Putting MRFs on a Tensor Train

Alexander Novikov[1] NOVIKOV@BAYESGROUP.RU
Anton Rodomanov[1] ANTON.RODOMANOV@GMAIL.COM
Anton Osokin[1] OSOKIN@BAYESGROUP.RU
Dmitry Vetrov[1,2] VETROVD@YANDEX.RU
[1] Moscow State University, Moscow, Russia
[2] Higher School of Economics, Moscow, Russia

Abstract

In the paper we present a new framework for dealing with probabilistic graphical models. Our approach relies on the recently proposed Tensor Train format (TT-format) of a tensor that while being compact allows for efficient application of linear algebra operations. We present a way to convert the energy of a Markov random field to the TT-format and show how one can exploit the properties of the TT-format to attack the tasks of the partition function estimation and the MAP-inference. We provide theoretical guarantees on the accuracy of the proposed algorithm for estimating the partition function and compare our methods against several state-of-the-art algorithms.

1. Introduction

Discrete graphical models have become a popular tool for many applications in the fields of computer vision, machine learning, social networking, etc. One of their advantages is the ability to define the (unnormalized) joint distribution of thousands of variables in a compact form of the product of low-order factors. Such a form allows to model complex dependencies and makes exact or approximate inference possible.

In this paper we consider undirected graphical models (Markov random fields, MRFs) and touch on two inference problems: the estimation of the normalization constant (the partition function) and the search of the most probable configuration of the variables (the MAP-inference). When the factor-graph of dependencies of an MRF contains cycles these problems become challenging. Both problems are of great practical importance, e.g. because they are the key elements of the two popular methods that obtain the parameters of a graphical model given training data. Specifically, structured-output SVMs (Taskar et al., 2004; Tsochantaridis et al., 2005) require accurate and fast MAP-estimators and maximum likelihood training methods (Nowozin & Lampert, 2010) require good approximations of the partition function. Although a number of approximate methods have been proposed in recent years for both problems, they (especially the partition function estimation) are still far from the ultimate solution.

One way of defining the joint distribution of discrete variables is to express it as a multi-dimensional array of (unnormalized) probabilities. We refer to these arrays as tensors. The problems of the MAP-inference and the partition function estimation are then reduced to the search of the maximal element in the tensor and to the summation of all the elements of the tensor respectively. Direct storage (and processing) of tensors requires exponential amount of memory and computational effort. There exist several tensor decomposition methods that allow for storing a tensor in a format that requires less memory and provides efficient algorithms for performing algebraic operations on tensors.

In our paper we make use of the recently proposed Tensor Train (TT) decomposition approach (Oseledets, 2011). The TT-decomposition method converts a tensor into a special chain-like TT-format that allows to operate with the tensor in an efficient manner. Oseledets & Tyrtyshnikov (2010) show that to convert a tensor that is in some sense simple (i.e. with low TT-ranks) into the TT-format it suffices to process only a small fraction of its elements. In our paper we show that MRF energy tensors (the negative logarithms of the joint distributions) have low TT-ranks but the unnormalized distribution tensors do not. Converting the energy into the TT-format allows us to perform the MAP-inference using the techniques coming from linear algebra. To tackle the partition function estimation problem in the case when the unnormalized distribution has large TT-ranks

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

we exploit the factorization structure of MRFs. We construct the TT-decomposition of each factor individually and then combine them in such a way that allows to estimate the partition function without explicitly building the TT-representation of the unnormalized distribution. Our approach can be generalized for the problem of computing the marginal distributions.

We evaluate our methods on several datasets and compare them against several state-of-the-art techniques. We report a significant improvement in the problems of the estimation of the partition function and the marginal distributions and show comparable results in the MAP-inference problem.

Our main contributions are:

- The algorithm for obtaining the TT-decomposition of the MRF energy (the negative logarithm of the joint distribution). We prove that for low-order graphical models the energy tensor has a low TT-rank and derive upper bounds on it.
- The algorithm for estimating the partition function through the TT-decomposition of the MRF factors. The key feature of the algorithm is that it does not explicitly construct the TT-representation of the unnormalized joint distribution.
- Theoretical bounds on the accuracy of the partition function estimation.

The rest of the paper is organized as follows. We start with the review of the related works in sec. 2. We introduce our notation in sec. 3 and review the TT-decomposition approach in sec. 4. In sec. 5 we apply the TT-decomposition to MRFs and in sec. 6 we propose an algorithm for computing the partition function. Sec. 7 contains our experimental evaluation.

2. Related work

In this section we briefly review the related works divided into three groups: different formats for compact representation of tensors, usage of tensors in graphical models and different inference methods.

Compared to the classical tensor formats (the canonical format (Caroll & Chang, 1970) and the Tucker format (Tucker, 1966)), the TT-format can be computed via stable and robust algorithms and has no intrinsic curse of dimensionality. The Hierarchical Tucker (HT) format (Hackbusch & Kühn, 2009; Grasedyck, 2010) is another stable tensor factorization. The TT-format can be considered as the HT-format with a linear dimension tree, and this comes as an advantage from the algorithmic viewpoint: most of the algorithms are much simpler in the TT-format.

Tensor-based techniques are often mentioned in articles related to graphical models with respect to the problem of uncovering the unknown structure of the model. Jernite et al. (2013); Ishteva et al. (2013) make local decisions based on the properties of 4th order tensors (quartets). Song et al. (2013) attack the problem using the hierarchical tensor decomposition.

