Stochastic Discrete Clenshaw-Curtis Quadrature

Nico Piatkowski NICO.PIATKOWSKI@TU-DORTMUND.DE
Artificial Intelligence Group, TU Dortmund, Germany

Katharina Morik KATHARINA.MORIK@TU-DORTMUND.DE
Artificial Intelligence Group, TU Dortmund, Germany

Abstract

The partition function is fundamental for probabilistic graphical models—it is required for inference, parameter estimation, and model selection. Evaluating this function corresponds to discrete integration, namely a weighted sum over an exponentially large set. This task quickly becomes intractable as the dimensionality of the problem increases. We propose an approximation scheme that, for any discrete graphical model whose parameter vector has bounded norm, estimates the partition function with arbitrarily small error. Our algorithm relies on a near minimax optimal polynomial approximation to the potential function and a Clenshaw-Curtis style quadrature. Furthermore, we show that this algorithm can be randomized to split the computation into a high-complexity part and a low-complexity part, where the latter may be carried out on small computational devices. Experiments confirm that the new randomized algorithm is highly accurate if the parameter norm is small, and is otherwise comparable to methods with unbounded error.

1. Introduction

Graphical models serve as the underlying framework of various machine learning techniques and facilitate important real world applications from computer vision, computational biology, signal processing and natural language processing, to mention just a few (Pearl, 1988; Lauritzen, 1996; Koller & Friedman, 2009). When using these models, a central problem is that of computing the partition function, since it is required for computing probabilities from the model and it plays a fundamental role in marginal inference (Jaakkola & Jordan, 1996), maximum-likelihood and maximum-a-posteriori parameter estimation (Wainwright & Jordan, 2008), and model selection (Cover & Thomas, 2006; Grünwald, 2007). Unfortunately, exact computation of the partition function is an intractable (**#P**-complete) problem for many graphical models of interest, such as those involving a non-tree conditional independence structure (short: *structure*) or high-order cliques/factors. Moreover, chances to find polynomial time algorithms for these tasks are rather low, unless **P = NP**. For some special cases, polynomial-time algorithms are known (Schraudolph & Kamenetsky, 2008; Goldberg & Jerrum, 2013). However, significant research efforts have been conducted in order to derive approximations for more general cases.

There are two well established general approaches to approximate large discrete sums: variational methods and sampling. Methods which provide reasonable tight *error bounds* that hold for any structure are either not available or require to solve a series of hard combinatorial optimization problems (Ermon et al., 2013).

Based on work from statistical physics, variational methods (Wainwright & Jordan, 2008) are often very fast but do not provide quality guarantees. Loopy belief propagation (LBP) (Pearl, 1988; Kschischang et al., 2001) is well known for computing locally consistent solutions. Nevertheless, this procedure might not converge and if it does, the result is a local minimum of a surrogate objective (Bethe free energy). Tree-reweighted approaches (Wainwright & Jordan, 2008) deliver the tightest upper bound, but the exact error can not be quantified. In addition, the approximation error depends on the structure of the corresponding probability density. In contrast, the worst-case error of our method is bounded and independent of the structure.

Sampling based techniques are popular, but they suffer from similar issues because the number of samples required to obtain a statistically reliable estimate for multi-dimensional random variables grows exponentially fast. Markov Chain Monte Carlo (MCMC) methods are asymptotically accurate, but guarantees exist only for certain spe-

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

cial cases. Their performance depend crucially on the choice of the proposal distribution, which often must be domain-specific and expert-designed (Girolami & Calderhead, 2011). Importance sampling methods (Liu et al., 2015) try to correct for biased proposal distributions, but their error is hard to quantify as well.

