0$$

finds a tensor $\widehat{\mathbf{A}}$:

1. $\|\mathbf{A} - \widehat{\mathbf{A}}\|_F \leq \varepsilon \|\mathbf{A}\|_F$;
2. its TT-ranks are minimal among all tensors $\mathbf{B}$: $\|\mathbf{A} - \mathbf{B}\|_F \leq \frac{\varepsilon}{\sqrt{n-1}} \|\mathbf{A}\|_F$.

Here $\|\mathbf{A}\|_F = \sqrt{\sum_{x_1,\ldots,x_n} \mathbf{A}^2(x_1, \ldots, x_n)}$.

Finding a TT-representation of a tensor

- There are analytical formulae for some special cases;

Example

$$\mathbf{A}(x_1, \ldots, x_n) = \sum_{i=1}^{n} f_i(x_i),$$

$$G_i^{\mathbf{A}}[x_i] = \begin{bmatrix} 1 & 0 \\ f_i(x_i) & 1 \end{bmatrix}, \quad i = 2, \ldots, n-1,$$

$$G_1^{\mathbf{A}}[x_1] = \begin{bmatrix} f_1(x_1) & 1 \end{bmatrix}, \quad G_n^{\mathbf{A}}[x_n] = \begin{bmatrix} 1 \\ f_n(x_n) \end{bmatrix}.$$

Finding a TT-representation of a tensor

- There are analytical formulae for some special cases;
- TT-SVD: finds an exact TT-representation for a tensor but suitable only for low dimensionality n;
- AMEn-cross: builds a TT-approximation of a tensor by using only a small fraction of its elements; suitable for high dimensionality n but doesn't have strong theoretical guarantees.

Semantic segmentation

Training

X_{train}

T_{train}

Testing

?

Probabilistic approach

- Define a probabilistic model on the set of all possible labellings.
- Let $p(T|X, W)$ measure the probability of the labelling T given the image X and the parameters of the model W.
- The goal is to find the labelling T^* that maximizes $p(T|X, W)$:

$$T^* = \arg\max_T p(T|X, W).$$

  This is called the maximum a posteriori (MAP) inference.
- We will use Markov random fields (MRFs) to define the probabilistic model $p(T|X, W)$.

MAP-inference

The MAP-inference problem now corresponds to the following problem:

$$\max_T p(T|X, W) = \max_T \frac{1}{Z(X, W)} \prod_{c \in \mathcal{C}} \Psi_c(T_c; X, W).$$

Further we demonstrate how one can address such a problem using the Tensor-Train (TT) framework.

We will assume that

- the parameters of the model W are already chosen;
- we are performing the MAP-inference for the concrete image X.

So, to simplify notation, we won't explicitly write X, W any more:

$$\max_T p(T) = \max_T \frac{1}{Z} \prod_{c \in \mathcal{C}} \Psi_c(T_c)$$

MAP-inference & energy minimization

The MAP-inference problem

$$\max_{\boldsymbol{x}} \boldsymbol{P}(\boldsymbol{x}) = \max_{\boldsymbol{x}} \frac{1}{Z} \prod_{\ell=1}^{m} \boldsymbol{\Psi}_{\ell}(\boldsymbol{x}^{\ell})$$

is equivalent to the following problem:

$$\min_{\boldsymbol{x}} \sum_{\ell=1}^{m} [-\ln \boldsymbol{\Psi}_{\ell}(\boldsymbol{x}^{\ell})].$$

MAP-inference & energy minimization

The MAP-inference problem

$$\max_{\boldsymbol{x}} \boldsymbol{P}(\boldsymbol{x}) = \max_{\boldsymbol{x}} \frac{1}{Z} \prod_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$$

is equivalent to the following problem:

$$\min_{\boldsymbol{x}} \sum_{\ell=1}^{m} [-\ln \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)].$$

Terminology:

- The terms $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell) = -\ln \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell)$ are called MRF potentials.
- Their sum $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ is called MRF energy.