There are two popular ways of constructing a labeling given learned parameters, i.e. performing inference, in a graphical model: MAP-inference (mode of the posterior distribution) and max-marginal inference (each variable is set to the argmax of its marginal distribution). The MAP-inference (a.k.a. energy minimization) is rather well-studied. Recent experimental results of Kappes et al. (2013) show that for many problems current methods give satisfying results. Specifically, a lot of attention has been given to the case of pairwise Markov random fields. The max-marginal inference is based on the computation of the marginal distributions. The latter is closely related to the task of estimating the normalization constant (i.e. the partition function) given the unnormalized distribution of an MRF. We can outline several families of methods for the computation of the partition function and the marginal distributions: sampling techniques (e.g. Annealed Importance Sampling (Neal, 2001; Grosse et al., 2013) for the partition function and Gibbs sampling (Wainwright & Jordan, 2008) for the marginal distributions), message passing techniques (e.g. classic Loopy Belief Propagation (Kschischang et al., 2001) and its numerous modifications), KL-minimization based methods (e.g. Mean Field (Wainwright & Jordan, 2008) and Expectation Propagation (Minka & Qi, 2004)), graph decomposition-based techniques (e.g. Tree-Reweighted Message-passing (Wainwright et al., 2005)), MAP-inference based methods (e.g. randomized MAP-predictors (Hazan et al., 2013) and the recent WISH system (Ermon et al., 2013)).

3. Notation

In this paper we extensively use multi-dimensional arrays of real numbers. We refer to one-dimensional arrays as *vectors*, two-dimensional arrays as *matrices* and, finally, arrays of higher dimensionality as *tensors*. We use bold upper case letters (e.g. $\mathbf{A}$) for tensors, ordinary upper case letters (e.g. A) for matrices, and bold lower case letters (e.g. $\boldsymbol{a}$) for vectors.

We treat all arrays as functions of their indices: $\boldsymbol{a}(i) = a_i$, $A(x_1, x_2)$, $\mathbf{A}(\boldsymbol{x}) = \mathbf{A}(x_1, \ldots, x_n)$, where n equals the dimensionality of the tensor $\mathbf{A}$.

By $\|\cdot\|_F$ we denote the matrix Frobenius norm and its trivial generalization for tensors:

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

By $\|\cdot\|_2$ we denote the vector Euclidean norm and the matrix spectral norm: $\|A\|_2 = \max_{\boldsymbol{x} \neq 0} \frac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}$.

In our derivations we use several different matrix products. The operator "$\odot$" corresponds to the Hadamard product (entrywise product), the operator "$\otimes$" – to the Kronecker product, and the operator "$\cdot$" (which is often omitted) – to the regular matrix product.

4. TT-format

An n-dimensional tensor $\mathbf{A}$ is said to be represented in the TT-format if for each dimension $i = 1, \dots, n$ and for each possible value of the i-th dimension index $x_i = 1, \dots, d_i$ ($d = \max_{i=1,\dots,n} d_i$) there exists a matrix $G_i^{\mathbf{A}}[x_i]$ such that all the elements of $\mathbf{A}$ can be computed as the following matrix product:

$$\mathbf{A}(x_1, \dots, x_n) = G_1^{\mathbf{A}}[x_1] G_2^{\mathbf{A}}[x_2] \dots G_n^{\mathbf{A}}[x_n]. \qquad (1)$$

All the matrices $G_i^{\mathbf{A}}[x_i]$ related to the same dimension i are restricted to be of the same size $\mathrm{r}_{i-1}(\mathbf{A}) \times \mathrm{r}_i(\mathbf{A})$. The values $\mathrm{r}_0(\mathbf{A})$ and $\mathrm{r}_n(\mathbf{A})$ equal 1 in order to make matrix product (1) a number. In what follows we refer to the representation of a tensor in the TT-format as the *TT-representation* or the *TT-decomposition*. We refer to the sequence $\{\mathrm{r}_i(\mathbf{A})\}_{i=0}^n$ as the *TT-ranks* of the TT-representation of $\mathbf{A}$ and to its maximum as the *maximal TT-rank* of the TT-representation of $\mathbf{A}$: $\mathrm{r}(\mathbf{A}) = \max_{i=0,\dots,n} \mathrm{r}_i(\mathbf{A})$. The collections of the matrices corresponding to the same dimension (technically, 3-dimensional arrays) are called the *TT-cores*.

Oseledets (2011, Th. 2.1) shows that for an arbitrary tensor $\mathbf{A}$ a TT-representation exists but is not unique.

We use the symbols $G_i^{\mathbf{A}}[x_i](\alpha_{i-1}, \alpha_i)$ to denote the element of the matrix $G_i^{\mathbf{A}}[x_i]$ in the position (α_{i-1}, α_i). Equation (1) can be equivalently rewritten as the sum of the products of the elements of the TT-cores:

$$\mathbf{A}(\boldsymbol{x}) = \sum_{\alpha_0, \dots, \alpha_n} G_1^{\mathbf{A}}[x_1](\alpha_0, \alpha_1) \dots G_n^{\mathbf{A}}[x_n](\alpha_{n-1}, \alpha_n). \quad (2)$$

Representation of a tensor $\mathbf{A}$ via the explicit enumeration of all its elements requires to store $\prod_{i=1}^n d_i$ numbers compared to $\sum_{i=1}^n d_i \, \mathrm{r}_{i-1}(\mathbf{A}) \, \mathrm{r}_i(\mathbf{A})$ numbers if the tensor is stored in the TT-format. Thus, the TT-format is very efficient in terms of memory if the corresponding TT-ranks are small.

There exist several algorithms that can compute the TT-representation of a tensor. The TT-SVD algorithm (Oseledets, 2011) can find an exact representation but is suitable only for small dimensionality n. The recently proposed alternating minimal energy (AMEn-cross) algorithm (Dolgov & Savostyanov, 2013; Oseledets & Tyrtyshnikov, 2010) can construct an approximation of a tensor using only a small fraction of its elements.

Table 1. Efficient operations on tensors in the TT-format. For each operation we show its complexity and the maximal TT-rank of the result.