In contrast to existing approaches, our method is based on a numerical approximation technique, which is called quadrature. It allows us to derive a scalable approximation procedure for partition functions, that is applicable to any structure, and delivers guarantees on the approximation error. Surprisingly, quadrature based approximations do not appear in the machine learning literature. The technically most related article is a recent method to compute the log-determinant of large matrices (Han et al., 2015), based on Chebyshev polynomials. Our main contributions can be summarized as follows:

- We present a new deterministic algorithm called *Discrete Clenshaw-Curtis Quadrature* (DCCQ). It gets a structure G, a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$ and a polynomial degree k as inputs, and outputs $\hat{Z}_k(\boldsymbol{\theta})$ such that $|Z(\boldsymbol{\theta}) - \hat{Z}_k(\boldsymbol{\theta})| \leq \frac{\varepsilon}{2} Z(\boldsymbol{\theta})$ in time $\mathcal{O}(k^2 n^{2k})$ whenever $k \in \omega(\|\boldsymbol{\theta}\|_2 - \ln \varepsilon)$.
- Moreover, we show how to split the computation of our approximation into an "expensive" part, that depends on the structure G and the state space $\mathcal{X}$, and a "cheap" part, which depends on the model parameters $\boldsymbol{\theta}$. If the target platform has limited computational resources (like a mobile phone or a sensor), the expensive computation may be carried out on a server. Since the expensive part is independent of the model parameters $\boldsymbol{\theta}$, it can be re-used for any repetitive computation of the partition function—a setting that is typical in iterative parameter estimation procedures. To this end, we present a randomized algorithm, called *Stochastic Discrete Clenshaw-Curtis Quadrature* (SDCCQ). It gets G, $\boldsymbol{\theta}$, k and $\boldsymbol{m}$ as inputs and outputs $\tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta})$ such that $\mathbb{P}[|Z(\boldsymbol{\theta}) - \tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta})| \leq \varepsilon Z(\boldsymbol{\theta})] \geq \zeta$ in time $\mathcal{O}(k^2 n^{2k}) + \mathcal{O}(k^2 \boldsymbol{m}_{\max})$, whenever $k \in \omega(\|\boldsymbol{\theta}\|_2 - \ln \varepsilon)$ and large enough $\boldsymbol{m}_i$.
- Numerical experiments on grid structures show, that the new SDCCQ delivers highly accurate approximations while not suffering from high runtimes, convergence issues or problems with mixed-type potentials, whenever the norm of the parameter vector is small.

As opposed to existing methods, our new approach (i) allows for a user-specified trade-off in terms of runtime and approximation quality by a single parameter, (ii) can outputs a bound of its own worst-case error, (iii) has no convergence issues, (iv) and allows us to split the computational complexity and re-use existing results.

2. Notation and background

In this section, the preliminaries of our new approximation to the partition function are explained. We provide some background on graphical models, followed by a short recap of quadrature rules and polynomial approximation.

2.1. Graphical Models

Consider a multi-variate random variable $\boldsymbol{X}$ where each $\boldsymbol{X}_i$ with $1 \leq i \leq n$ takes values $\boldsymbol{x}_i$ from space $\mathcal{X}_i$. The concatenation of all n variables yields the random variable $\boldsymbol{X}$ with product state space $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$. We denote assignments to single vertices $v \in V$ by $\boldsymbol{x}_v = y$ with $y \in \mathcal{X}_v$. For any subset of variables $U \subseteq V$, $\mathcal{X}_U = \bigotimes_{v \in U} \mathcal{X}_v$ is its joint state space. Moreover, for any $\{v, u, w\} \subset V$, we define $vuw := \{v, u, w\}$ to simplify notation.

The formalism of exponential families provides a unifying framework for a large variety of probability distributions. In particular, they cover all types of probabilistic graphical models (Lauritzen, 1996; Koller & Friedman, 2009) including Bayesian networks (Pearl, 1988), Markov random fields (Wainwright & Jordan, 2008), conditional random fields (Sutton & McCallum, 2011), logistic regression, latent Dirichlet allocation (Blei et al., 2003) and recent deep models (Ranganath et al., 2015). Probability densities which are member of an exponential family can be written as

$$\mathbb{P}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \exp(\langle \boldsymbol{\theta}, \phi(\boldsymbol{x}) \rangle - A(\boldsymbol{\theta})) = \frac{1}{Z(\boldsymbol{\theta})} \psi(\boldsymbol{x}). \quad (1)$$

Beside its *parameter vector* $\boldsymbol{\theta} \in \mathbb{R}^d$, $\mathbb{P}_{\boldsymbol{\theta}}$ consists of two major ingredients, namely the *potential* $\psi(\boldsymbol{x}) = \exp(\langle \boldsymbol{\theta}, \phi(\boldsymbol{x}) \rangle)$ and the *partition function* $Z(\boldsymbol{\theta}) = \exp(A(\boldsymbol{\theta}))$.

