Putting Markov Random Fields on a Tensor Train

Alexander Novikov[1] Anton Rodomanov[1] Anton Osokin[1]
Dmitry Vetrov[1,2]

[1] Moscow State University, Russia [2] Higher School of Economics, Russia

ICML, June 22, 2014

Summary

- Many tasks in Markov random fields (MRFs) are hard.
- Energy and probability of MRFs are tensors.
- Tensor Train (TT) decomposition: compact representation of high-dimensional tensors (Oseledets, 2011); efficient operations.
- We use TT-format for:
  - partition function (normalization constant);
  - marginal distributions;
  - MAP-inference.

MRFs and tensors

Potential

$$\left.\begin{aligned} \boldsymbol{E}(x_1, \ldots, x_n) &= \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell) \\ \widehat{\boldsymbol{P}}(x_1, \ldots, x_n) &= \prod_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell) \end{aligned}\right\} \quad \text{tensors (multidimensional arrays)}$$

Factor

$x_i \in \{1, \ldots, d\}$.

MAP-inference	$\Longleftrightarrow$	minimal element in $\boldsymbol{E}$
Partition function	$\Longleftrightarrow$	sum of all elements of $\widehat{\boldsymbol{P}}$

TT-format

TT-format for tensor $\boldsymbol{A}$:

$$\boldsymbol{A}(x_1, \dots, x_n) = \underbrace{G_1^{\boldsymbol{A}}[x_1]}_{1 \times \mathrm{r}_1(\boldsymbol{A})} \underbrace{G_2^{\boldsymbol{A}}[x_2]}_{\mathrm{r}_1(\boldsymbol{A}) \times \mathrm{r}_2(\boldsymbol{A})} \dots \underbrace{G_n^{\boldsymbol{A}}[x_n]}_{\mathrm{r}_{n-1}(\boldsymbol{A}) \times 1}$$

Terminology:

- $G_i^{\boldsymbol{A}}$ — TT-cores;
- $\mathrm{r}_i(\boldsymbol{A})$ — TT-ranks;
- $r(\boldsymbol{A}) = \max\limits_{i=1,\dots,n-1} r_i(\boldsymbol{A})$ — maximal TT-rank.

TT-format uses $O\left(ndr^2(\boldsymbol{A})\right)$ memory to store $O\left(d^n\right)$ elements.
Efficient only if TT-ranks are small.

Example

$$\boldsymbol{A}(x_1, x_2, x_3) = x_1 + x_2 + x_3,$$

$$\boldsymbol{A}(x_1, x_2, x_3) = G_1^{\boldsymbol{A}}[x_1] G_2^{\boldsymbol{A}}[x_2] G_3^{\boldsymbol{A}}[x_3],$$

Example

$$\boldsymbol{A}(x_1, x_2, x_3) = x_1 + x_2 + x_3,$$

$$\boldsymbol{A}(x_1, x_2, x_3) = G_1^{\boldsymbol{A}}[x_1] G_2^{\boldsymbol{A}}[x_2] G_3^{\boldsymbol{A}}[x_3],$$

$$G_1^{\boldsymbol{A}}[x_1] = \begin{bmatrix} x_1 & 1 \end{bmatrix} \quad G_2^{\boldsymbol{A}}[x_2] = \begin{bmatrix} 1 & 0 \\ x_2 & 1 \end{bmatrix} \quad G_3^{\boldsymbol{A}}[x_3] = \begin{bmatrix} 1 \\ x_3 \end{bmatrix}$$

Indeed:

$$A(x_1, x_2, x_3) = \begin{bmatrix} x_1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ x_3 \end{bmatrix} =$$

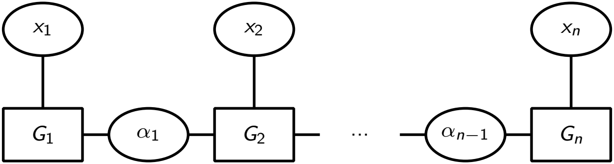
$$= \begin{bmatrix} x_1 + x_2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ x_3 \end{bmatrix} = x_1 + x_2 + x_3.$$