So, the MAP-inference is equivalent to energy minimization:

$$\min_{\boldsymbol{x}} \boldsymbol{E}(\boldsymbol{x}) = \min_{\boldsymbol{x}} \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell).$$

Learning

Maximum likelihood estimation:

$$
\begin{aligned}
W^* &= \arg\max_W \mathbf{P}(T_{\text{train}} | X_{\text{train}}, W) \\
&= \arg\max_W \frac{1}{Z(X_{\text{train}, W})} \prod_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(T_{\text{train}}; X_{\text{train}}, W) \\
&= \arg\max_W \left(\sum_{\ell=1}^{m} \log \boldsymbol{\Psi}_\ell(T_{\text{train}}; X_{\text{train}}, W) - \log Z(X_{\text{train}}, W) \right)
\end{aligned}
$$

Optimization

We need to solve the following optimization task:

$$\max_{W} \left(\sum_{\ell=1}^{m} \log \Psi_\ell(T_{\text{train}}; X_{\text{train}}, W) - \log Z(X_{\text{train}}, W) \right)$$

$$Z(X_{\text{train}}, W) = \sum_{T} \prod_{\ell=1}^{m} \Psi_\ell(T; X_{\text{train}}, W)$$

To use optimization methods we need to compute the value and the gradient of the objective function.
It is easy to compute the gradient of the log-partition function if we know the marginal distributions[1]:

$$\mathbf{P}(t_i) = \sum_{T \setminus t_i} \mathbf{P}(T)$$

[1] S. Nowozin and C. Lampert (2010). "Structured Learning and Prediction in Computer Vision". In: *Foundations and Trends in Computer Graphics and Vision* 6.3–4, pp. 185–365.

Problems of interest

Assume that we fixed the parameters W and the training data X_{train}, T_{train}.
We will focus on three problems (hereinafter we use $\boldsymbol{x}$ to denote the variables of the model):

- MAP-inference: $\min_{\boldsymbol{x}} \mathbf{E}(\boldsymbol{x})$
- The partition function estimation: $Z = \sum_{\boldsymbol{x}} \widehat{\mathbf{P}}(\boldsymbol{x})$;
- The marginal distributions estimation: $\mathbf{P}(x_i) = \frac{1}{Z} \sum_{\boldsymbol{x} \setminus x_i} \widehat{\mathbf{P}}(\boldsymbol{x})$.

Tensor approach

- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be considered as an n-dimensional tensor:
$$\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{E}(x_1, \ldots, x_n).$$
- Then energy minimization corresponds to finding the minimal element in the tensor $\boldsymbol{E}(\boldsymbol{x})$.
- If the energy $\boldsymbol{E}(\boldsymbol{x})$ were represented in the TT-format, we could use a special algorithm from the TT-framework to find the minimal element.
- How to convert the energy tensor into the TT-format? AMEn-algorithm?

Tensor approach

- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be considered as an n-dimensional tensor:
$$\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{E}(x_1, \ldots, x_n).$$
- Then energy minimization corresponds to finding the minimal element in the tensor $\boldsymbol{E}(\boldsymbol{x})$.
- If the energy $\boldsymbol{E}(\boldsymbol{x})$ were represented in the TT-format, we could use a special algorithm from the TT-framework to find the minimal element.
- How to convert the energy tensor into the TT-format?
AMEn-algorithm? Possible, but there is also a much better way!

The idea of the algorithm

- Let's try to take into account the structure of the energy tensor $\boldsymbol{E}$. Recall: $\boldsymbol{E}(\boldsymbol{x}) = \sum\limits_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
- Each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ can be considered as an n-dimensional tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$ if we add inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ for non-existing dimensions: $\boldsymbol{\Theta}_\ell(\boldsymbol{x}) \equiv \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
- The energy $\boldsymbol{E}(\boldsymbol{x})$ can be expressed as a sum of the tensors $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$:

$$\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}).$$

- If the tensors $\boldsymbol{\Theta}_\ell$ were represented in the TT-format, we could exploit the summation operation on tensors in the TT-format to build the TT-representation for the tensor $\boldsymbol{E}$.
- How to find the TT-decomposition for each tensor $\boldsymbol{\Theta}_\ell$?