OPERATION	OUTPUT RANKS	COMPLEXITY
$\mathbf{C} = \mathbf{A} \cdot \mathrm{const}$	$\mathrm{r}(\mathbf{C}) = \mathrm{r}(\mathbf{A})$	$O(d\,\mathrm{r}(\mathbf{A}))$
$\mathbf{C} = \mathbf{A} + \mathrm{const}$	$\mathrm{r}(\mathbf{C}) = \mathrm{r}(\mathbf{A})+1$	$O(nd\,\mathrm{r}^2(\mathbf{A}))$
$\mathbf{C} = \mathbf{A} + \mathbf{B}$	$\mathrm{r}(\mathbf{C}) \leq \mathrm{r}(\mathbf{A})+\mathrm{r}(\mathbf{B})$	$O(nd\,\mathrm{r}^2(\mathbf{C}))$
$\mathbf{C} = \mathbf{A} \odot \mathbf{B}$	$\mathrm{r}(\mathbf{C}) \leq \mathrm{r}(\mathbf{A})\,\mathrm{r}(\mathbf{B})$	$O(nd\,\mathrm{r}^2(\mathbf{A})\,\mathrm{r}^2(\mathbf{B}))$
$\boldsymbol{c} = M\boldsymbol{b}$	$\mathrm{r}(\boldsymbol{c}) \leq \mathrm{r}(M)\,\mathrm{r}(\boldsymbol{b})$	$O(nd^2\mathrm{r}^2(M)\mathrm{r}^2(\boldsymbol{b}))$
$\mathrm{sum}\,\mathbf{A}$	–	$O(nd\,\mathrm{r}^2(\mathbf{A}))$
$\|\mathbf{A}\|_F$	–	$O(nd\,\mathrm{r}^3(\mathbf{A}))$
$\mathbf{C} = \mathrm{round}(\mathbf{A}, \varepsilon)$	$\mathrm{r}(\mathbf{C}) \leq \mathrm{r}(\mathbf{A})$	$O(nd\,\mathrm{r}^3(\mathbf{A}))$

An attractive property of the TT-format is the ability to efficiently perform several types of operations on tensors if they are in the TT-format: basic linear algebra operations, such as the addition of a constant and the multiplication by a constant, the summation and the entrywise product of tensors (the results of these operations are tensors in the TT-format generally with the increased TT-ranks); computation of global characteristics of a tensor, such as the sum of all elements and the Frobenius norm; and the important *rounding* operation. See table 1 for a review of the operations that we use in this work and the paper (Oseledets, 2011) for a detailed description.

The rounding operation builds a lower-rank approximation of a tensor. Given a tensor $\mathbf{A}$ in the TT-format and a precision parameter $\varepsilon \geq 0$ the rounding operation (we refer to it as the *TT-rounding* and denote by $\mathrm{round}(\mathbf{A}, \varepsilon)$) finds a tensor $\tilde{\mathbf{A}}$ that on the one hand is close to the tensor $\mathbf{A}$: $\|\mathbf{A} - \tilde{\mathbf{A}}\|_F \leq \varepsilon \|\mathbf{A}\|_F$, but on the other hand has minimal TT-ranks among all tensors $\mathbf{B} : \|\mathbf{A} - \mathbf{B}\|_F \leq \frac{\varepsilon}{\sqrt{n-1}} \|\mathbf{A}\|_F$. Although the TT-rounding is only suboptimal (in a sense that there are no guarantees on finding the minimal TT-ranks within the desired accuracy), in practice it often performs well and allows us to apply multiple operations to tensors while controlling the growth of the TT-ranks.

In addition to tensors one can also use the TT-format to operate on huge matrices and vectors. For a vector $\boldsymbol{b}$ with a mapping from its indices to n-dimensional vectors $\boldsymbol{y} = (y_1, \dots, y_n)$[1] we define a TT-representation of the vector $\boldsymbol{b}$ as a TT-representation of the tensor $\mathbf{B}$ where $\mathbf{B}(\boldsymbol{y}) = b_{\boldsymbol{y}}$.

Now we define a TT-representation of a matrix M. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be n-dimensional vectors corresponding to the row and column indices of the matrix M respectively. We can

[1] The length of the vector $\boldsymbol{b}$ should be equal to $\prod_{i=1}^n d_i$.

rearrange the elements of the matrix M into the tensor $\mathbf{M}$ and convert the latter into the TT-format:

$$\mathbf{M}((x_1,y_1),\dots,(x_n,y_n)) = G_1^{\mathbf{M}}[x_1,y_1]\dots G_n^{\mathbf{M}}[x_n,y_n]$$

where the matrices $G_i^{\mathbf{M}}[x_i,y_i]$, $i=1,\dots,n$, serve as the TT-cores with compound indices (x_i,y_i). In what follows we denote the elements of matrices in the TT-format without nested parentheses: $M(x_1,\dots,x_n;y_1,\dots,y_n)$. Note that a matrix in the TT-format is not restricted to be square. Although index-vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are of the same length the sizes of the domains of the dimensions can vary.

When both a matrix M and a vector $\boldsymbol{b}$ are represented in the TT-format, it is possible to efficiently compute the matrix-by-vector product $\boldsymbol{c} = M\boldsymbol{b}$. The result of this operation (the vector $\boldsymbol{c}$) will be represented in the TT-format as well.

Matrix operations on top of the TT-representations allow us to apply different linear algebra methods on a large scale. For example, we can use the DMRG method based on the search of the smallest eigenvalue in a matrix (Khoromskij & Oseledets, 2010) to do an approximate MAP-inference in an MRF.

5. Markov random fields as tensors

Let $\mathcal{G} = (\mathcal{V},\mathcal{E})$ be a hypergraph. The set of nodes $\mathcal{V}$ and the set of hyperedges $\mathcal{E}$ are both finite. Let all the nodes be indexed from 1 to n and all the hyperedges from 1 to m.

With each node $i=1,\dots,n$ we associate a variable x_i that takes values from a domain $\mathcal{X}_i = \{1,\dots,d_i\}$. For each hyperedge $\ell = 1,\dots,m$ we denote the tuple of the incident variables by $\boldsymbol{x}^\ell$. The ℓ-th *potential* $\boldsymbol{\Theta}_\ell$ is a real-valued function defined on the joint domain of the variables $\boldsymbol{x}^\ell$.