TT-format for probability

Consider TT-format for probability $\boldsymbol{P} = \frac{1}{Z}\widehat{\boldsymbol{P}}$ of MRF:

$$\begin{aligned}\boldsymbol{P}(\boldsymbol{x}) &= G_1[x_1]G_2[x_2]\cdots G_n[x_n] \\ &= \sum_{\alpha_1,\dots,\alpha_{n-1}} \underbrace{G_1[x_1](\alpha_1)G_2[x_2](\alpha_1,\alpha_2)\cdots G_n[x_n](\alpha_{n-1})}_{\boldsymbol{P}(\boldsymbol{x},\boldsymbol{\alpha})}.\end{aligned}$$

Chain-like model with hidden variables $\alpha_i = 1,\dots,r_i(\boldsymbol{P})$ is constructed.



Some efficient operations in the TT-format

Operation	Output rank
$\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B}$	$\mathrm{r}(\boldsymbol{A}) + \mathrm{r}(\boldsymbol{B})$
$\boldsymbol{C} = \boldsymbol{A} \odot \boldsymbol{B}$	$\mathrm{r}(\boldsymbol{A})\,\mathrm{r}(\boldsymbol{B})$
$\min \boldsymbol{A}$	–
$\operatorname{sum} \boldsymbol{A}$	–

TT-approach for MRFs

MAP-inference	$\Longleftrightarrow$	minimal element in $\boldsymbol{E}$
Partition function	$\Longleftrightarrow$	sum of all elements of $\widehat{\boldsymbol{P}}$

Both operations are provided by the TT-format.

Let's convert $\boldsymbol{E}$ and $\boldsymbol{P}$ into the TT-format.

Finding a TT-representation of an MRF

- TT-SVD (Oseledets, 2011): exact TT-representation but only for small tensors
  No, MRF tensor is too big.

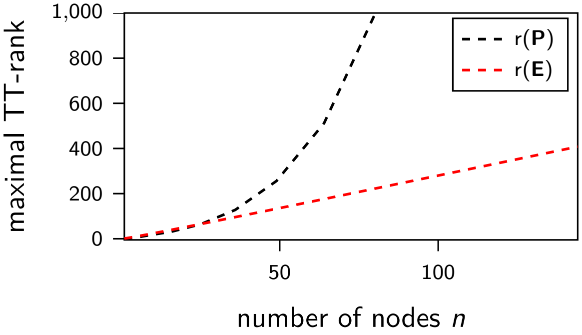
- AMEn-cross (Oseledets & Tyrtyshnikov, 2010): approximate TT-representation; uses only a small fraction of tensor's elements
  Possible, but there is also a much better way!

The algorithm & its theoretical guarantees

1. Convert the potentials $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$ (factors $\boldsymbol{\Psi}_\ell(\boldsymbol{x})$) into the TT-format.
2. Use the TT-operations: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x})$ $(\widehat{\boldsymbol{P}} = \bigodot_{\ell=1}^{m} \boldsymbol{\Psi}_\ell)$.

The algorithm & its theoretical guarantees

1. Convert the potentials $\boldsymbol{\Theta}_\ell(\boldsymbol{x})$ (factors $\boldsymbol{\Psi}_\ell(\boldsymbol{x})$) into the TT-format.
2. Use the TT-operations: $\boldsymbol{E}(\boldsymbol{x}) = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell(\boldsymbol{x})$ $\;(\widehat{\boldsymbol{P}} = \bigodot_{\ell=1}^{m} \boldsymbol{\Psi}_\ell)$.

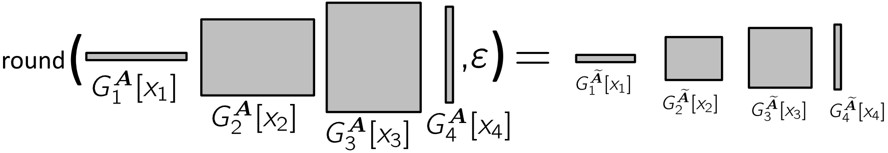


Theorem. The maximal TT-rank of $\boldsymbol{E}$ constructed by the algorithm is polynomially bounded: $\mathrm{r}(\boldsymbol{E}) \le d^{\frac{p}{2}} m$, where p is the order of MRF.