Converting potentials into the TT-Format

- As opposed to the energy $\boldsymbol{E}(\boldsymbol{x})$, each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on part of the all variables and is usually of low dimensionality.
- To compute the TT-decomposition of the tensor $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$, we can use the TT-SVD algorithm.
- All that remains is to add the inessential variables $\boldsymbol{x} \setminus \boldsymbol{x}^\ell$ to $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ so as to make it n-dimensional.
- These inessential variables can be added constructively:
  - Let $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)$, $\boldsymbol{x}^\ell = (x_1, x_2, x_4)$.
  - Suppose that after TT-SVD we have:
$$\boldsymbol{\Theta}_\ell(x_1, x_2, x_4) = G_1[x_1] G_2[x_2] G_4[x_4].$$
  - To introduce x_3, x_5, we need to define the missing cores $G_3[x_3]$, $G_5[x_5]$.
  - Define them as identity matrices:
$$\boldsymbol{\Theta}_\ell(x_1, x_2, x_3, x_4, x_5) = G_1[x_1] G_2[x_2] \underbrace{I}_{\equiv G_3[x_3]} G_4[x_4] \underbrace{I}_{\equiv G_5[x_5]}.$$
  - The maximal TT-rank hasn't increased!

The algorithm & its theoretical guarantees

1. Compute the TT-decomposition for each individual potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
2. Add the inessential variables to each $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ to obtain $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$.
3. Use the TT-summation to build $\boldsymbol{E}(\boldsymbol{x})$: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x})$.

The algorithm & its theoretical guarantees

1. Compute the TT-decomposition for each individual potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$.
2. Add the inessential variables to each $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ to obtain $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$.
3. Use the TT-summation to build $\boldsymbol{E}(\boldsymbol{x})$: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x})$.

Theorem

The maximal TT-rank of the tensor $\boldsymbol{E}$ constructed by the algorithm is polynomially bounded:

$$\mathrm{r}(\boldsymbol{E}) \leq d^{\frac{p}{2}} m,$$

where

- *d is the number of values that each variable can take;*
- *m is the total number of potentials;*
- *p is the maximal order of a potential (i.e. the maximal $|\boldsymbol{x}^\ell|$).*

Consider $d = 2$, $p = 2$. Then $\mathrm{r}(\boldsymbol{E}) \leq 2m$ (linear dependence on m).

The TT-format for probability tensor

We can find the TT-representation of $\widehat{\mathbf{P}}$

$$\widehat{\mathbf{P}} = \bigodot_{\ell=1}^{m} \boldsymbol{\Psi}_\ell.$$

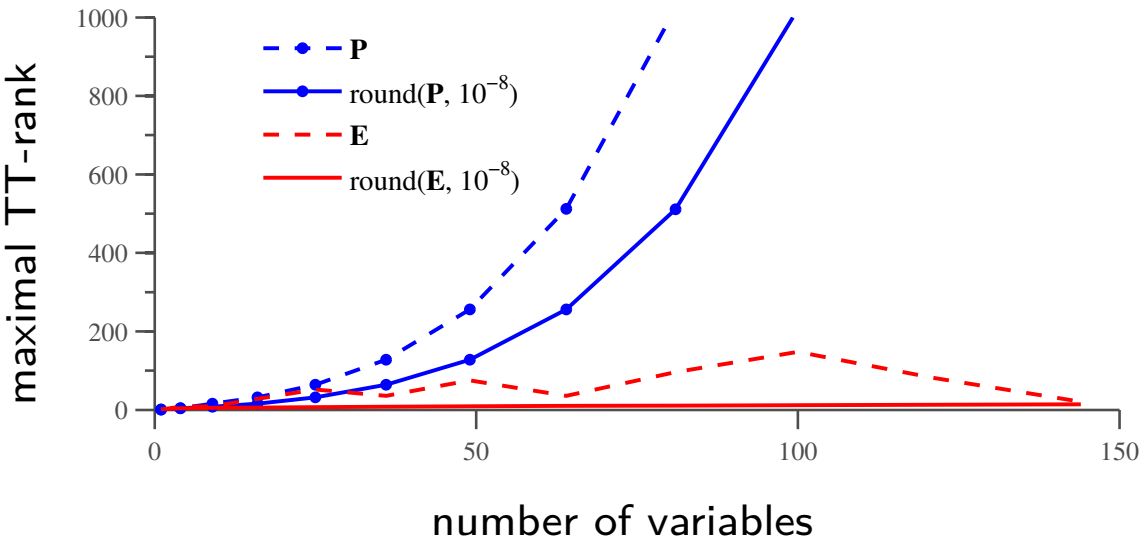
However, the TT-ranks of $\widehat{\mathbf{P}}$ are exponential.