The *energy* function of the Markov random field (MRF) defined on the hypergraph $\mathcal{G}$ is the sum of all the potentials: $\mathbf{E}(\boldsymbol{x}) = \sum_{\ell=1}^m \boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$. The exponentiation of the negative energy defines the *unnormalized (Gibbs) distribution* $\widehat{\mathbf{P}}(\boldsymbol{x}) = \exp(-\mathbf{E}(\boldsymbol{x}))$. We use the symbol Z to denote the normalization constant (which is frequently called the *partition function*), i.e. $Z = \sum_{\boldsymbol{x}} \widehat{\mathbf{P}}(\boldsymbol{x})$. The functions $\boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell) = \exp(-\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell))$ are called the *factors* of MRF.

Both the energy and the unnormalized probability can be considered as n-dimensional tensors in which the values of the variables $\boldsymbol{x}$ act as indices. The potentials and the factors become n-dimensional tensors as well if we add non-essential variables for non-existing dimensions: $\boldsymbol{\Theta}_\ell$, $\boldsymbol{\Psi}_\ell$. Using this notation we can write the following identities: $\mathbf{E} = \sum_{\ell=1}^m \boldsymbol{\Theta}_\ell$ and $\widehat{\mathbf{P}} = \bigodot_{\ell=1}^m \boldsymbol{\Psi}_\ell$.

The factors (or the potentials) allow to store the joint distribution (or the energy) in memory in a compact form. The TT-format is an alternative way to compactly represent a multi-dimensional array. In the next section we explore whether it is possible to build an accurate TT-representation of the unnormalized distribution and the energy of an MRF.

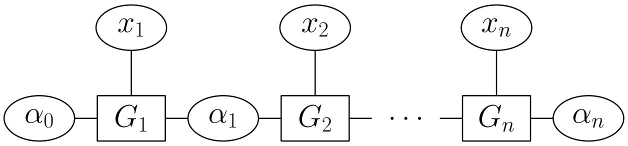


Figure 1. A graphical model obtained by the TT-decomposition of the joint distribution tensor.

The TT-representation of the joint distribution tensor $\mathbf{P} = \frac{1}{Z}\widehat{\mathbf{P}}$ has a special interpretation. The inner variables α_i, $i=0,\dots,n$ can be viewed as hidden variables in the chain model (see fig. 1). Marginalization w.r.t. them gives us the probability distribution w.r.t. the variables $\boldsymbol{x}$:

$$\mathbf{P}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}} \mathbf{P}(\boldsymbol{x},\boldsymbol{\alpha}), \tag{3}$$

where $\mathbf{P}(\boldsymbol{x},\boldsymbol{\alpha})$ is the joint distribution of the variables $\boldsymbol{x}$ and $\boldsymbol{\alpha}$. The domain size of each α_i is equal to the corresponding TT-rank $\mathrm{r}_i(\mathbf{P})$.

5.1. The TT-format for MRF energy

In this section we present the general algorithm that converts the energy tensor $\mathbf{E}$ into the TT-format. The proposed algorithm is much faster and more accurate than the direct application of the AMEn algorithm.

The algorithm consists of the following three main steps.

Step 1. For each potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$, which we treat as a tensor indexed only by the variables $\boldsymbol{x}^\ell$, we find its TT-representation. Potentials are usually tensors of a relatively small dimensionality which allows for computing the required TT-decomposition using the TT-SVD algorithm. In some cases it is possible to explicitly construct the TT-representation for a potential (several examples are provided in the supplementary material).

Step 2. Once all the potential tensors $\boldsymbol{\Theta}_\ell$ are in the TT-format, we add non-essential variables in order to make each of them n-dimensional. These non-essential variables can be added to TT-representations constructively.

For clarity we describe this procedure with the following example. Suppose that our MRF has 5 nodes in total with their associated variables x_1,x_2,x_3,x_4,x_5 and our potential $\boldsymbol{\Theta}_\ell(\boldsymbol{x}^\ell)$ depends only on variables x_1,x_2,x_4 (i.e. $\boldsymbol{x}^\ell = (x_1,x_2,x_4)$). After the first step of the algorithm we have the following TT-representation of our potential:

$$\boldsymbol{\Theta}_\ell(x_1,x_2,x_4) = G_1^{\boldsymbol{\Theta}_\ell}[x_1]G_2^{\boldsymbol{\Theta}_\ell}[x_2]G_4^{\boldsymbol{\Theta}_\ell}[x_4], \tag{4}$$

where the cores $G_1^{\boldsymbol{\Theta}_\ell}[x_1]$, $G_2^{\boldsymbol{\Theta}_\ell}[x_2]$, $G_4^{\boldsymbol{\Theta}_\ell}[x_4]$ are matrices of sizes $\mathrm{r}_0 \times \mathrm{r}_1$, $\mathrm{r}_1 \times \mathrm{r}_2$, $\mathrm{r}_3 \times \mathrm{r}_4$, respectively, and due to the matrix multiplication operation $\mathrm{r}_0 = \mathrm{r}_4 = 1$, $\mathrm{r}_2 = \mathrm{r}_3$. To introduce non-essential variables we need to define the missing cores as the identity matrices: $G_3^{\boldsymbol{\Theta}_\ell}[x_3] \equiv I_{\mathrm{r}_2} = I_{\mathrm{r}_3}$ and $G_5^{\boldsymbol{\Theta}_\ell}[x_5] \equiv I_{\mathrm{r}_4} = I_1$.[2] Note that this procedure does not increase the maximal TT-rank of $\boldsymbol{\Theta}_\ell$.

Step 3. By exploiting the summation operation of the TT-format we combine all the tensors obtained after step 2 into the energy tensor

$$\mathbf{E} = \sum_{\ell=1}^{m} \boldsymbol{\Theta}_\ell. \tag{5}$$

The following theorem provides an upper bound on the maximal TT-rank of the tensor constructed by the described algorithm (the proof is provided in the supplementary material).

Theorem 1. *If the order of each potential $\boldsymbol{\Theta}_\ell$, $\ell = 1, \dots, m$ does not exceed p, then the aforementioned algorithm constructs a TT-representation for the energy $\mathbf{E}$ in such a way that its maximal TT-rank is polynomially bounded:*

$$\mathrm{r}(\mathbf{E}) \leq d^{\frac{p}{2}} \cdot m, \tag{6}$$

where each variable x_i takes at most d possible values.