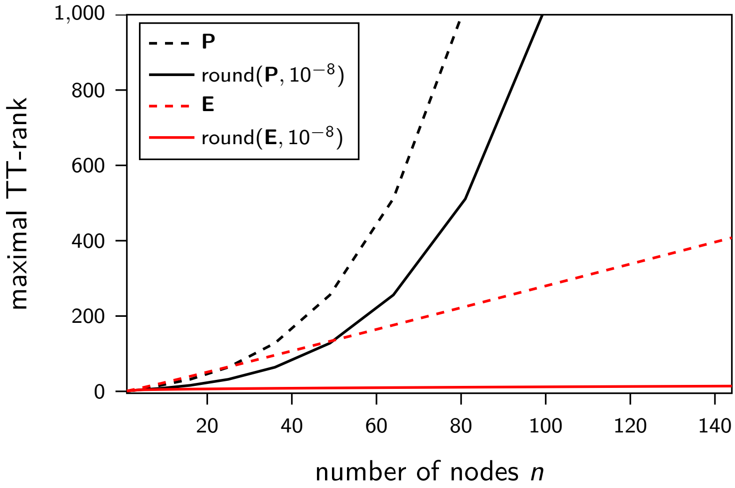
TT-rounding

TT-rounding procedure $\widetilde{\boldsymbol{A}} = \text{round}(\boldsymbol{A}, \varepsilon)$:

1. reduces TT-ranks
2. tensors are close



TT-rounding example



The algorithm motivation

- TT-ranks of $\widehat{\boldsymbol{P}}$ are exponential;
- We will compute partition function Z without explicitly building the TT- representation of $\widehat{\boldsymbol{P}}$.

Partition function estimation

$$
\begin{aligned}
\widehat{\boldsymbol{P}}(\boldsymbol{x}) &= \prod_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(\boldsymbol{x}) \\
&= \bigotimes_{\ell=1}^{m} \boldsymbol{\Psi}_\ell(\boldsymbol{x}) = \bigotimes_{\ell=1}^{m} \left(G_1^\ell[x_1] \cdots G_n^\ell[x_n] \right) \\
&= \left(G_1^1[x_1] \otimes \cdots \otimes G_1^m[x_1] \right) \cdots \left(G_n^1[x_n] \otimes \cdots \otimes G_n^m[x_n] \right).
\end{aligned}
$$

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

Finally,

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

The algorithm

$$Z = B_1 \cdots B_n,$$

Each matrix B_i is huge but can be exactly represented in the TT-format.

The algorithm:

1. $\boldsymbol{f}_1 := B_1$
2. $\boldsymbol{f}_2 := \text{round}(\boldsymbol{f}_1 B_2, \varepsilon)$
3. $\boldsymbol{f}_3 := \text{round}(\boldsymbol{f}_2 B_3, \varepsilon)$
4. ...
5. $\boldsymbol{f}_n := \text{round}(\boldsymbol{f}_{n-1} B_n, \varepsilon)$
6. $\widetilde{Z} := \boldsymbol{f}_n;$

Marginal distributions

Our approach can be generalized to the marginal distributions as well:

$$\hat{\boldsymbol{P}}_i(x_i) = B_1 \dots B_{i-1}\, A_i[x_i]\, B_{i+1} \dots B_n,$$

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

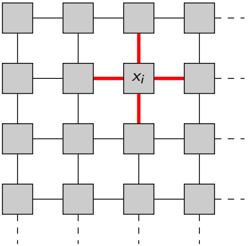
Spin glass model:

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

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

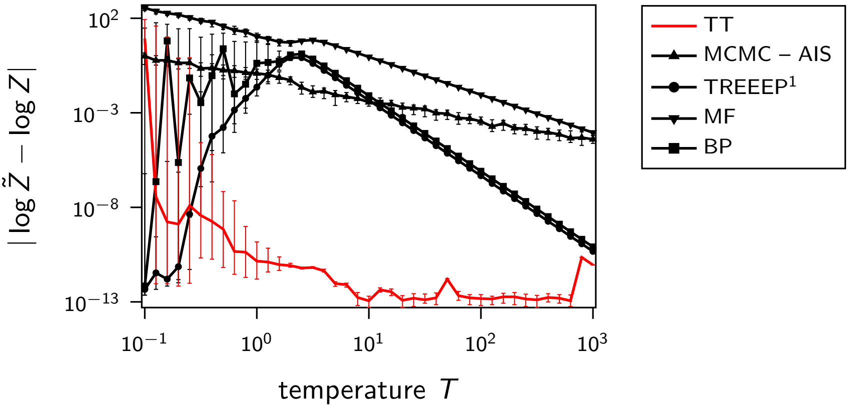
Notation:

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



We compare against methods from the LibDAI library (Mooij 2010).

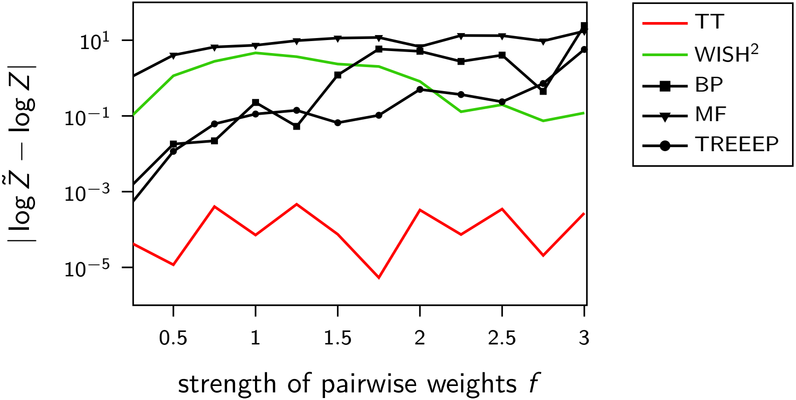
Comparison



Comparison on Ising model (all pairwise weights are equal $c_{ij} = 1$).

[1]Minka and Qi 2004.

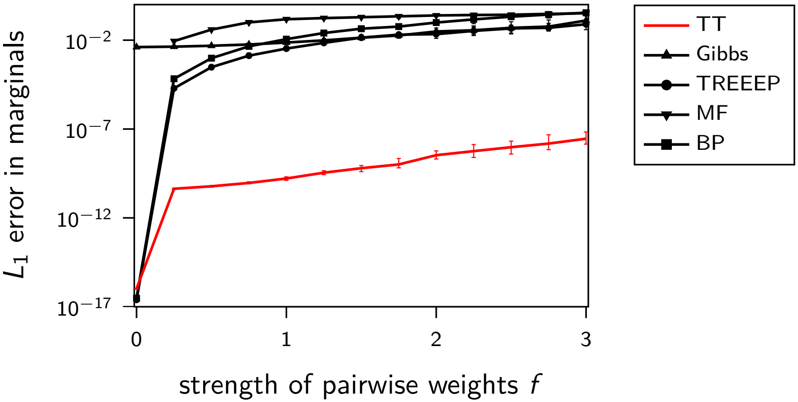
WISH



Comparison on the data from the WISH paper, $T = 1$, $c_{ij} \sim U[-f, f]$.

[2]Ermon et al. 2013.

Marginal distributions



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

Conclusion

Our contributions:

- Algorithm that finds an exact TT-representation of MRF energy;
- Algorithm that estimates the partition function and the marginals;
- Theoretical guaranties for the proposed algorithms.

Source code is availible online: `https://github.com/bihaqo/TT-MRF`.

See poster **S38** tonight!

References I

Ermon, S. et al. (2013). "Taming the Curse of Dimensionality: Discrete Integration by Hashing and Optimization". In: *International Conference on Machine Learning (ICML)*.

Minka, T. and Y. Qi (2004). "Tree-structured Approximations by Expectation Propagation". In: *Advances in Neural Information Processing Systems 16 (NIPS)*, pp. 193–200.

Mooij, J. M. (2010). "libDAI: A Free and Open Source C++ Library for Discrete Approximate Inference in Graphical Models". In: *Journal of Machine Learning Research* 11, pp. 2169–2173.