The TT-format for probability tensor

We can find the TT-representation of $\widehat{\mathbf{P}}$

$$\widehat{\mathbf{P}} = \bigodot_{\ell=1}^{m} \boldsymbol{\Psi}_\ell.$$

However, the TT-ranks of $\widehat{\mathbf{P}}$ are exponential.



The algorithm motivation

- The TT-ranks of $\widehat{\mathbf{P}}$ grow exponentially;
- We must approximate the partition function Z without explicitly building the TT- representation of $\widehat{\mathbf{P}}$.

The partition function estimation

Recall the definition of the partition function: $Z = \sum_{\boldsymbol{x}} \widehat{\mathbf{P}}(\boldsymbol{x})$.

Let us transform $\widehat{\mathbf{P}}(\boldsymbol{x})$.

Kronecker product property:

$$a \cdot b = a \otimes b, \quad a \in \mathbb{R},\ b \in \mathbb{R}.$$

Applying this property we get

$$\widehat{\mathbf{P}}(\boldsymbol{x}) = \prod_{\ell=1}^{m} \underbrace{\boldsymbol{\Psi}_\ell(\boldsymbol{x})}_{\in \mathbb{R}} = \bigotimes_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(\boldsymbol{x}) = \bigotimes_{\ell=1}^{m} \left(G_1^\ell[x_1] \cdots G_n^\ell[x_n] \right).$$

The partition function estimation

Recall the definition of the partition function: $Z = \sum_{\boldsymbol{x}} \widehat{\mathbf{P}}(\boldsymbol{x})$.

Let us transform $\widehat{\mathbf{P}}(\boldsymbol{x})$.

Kronecker product property:

$$a \cdot b = a \otimes b, \quad a \in \mathbb{R},\ b \in \mathbb{R}.$$

Applying this property we get

$$\widehat{\mathbf{P}}(\boldsymbol{x}) = \prod_{\ell=1}^{m} \underbrace{\boldsymbol{\Psi}_\ell(\boldsymbol{x})}_{\in \mathbb{R}} = \bigotimes_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(\boldsymbol{x}) = \bigotimes_{\ell=1}^{m} \left(G_1^\ell[x_1] \cdots G_n^\ell[x_n] \right).$$

Mixed product property:

$$AC \otimes BD = (A \otimes B)(C \otimes D).$$

Hence:

$$\widehat{\mathbf{P}}(\boldsymbol{x}) = \left(G_1^1[x_1] \otimes \cdots \otimes G_1^m[x_1] \right) \cdots \left(G_n^1[x_n] \otimes \cdots \otimes G_n^m[x_n] \right).$$

The partition function estimation cont'd

$$\widehat{\mathbf{P}}(\boldsymbol{x}) = \left(G_1^1[x_1] \otimes \cdots \otimes G_1^m[x_1] \right) \cdots \left(G_n^1[x_n] \otimes \cdots \otimes G_n^m[x_n] \right).$$

Denote: $A_i[x_i] = G_i^1[x_i] \otimes \cdots \otimes G_i^m[x_i]$.

The partition function estimation cont'd

$$\widehat{\mathbf{P}}(\boldsymbol{x}) = \left(G_1^1[x_1] \otimes \cdots \otimes G_1^m[x_1]\right) \cdots \left(G_n^1[x_n] \otimes \cdots \otimes G_n^m[x_n]\right).$$

Denote: $A_i[x_i] = G_i^1[x_i] \otimes \cdots \otimes G_i^m[x_i]$.