The algorithm often allows us to construct an accurate TT-decomposition of the energy tensor. Generally, after step 3 we can apply the rounding procedure to the energy tensor $\mathbf{E}$ in order to reduce its TT-ranks and to get a more compact representation, if possible.

5.2. Probability

The algorithm described in section 5.1 can easily be modified to construct a TT-representation for the unnormalized probability $\widehat{\mathbf{P}}$.

All the steps of the algorithm are essentially the same as those from the previous section. The difference is that instead of the potentials $\boldsymbol{\Theta}_\ell$ we operate on the factors $\boldsymbol{\Psi}_\ell$ and in step 3 instead of the summation we compute the entrywise product

$$\widehat{\mathbf{P}} = \bigodot_{\ell=1}^{m} \boldsymbol{\Psi}_\ell. \tag{7}$$

Note that this algorithm computes a precise TT-representation of $\widehat{\mathbf{P}}$. However, the TT-ranks of $\widehat{\mathbf{P}}$ are exponential in the number of nodes, because the TT-ranks are multiplied within the entrywise product operation. Thus, it becomes impossible to use the TT-format for $\widehat{\mathbf{P}}$ in large problems. The TT-ranks remain huge even after applying the TT-rounding operation, so an accurate low-rank TT-representation of the unnormalized distribution probably does not exist. We illustrate this effect experimentally on fig. 2. See sec. 7.1 for the detailed description of the experiment.

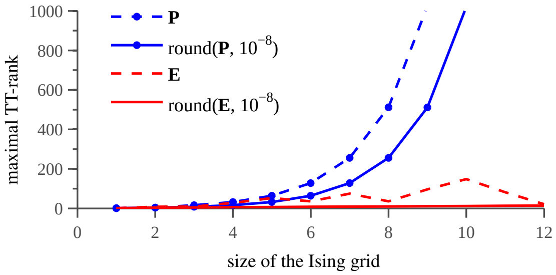


Figure 2. The maximal TT-ranks of TT-representations constructed for the energy tensor $\mathbf{E}$ and the corresponding unnormalized probability tensor $\widehat{\mathbf{P}}$ for homogeneous Ising models with temperature 10 and pairwise weight 1. Details are in section 7.1.

6. Partition function

A straightforward approach to compute the partition function is to represent the entire unnormalized probability tensor $\widehat{\mathbf{P}}$ in the TT-format and compute the sum of its elements. However, the tensor $\widehat{\mathbf{P}}$ turns out to have huge TT-ranks and therefore its TT-representation does not fit into memory. An attempt to approximate the tensor $\widehat{\mathbf{P}}$ with another tensor that has moderate TT-ranks leads to large inaccuracies. On the other hand, each individual factor $\boldsymbol{\Psi}_\ell$ has low TT-ranks and can be represented in the TT-format exactly. In this section we propose an algorithm that estimates the partition function by using only the TT-representations of the factors $\{\boldsymbol{\Psi}_\ell\}_{\ell=1}^m$ without building the TT-representation of the entire unnormalized probability tensor $\widehat{\mathbf{P}}$.

6.1. The algorithm

We assume that all the factors are already in the TT-format (see section 5.2 for details):

$$\boldsymbol{\Psi}_\ell(\boldsymbol{x}^\ell) = \boldsymbol{\Psi}_\ell(\boldsymbol{x}) = G_1^{\boldsymbol{\Psi}_\ell}[x_1] \dots G_n^{\boldsymbol{\Psi}_\ell}[x_n]. \tag{8}$$

Hereinafter we use $G_i^\ell[x_i](\alpha_{i-1}^\ell, \alpha_i^\ell)$ as a shorthand for $G_i^{\boldsymbol{\Psi}_\ell}[x_i](\alpha_{i-1}^{\boldsymbol{\Psi}_\ell}, \alpha_i^{\boldsymbol{\Psi}_\ell})$ to lighten the notation.

Recall the definition of the partition function Z:

$$Z = \sum_{\boldsymbol{x}} \prod_{\ell=1}^{m} \underbrace{\boldsymbol{\Psi}_\ell(\boldsymbol{x})}_{\in \mathbb{R}} = \sum_{x_1,\dots,x_n} \bigotimes_{\ell=1}^{m} \left(G_1^\ell[x_1] \dots G_n^\ell[x_n]\right).$$

Here we use the fact that the Kronecker product "$\otimes$" and the regular product coincide when applied to numbers.

Using the mixed-product property of the Kronecker prod-

[2] By I_k we denote the $k \times k$ identity matrix.

Algorithm 1 Compute the partition function Z

```
Input: factors $\mathbf{\Psi}_1, \dots, \mathbf{\Psi}_m$, rounding precision $\varepsilon$
Output: $\tilde{Z} \approx Z$
for $\ell := 1$ to $m$ do
    Find TT-cores $G_1^\ell, \dots, G_n^\ell$ for $\mathbf{\Psi}_\ell$
end for
Initialize $\boldsymbol{f}_{n+1} := 1$
for $i := n$ downto $1$ do
    Initialize $B_i := 0$
    for $x_i := 1$ to $d_i$ do
        Construct TT-matrix $A_i[x_i] = \bigotimes_{\ell=1}^{m} G_i^\ell[x_i]$
        $B_i := B_i + A_i[x_i]$
    end for
    $\overline{\boldsymbol{f}_i} := B_i \cdot \boldsymbol{f}_{i+1}$
    $\boldsymbol{f}_i := \mathrm{round}(\overline{\boldsymbol{f}_i}, \varepsilon)$
end for
$\tilde{Z} := \boldsymbol{f}_1$
```

uct $AC \otimes BD = (A \otimes B)(C \otimes D)$ we can rewrite Z as

$$Z = \sum_{x_1,\dots,x_n} \left(G_1^1[x_1] \otimes \dots \otimes G_1^m[x_1]\right) \dots \left(G_n^1[x_n] \otimes \dots \otimes G_n^m[x_n]\right).$$

Denote the Kronecker product of matrices $G_i^\ell[x_i]$, $\ell = 1, \dots, m$ by $A_i[x_i]$:

$$A_i[x_i] = G_i^1[x_i] \otimes \dots \otimes G_i^m[x_i].$$

The matrix $A_i[x_i]$ for any fixed x_i is of size $(\mathrm{r}_{i-1}(\mathbf{\Psi}_1) \dots \mathrm{r}_{i-1}(\mathbf{\Psi}_m)) \times (\mathrm{r}_i(\mathbf{\Psi}_1) \dots \mathrm{r}_i(\mathbf{\Psi}_m))$ and consists of the following elements:

$$A_i[x_i](\alpha_{i-1}^1, \dots, \alpha_{i-1}^m; \alpha_i^1, \dots, \alpha_i^m) = \\ = G_i^1[x_i](\alpha_{i-1}^1, \alpha_i^1) \dots G_i^m[x_i](\alpha_{i-1}^m, \alpha_i^m).$$

So the matrix $A_i[x_i]$ can be treated as a matrix in the TT-format with all TT-ranks equal to 1 (since $G_i^\ell[x_i](\alpha_{i-1}^\ell, \alpha_i^\ell)$ is a 1×1 matrix).

Now the partition function Z can be expressed as a product of n matrices:

$$Z = \sum_{x_1,\dots,x_n} A_1[x_1] \dots A_n[x_n] = \\ = \left(\sum_{x_1} A_1[x_1]\right) \dots \left(\sum_{x_n} A_n[x_n]\right) = B_1 \dots B_n,$$

where $B_i = \sum_{x_i=1}^{d_i} A_i[x_i]$. A matrix B_i can be represented in the TT-format as a sum of d_i matrices of TT-rank 1. Therefore the maximal TT-rank of B_i does not exceed d_i and so we can store the TT-representation of the matrix B_i in memory.

We have rewritten Z as a product of n matrices in the TT-format with the maximal TT-rank at most d. Note that B_1 and B_n are row and column vectors respectively, so $B_1 \dots B_n$ is a number.

To compute Z we build the matrices B_i in the TT-format and then multiply them one by one. During the process we apply the TT-rounding procedure after each multiplication to keep the maximal TT-rank of the intermediate product as small as possible. The resulting algorithm is summarized as algorithm 1. One can vary performance vs. accuracy trade-off by changing the rounding precision parameter ε.

In addition to the partition function Z the described approach allows one to efficiently compute the unnormalized marginal distributions of the variables. Specifically, the unnormalized unary marginals $\hat{p}_i(x_i)$ can be computed as follows: $\hat{p}_i(x_i) = B_1 \dots B_{i-1} A_i[x_i] B_{i+1} \dots B_n$. Note that all the products $B_1 \dots B_{i-1}$ and $B_{i+1} \dots B_n$ for $i = 1, \dots, n$ can be pre-computed with only 2 passes through the data. If we additionally pre-compute all the products $B_i \dots B_j$ for $1 \le i < j \le n$, we will be able co compute the unnormalized marginal distribution for an arbitrary subset of variables.

6.2. Analysis of the accuracy of algorithm 1

In this section we provide theoretical guaranties on the accuracy of the partition function estimation via algorithm 1.

Denote the approximation of the product of the matrices $\{B_j\}_{j=i}^n$ in the TT-format by $\boldsymbol{f}_i$. We have $\boldsymbol{f}_n = B_n$ and $\tilde{Z} = \boldsymbol{f}_1$. Using the matrix-by-vector product and the TT-rounding procedure the algorithm sequentially computes $\boldsymbol{f}_i = \mathrm{round}(B_i \boldsymbol{f}_{i+1}, \varepsilon)$, where the precision ε of the TT-rounding controls the relative accuracy: $\|B_i \boldsymbol{f}_{i+1} - \boldsymbol{f}_i\|_2 \le \varepsilon \|B_i \boldsymbol{f}_{i+1}\|_2$. Note that the TT-rounding procedure guarantees the previous inequality for the Frobenius norm instead of the 2-norm. However, here both $B_i \boldsymbol{f}_{i+1}$ and $\boldsymbol{f}_i$ are vectors and so the Frobenius norm and the 2-norm coincide.

The next theorem contains the main result on the accuracy of algorithm 1. After the theorem we provide a more interpretable corollary.

Theorem 2. *For an MRF and a rounding parameter $\varepsilon \ge 0$ the absolute error of the partition function estimation $\tilde{Z}$ computed by algorithm 1 is bounded as follows:*

$$\left|Z - \tilde{Z}\right| \le \|B_1\|_2 \dots \|B_{n-2}\|_2 \cdot \|B_{n-1}\boldsymbol{f}_n - \boldsymbol{f}_{n-1}\|_2 + \\ + \|B_1\|_2 \dots \|B_{n-3}\|_2 \cdot \|B_{n-2}\boldsymbol{f}_{n-1} - \boldsymbol{f}_{n-2}\|_2 + \dots + \\ + \|B_1 \boldsymbol{f}_2 - \boldsymbol{f}_1\|_2 . \tag{9}$$

The proof of the theorem is provided in the supplementary material. Note that all the items on the right-hand side are computable.

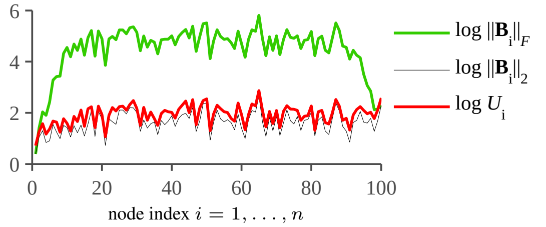


Figure 3. Comparison of the 2-norm and the Frobenius norm of the matrix B_i against the upper bound U_i. The plot was constructed for a heterogeneous Ising model of size 10.

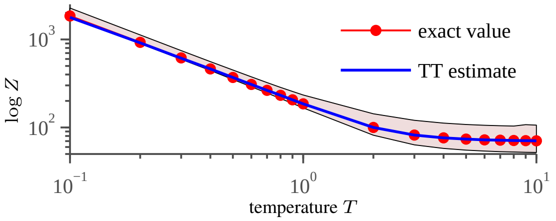


Figure 5. Confidence bounds on the computation of the partition function Z obtained by theorem 2 and upper bound (11). The details are provided in section 7.2.