Finally,

$$\begin{aligned} Z = \sum_{\boldsymbol{x}} \widehat{\mathbf{P}}(\boldsymbol{x}) &= \sum_{x_1,\ldots,x_n} A_1[x_1] \ldots A_n[x_n] \\ &= \left(\sum_{x_1} A_1[x_1]\right) \ldots \left(\sum_{x_n} A_n[x_n]\right) = B_1 \cdots B_n, \end{aligned}$$

where $B_i = \sum_{x_i=1}^{d} A_i[x_i]$.

The algorithm

$$Z = B_1 \cdots B_n,$$

The algorithm:

1: Find TT-cores $G_1^\ell, \ldots, G_n^\ell$ for $\boldsymbol{\Psi}_\ell$
2: Initialize $\boldsymbol{f}_{n+1} := 1$
3: **for** $i := n$ **downto** 1 **do**
4: Construct TT-matrix $A_i[x_i] := \bigotimes_{\ell=1}^{m} G_i^\ell[x_i]$
5: $B_i := \sum_{x_i=1}^{d} A_i[x_i]$
6: $\boldsymbol{f}_i := \text{round}(B_i \boldsymbol{f}_{i+1}, \varepsilon)$
7: **end for**
8: $\widetilde{Z} := \boldsymbol{f}_1$

Marginal distributions

Our approach can be generalized to compute the marginal distributions:

$$\hat{p}_i(x_i) = \sum_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n} \widehat{\mathbf{P}}(\boldsymbol{x}) = \sum_{\boldsymbol{x}\setminus x_i} A_1[x_1]\ldots A_n[x_n] =$$

$$= B_1 \ldots B_{i-1}\, A_i[x_i]\, B_{i+1} \ldots B_n,$$

where

$$B_i = \sum_{x_i=1}^{d} A_i[x_i].$$

We can explicitly normalize the obtained distribution

$$p_i(x_i) = \frac{\hat{p}_i(x_i)}{\sum_{x=1}^{d} \hat{p}_i(x)}$$

MAP-inference

The TT-method for the MAP-inference:

1. Convert the energy into the TT-format;
2. Find the minimal element in the energy tensor.

We compare the TT-method with the popular TRW-S algorithm on several real-world image segmentation problems from the OpenGM database.

Problem	**Variables**	**Labels**	**TRW-S**	**TT**	**Time** (sec)
gm6	320	3	45.03	43.11	637
gm29	212	3	56.81	56.21	224
gm66	198	3	75.19	74.92	172
gm105	237	3	67.81	67.71	230
gm32	100	7	150.50	289.29	257
gm70	122	7	121.78	163.60	399
gm85	143	7	168.30	228.40	1 912
gm192	99	7	114.51	174.78	180

Partition function

Ising model:

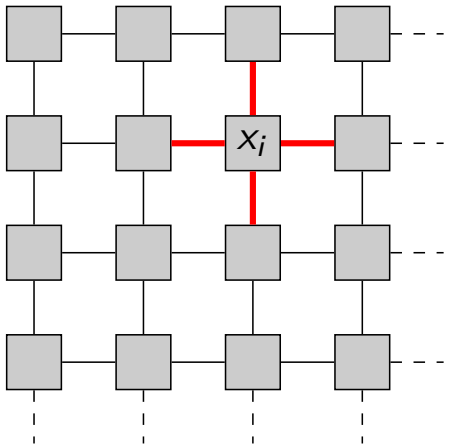
$$\widehat{\mathbf{P}}(\boldsymbol{x}) = \prod_{i=1}^{n} \exp\left(-\frac{1}{T} h_i x_i\right) \prod_{(i,j)\in\mathcal{E}} \exp\left(-\frac{1}{T} c_{ij} x_i x_j\right)$$

where $x_i \in \{-1, 1\}$.