Corollary 1. *For an MRF and a rounding parameter $\varepsilon \geq 0$ the absolute error of the partition function estimation $\tilde{Z}$ computed by algorithm 1 is bounded as follows:*

$$\left|Z - \tilde{Z}\right| \leq \|B_1\|_2 \ldots \|B_n\|_2 \left((1+\varepsilon)^{n-1} - 1\right). \quad (10)$$

Although the bound from the corollary is less tight, it allows one to find a sufficient ε for the target accuracy.

To use theorem 2 we need to evaluate the 2-norms of some vectors and matrices in the TT-format. The vector 2-norm coincides with the Frobenius norm of the corresponding tensor, so one can compute $\|B_i \boldsymbol{f}_{i+1} - \boldsymbol{f}_i\|_2$ using the Frobenius norm operation of the TT-format. For matrices the direct computation of the 2-norm in the TT-format is difficult. However, it can be bounded from above by the Frobenius norm or empirically tighter bound that uses the specific structure of the matrix B_i:

$$\begin{aligned} \|B_i\|_2 &= \left\|\sum_{x_i} G_i^1[x_i] \otimes \ldots \otimes G_i^m[x_i]\right\|_2 \leq \\ &\leq \sum_{x_i} \left\|G_i^1[x_i] \otimes \ldots \otimes G_i^m[x_i]\right\|_2 = \\ &= \sum_{x_i} \left\|G_i^1[x_i]\right\|_2 \ldots \left\|G_i^m[x_i]\right\|_2 = U_i. \end{aligned} \quad (11)$$

Here we have used the following identity: $\|A \otimes B\|_2 = \|A\|_2 \, \|B\|_2$. Note that here $G_i^\ell[x_i]$ for all $\ell = 1, \ldots, m$ are ordinary (small) matrices, so we can easily compute their 2-norms.

In fig. 3 we experimentally compare the values of the Frobenius norm $\|B_i\|_F$, the 2-norm $\|B_i\|_2$, and the suggested upper bound U_i. For a specific Ising model for each node i we report the logarithms of the three values.

7. Experiments

In all our experiments we use a non-optimized MATLAB implementation[3] of our methods. For operations related to the TT-format we use the TT-Toolbox[4] implemented in MATLAB as well. In our evaluation we mainly use the homogeneous and heterogeneous Ising models, and we refer to the supplementary material for the detailed description of the experimental setup.

7.1. TT-ranks of energies and distributions

In this experiment we illustrate the claims made in sec. 5.1 and 5.2 about the TT-ranks of the energy and the unnormalized distribution tensors constructed for an MRF. In fig. 2 for each size of the Ising model we report the maximal TT-ranks of both the energy and the unnormalized probability for the cases when they are represented in the TT-format exactly and after the TT-rounding procedure with precision 10^{-8}. As was claimed, for the energies the TT-ranks grow slowly with the increase of the size of the model as opposed to the probabilities. The exact TT-representations of the probability tensors of sizes 11 and 12 did not fit into 8GB of memory.

7.2. Partition functions

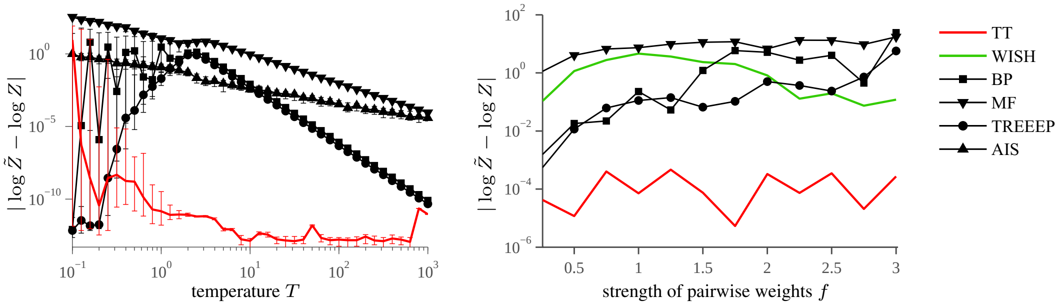
In this experiment we evaluate algorithm 1 for computing the partition function of an MRF.

First we compare our method (TT) against the following methods implemented in the LibDAI system (Mooij, 2010): Belief Propagation (BP) (Kschischang et al., 2001), Tree Expectation Propagation (TREEEP) (Minka & Qi, 2004), and the Mean Field method (MF) (Wainwright & Jordan, 2008). In addition, we compare our method against the annealed importance sampling method (AIS) (Neal, 2001) as a representative of the family of MCMC methods. The results are shown in plot 4a. For AIS method we select parameters (1000 intermediate distributions, 70 samples each) to maximize the accuracy achieved within 60 seconds per model. The ground-truth values are computed using the junction tree algorithm. For each value of temperature T we generate 50 homogeneous 10×10 Ising models and report the absolute error of the logarithm of the computed partition functions (we show the median, lower and upper quartiles).

In the second experiment we compare our method with the recently proposed WISH method (Ermon et al., 2013) on a dataset taken form their paper. The results of the comparison are presented in plot 4b. Here pairwise weights are generated uniformly from $[-f, f]$ with parameter f varying from 0.25 to 3.

[3] https://github.com/bihaqo/TT-MRF

[4] http://spring.inm.ras.ru/osel/download/tt22.zip



(a) (b)

Figure 4. The computation of the partition function Z. Plot (a) shows the comparison of our method (TT) against competitors on a set of homogeneous Ising models at different values of the temperature. Plot (b) shows the comparison of our method against the WISH algorithm on a set of heterogeneous Ising models. In both plots we report the errors, so the lower, the better.