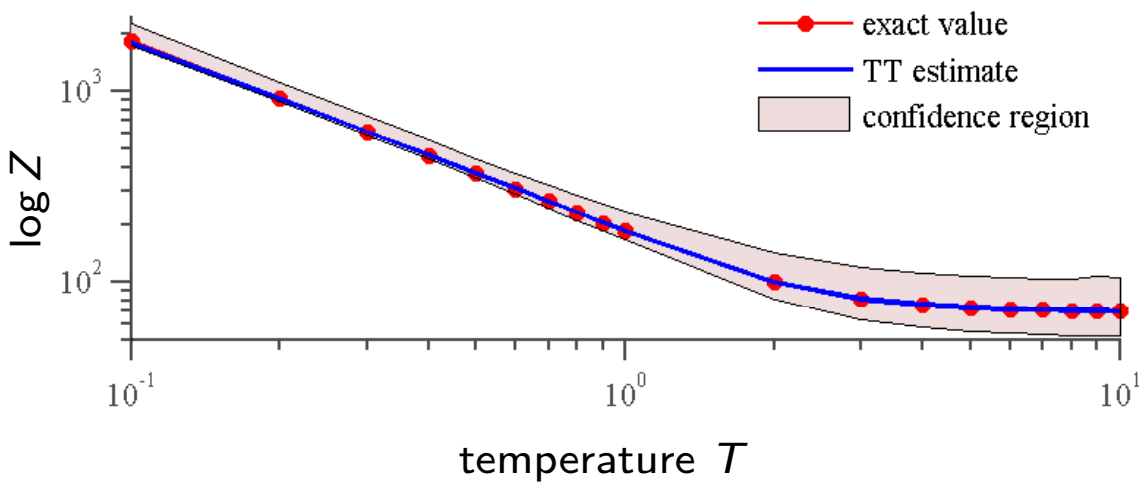
Notation:

- *Temperature* T;
- *Unary coefficients* h_i;
- *Pairwise coefficients* c_{ij}.

We use 4-connected grid of size 10×10.



Theoretical guaranties



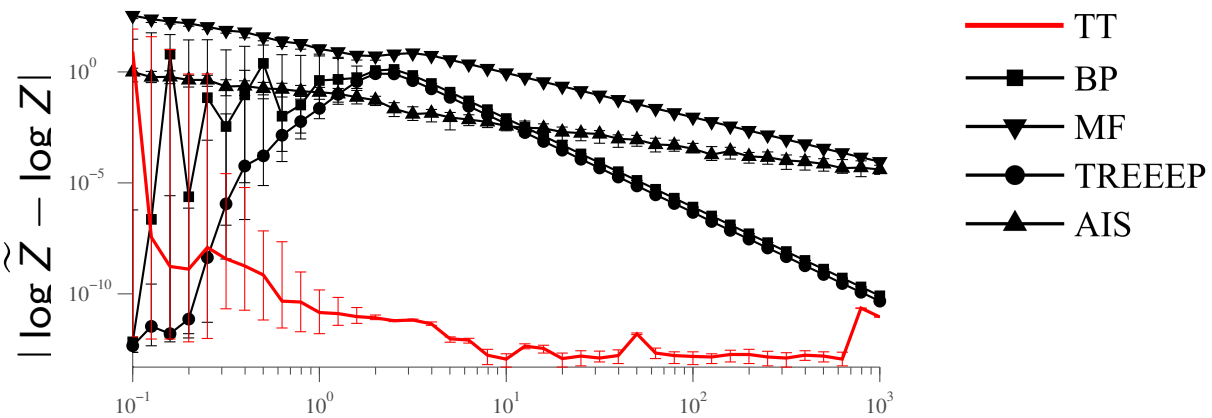
Confidence bounds on the computation of the partition function Z ($c_{ij} = 1$).

Methods

Methods we compare with:

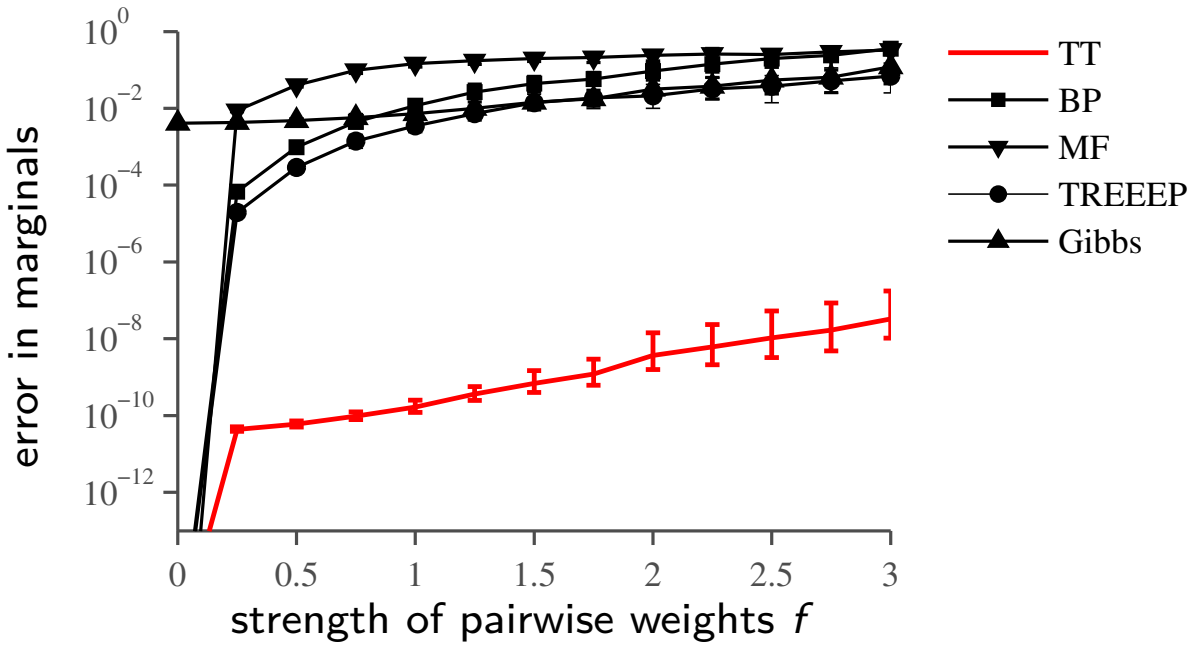
- Belief Propagation (BP, Kschischang, Frey, and Loeliger 2001 Kschischang et al., 2001);
- Tree Expectation Propagation (TREEEP, Minka and Qi 2004 Minka & Qi, 2004);
- Mean Field method (MF, Wainwright and Jordan 2008 Wainwright & Jordan, 2008);
- Annealed importance sampling method (AIS, Neal 2001 Neal, 2001);
- Gibbs sampling (Gibbs, Wainwright and Jordan 2008 Wainwright & Jordan, 2008).

Comparison



Comparison on Ising model (all pairwise weights are equal $c_{ij} = 1$) against state-of-the-art algorithms.

Marginal distributions



Ising models, $T = 1$, $c_{ij} \sim U[-f, f]$.

High Order Potentials

- Sometimes it is convenient to use potentials of high order, i.e. those which depend on many variables. E.g., the potential

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = \underbrace{\left[\sum_{i=1}^{n} x_i \leq a\right]}_{\text{indicator function}},$$

  which depends on all the variables, could be used to specify some preference on the minimal value of the area of foreground in the problem of segmenting an image into background/foreground.
- We can't use the TT-SVD algorithm any more to convert such potentials into the TT-format!
- However, for some of these potentials we can explicitly construct the TT-representation, i.e. we can derive analytical formulae for the corresponding TT-cores.
- Such TT-representations will be of low TT-rank!

Sparse Potential

- Consider a so-called sparse potential:

$$\boldsymbol{\Theta}_\ell(x_{i_1}, \ldots, x_{i_w}) = [x_{i_1} = \beta_1] \ldots [x_{i_w} = \beta_w].$$

  It always equals zero with the exception of only one configuration.