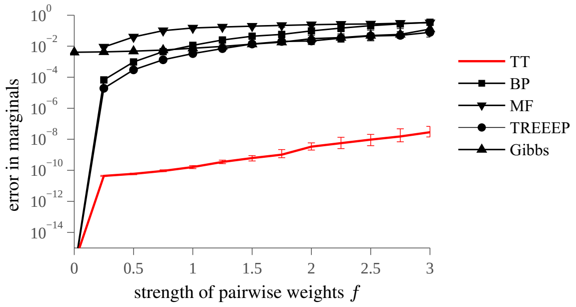


Figure 6. The computation of the marginal distributions of the individual variables. We use a set of heterogeneous Ising models and report the mean absolute error of the marginal for the "+1" class (the lower, the better).

In this experiment the running time of our non-optimized MATLAB implementation of the TT method was at most 53 seconds (32 seconds on average). The running times of the BP, MF, and TREEEP methods were 0.018, 0.05, and 0.1 seconds correspondingly. The WISH system was run on a cluster with timeouts of 15 minutes on each core.

Finally, fig. 5 presents the confidence bounds obtained by upper-bounding the result of theorem 2 using eq. (11) and taking the logarithm of the inequality.

7.3. Unary marginals

In this experiment we evaluate the ability of our method to compute the marginal distributions. We compare our method (TT) against Belief Propagation (BP) (Kschischang et al., 2001), Tree Expectation Propagation (TREEEP) (Minka & Qi, 2004), Mean Field (MF) (Wainwright & Jordan, 2008), and Gibbs sampling (Wainwright & Jordan, 2008) all implemented in the LibDAI system. The results of the comparison are presented in fig. 6. The details of the experimental setup are in the supplementary material.

7.4. MAP

Table 2. The relative value of the energy for the MAP-inference method based on the TT-decomposition. The lower bound obtained by TRW-S is 0%, the energy of the TRW-S primal solution is 100%.

PROBLEM	n	d	TT ENERGY
GEO-SURF-3/GM6	320	3	48.41%
GEO-SURF-3/GM20	348	3	95.83%
GEO-SURF-3/GM203	187	3	98.69%
GEO-SURF-7/GM11	125	7	1769.65%
MATCHING/MATCHING1	19	19	135.47%

In this experiment we evaluate the algorithm for finding the minimal element in the energy tensor that corresponds to MAP-inference. To find the minimal element in a tensor, which is in the TT-format, we construct a diagonal matrix in the TT-format containing all the elements of the tensor and use the DMRG algorithm to find its minimal eigenvalue (Khoromskij & Oseledets, 2010). We run the method on several real-world problems from a recent benchmark (Kappes et al., 2013) and compare it against the popular TRW-S algorithm (Kolmogorov, 2006). Some of the results are reported in table 2. The DMRG algorithm shows comparable results and on geo-surf-3 problems almost uniformly outperforms TRW-S.

8. Conclusion

In the paper we show how the modern tensor decomposition framework can be used for solving important problems arising in probabilistic graphical models. Although we consider only low-order models, the framework can further be generalized for many important high-order cases. Another direction for future work could be the search of a new parametrization of graphical models, different from the state-of-the-art log-linear one, with respect to which the derivatives of the log-partition function can be computed with high accuracy through the TT-decomposition.

Acknowledgements. We would like to thank Ivan Oseledets for valuable discussions and Stefano Ermon for providing the results of the WISH method. This work was supported by RFBR (projects 12-01-00938 and 12-01-33085).

References

Caroll, J. D. and Chang, J. J. Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart-Young decomposition. *Psychometrika*, 35:283–319, 1970.

Dolgov, S. V. and Savostyanov, D. V. Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems. arXiv preprint 1304.1222, 2013.

Ermon, S., Gomes, C., Sabharwal, A., and Selman, B. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *International Conference on Machine Learning (ICML)*, 2013.

Grasedyck, L. Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.*, 31:2029–2054, 2010.

Grosse, R., Maddison, C., and Salakhutdinov, R. Annealing between distributions by averaging moments. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pp. 2769–2777. 2013.

Hackbusch, W. and Kühn, S. A new scheme for the tensor representation. *J. Fourier Anal. Appl.*, 15:706–722, 2009.

Hazan, T., Maji, S., and Jaakkola, T. On sampling from the Gibbs distribution with random maximum a posteriori perturbations. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pp. 1268–1276. 2013.

Ishteva, M., Park, H., , and Song, L. Unfolding latent tree structures using 4th order tensors. In *International Conference on Machine Learning (ICML)*, 2013.

Jernite, Y., Halpern, Y., and Sontag, D. Discovering hidden variables in noisy-or networks using quartet tests. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pp. 2355–2363. 2013.

Kappes, J., Andres, B., Hamprecht, F., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B., Lellmann, J., Komodakis, N., and Rother, C. A comparative study of modern inference techniques for discrete energy minimization problems. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1328–1335, 2013.

Khoromskij, B. N. and Oseledets, I. V. DMRG+QTT approach to computation of the ground state for the molecular Schrödinger operator. Preprint 69, MPI MIS, Leipzig, 2010.

Kolmogorov, V. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(10): 1568–1583, 2006.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

Minka, T. and Qi, Y. Tree-structured approximations by expectation propagation. In *Advances in Neural Information Processing Systems 16 (NIPS)*, pp. 193–200. 2004.

Mooij, J. M. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, August 2010.

Neal, R. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

Nowozin, S. and Lampert, C. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3–4):185–365, 2010.

Oseledets, I. V. Tensor-Train decomposition. *SIAM J. Scientific Computing*, 33(5):2295–2317, 2011.

Oseledets, I. V. and Tyrtyshnikov, E. E. TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.*, 432(1):70–88, 2010.

Song, L., Ishteva, M., Parikh, A., Xing, E., and Park, H. Hierarchical tensor decomposition of latent tree graphical models. In *International Conference on Machine Learning (ICML)*, 2013.

Taskar, B., Guestrin, C., and Koller, D. Max-Margin Markov networks. In *Advances in Neural Information Processing Systems 16 (NIPS)*, pp. 25–32. 2004.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, 2005.

Tucker, L. R. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

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

Wainwright, M. J., Jaakkola, T., and Willsky, A. S. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335, 2005.