- Such a potential admits a TT-representation

$$\boldsymbol{\Theta}_\ell(x_{i_1}, \ldots, x_{i_w}) = G_{i_1}[x_{i_1}] \ldots G_{i_w}[x_{i_w}]$$

  with the following TT-cores:

$$G_{i_v}[x_{i_v}] = [x_{i_v} = \beta_v], \quad v = 1, \ldots, w.$$

- In this case each TT-core is simply a number (1-by-1 matrix) for every concrete value of x_{i_v}. Hence, the maximal TT-rank equals 1.
- A more general sparse potential which differs from zero on $s > 1$ configurations can be obtained as a sum of several potentials of the above type. Thus, the TT-rank of a general sparse potential is bounded above by s.

Area Potential

- Consider the potential

$$\boldsymbol{\Theta}_\ell(\boldsymbol{x}) = \left[\sum_{i=1}^{n} x_i \leq a\right],$$

where $x_i \in \{0, 1\}$ and $a \in \mathbb{Z}_+$.

- This potential can be analytically represented in the TT-format with the maximal TT-rank equal to $a + 1$:

$$G_i[x_i] = (S_a)^{x_i}, \quad (i = 2, \ldots, n-1),$$

$$G_1[x_1] = [\overbrace{\underbrace{0 \ldots 0}_{x_1} 1 \ldots 1}^{a+1}], \quad G_n[x_n] = (S_a)^{x_n}[\underbrace{0 \ldots 0}_{a} \; 1]^T,$$

where $S_a = \underbrace{\left[\begin{array}{c|c} O & I_a \\ \hline O & O \end{array}\right]}_{(a+1)\times(a+1)}$.

Area Potential cont'd

Key property of S_a: $[\overbrace{0\ldots0}^{k}\ 1\ldots1]S_a = [\overbrace{0\ldots0}^{k+1}\ 1\ldots1]$.

Consider, e.g., that $a = 3$. In this case

$$S_a = \left[\begin{array}{c|ccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right].$$

- $[1\,1\,1\,1]S_a = [0\,1\,1\,1]$ (the sum of all rows);
- $[0\,1\,1\,1]S_a = [0\,0\,1\,1]$ (the sum of rows 2, 3, 4);
- and so on.

Then

$$\begin{aligned} \boldsymbol{\Theta}_\ell(\boldsymbol{x}) &= G_1[x_1]G_2[x_2]G_3[x_3]\ldots G_n[x_n] \\ &= [\overbrace{0\ldots0}^{x_1}\ 1\ldots1](S_a)^{x_2}(S_a)^{x_3}\ldots(S_a)^{x_n}[\underbrace{0\ldots0}_{a}\ 1]^T \end{aligned}$$

Conclusion

Our main contributions are:

- We have proposed an algorithm that converts MRF energy into the TT-format exactly.
- We have derived an upper bound on the TT-ranks of the energy tensor constructed by the proposed algorithm.
- We have proposed an algorithm for estimating the partition function and the marginal distributions. The key feature of the algorithm is that it does not explicitly construct the TT-representation of the unnormalized joint distribution.
- We have bounded the errors of the partition function estimation.

Future Work

- Analytical formulae of TT-representations for other types of high order potentials.
- Better theoretical guaranties for the partition function estimation.
- Better algorithm for finding the minimal element in a tensor represented in the TT-format.

References I

- Kschischang, F. R., B. J. Frey, and H.-A. Loeliger (2001). "Factor Graphs and the Sum-Product Algorithm". In: *IEEE Transactions on Information Theory* 47.2, pp. 498–519.
- Minka, T. and Y. Qi (2004). "Tree-structured Approximations by Expectation Propagation". In: *Advances in Neural Information Processing Systems 16 (NIPS)*, pp. 193–200.
- Neal, R. (2001). "Annealed importance sampling". In: *Statistics and Computing* 11, pp. 125–139.
- Nowozin, S. and C. Lampert (2010). "Structured Learning and Prediction in Computer Vision". In: *Foundations and Trends in Computer Graphics and Vision* 6.3–4, pp. 185–365.
- Oseledets, I. V. (2011). "Tensor-Train Decomposition". In: *SIAM J. Scientific Computing* 33.5, pp. 2295–2317.

References II

Wainwright, M. J. and M. I. Jordan (2008). "Graphical models, exponential families, and variational inference". In: *Foundations and Trends in Machine Learning* 1.1–2, pp. 1–305.