Let us first have a closer look at the potential. It assigns a positive real number to every possible instance $\boldsymbol{x}$ by mapping it from $\mathcal{X}$ into a real vector space via a *sufficient statistic* (or feature map) $\phi : \mathcal{X} \to \mathbb{R}^d$, where ϕ encodes the structure of $\boldsymbol{X}$. We ease the notation by identifying components of $\boldsymbol{X}$ with vertices of a graph $G = (V, E)$. In discrete state space models, the state of variables is encoded by indicator functions:

$$\phi_{v=y}(\boldsymbol{x}) = \mathbb{1}_{\{\boldsymbol{x}_v = y\}} \quad (2)$$

$$\phi_{vu=yz}(\boldsymbol{x}) = \mathbb{1}_{\{\boldsymbol{x}_v = y\}} \mathbb{1}_{\{\boldsymbol{x}_u = z\}} \quad (3)$$

$$\cdots$$

$$\phi_{U=\boldsymbol{y}}(\boldsymbol{x}) = \prod_{v \in U} \mathbb{1}_{\{\boldsymbol{x}_v = \boldsymbol{y}_v\}}, U \subseteq V \quad (4)$$

To exactly represent the probability mass of a multi-variate random variable $\boldsymbol{X}$, $\phi(\boldsymbol{x})$ has to contain at least one indica-

tor per possible clique assignments for all maximal cliques[1] of a structure G (Clifford, 1990). If $\mathcal{C}(G)$ is the set of maximal cliques of G, then $\phi(\boldsymbol{x}) = (\phi_{U=\boldsymbol{u}}(\boldsymbol{x}) : \forall U \in \mathcal{C}(G), \forall \boldsymbol{u} \in \mathcal{X}_U)^\top$.

Sufficiency of ϕ is declared with respect to $\boldsymbol{\theta}$, i.e., knowledge about $\boldsymbol{x}$ is not required to infer $\boldsymbol{\theta}$, once $\phi(\boldsymbol{x})$ is known. This property is in particular useful when exponential families have to be applied in resource constrained, autonomous, medical or self quantification devices, because arbitrary large data sets may be aggregated into a finite dimensional representation. In fact, only members of exponential families have this property (Pitman, 1936). We now identify another property of sufficient statistics which will become crucial for finding an efficient inference procedure:

Definition 1. *The sufficient statistics $\phi : \mathcal{X} \to \mathbb{R}^d$ is called χ-integrable if $\chi_\phi : [d]^k \to \mathbb{R}$ with*

$$\chi_\phi(\boldsymbol{j}) = \int_{\mathcal{X}} \left(\prod_{i=1}^{k} \phi(\boldsymbol{x})_{\boldsymbol{j}_i} \right) d\nu(\boldsymbol{x}), \forall k \in \mathbb{N}, \forall \boldsymbol{j} \in [d]^k$$

admits a closed-form expression, that is computable in polynomial time. $[d]^k$ is an abbreviation for $\{1, 2, \ldots, d\}^k$.

In fact, if ϕ is composed out of indicator functions like (2), (3) or (4), it is χ-integrable. We will prove this statement in Section 4. Note, however, that our method is not restricted to binary sufficient statistics or discrete state space models—any χ-integrable ϕ suffices.

The partition function

$$Z(\boldsymbol{\theta}) = \int_{\mathcal{X}} \psi(\boldsymbol{x}) \mathrm{d}\nu(\boldsymbol{x}) \tag{5}$$

accumulates the potential of every possible instance and ensures normalization of $\mathbb{P}_{\boldsymbol{\theta}}$. It is defined w.r.t. a reference measure ν. For continuous (discrete) state spaces, ν is the Lebesgue (counting) measure. If the graph G contains no loops, i.e., is tree-structured, $Z(\boldsymbol{\theta})$ can be evaluated exactly in polynomial time. The same holds for planar Ising models (Schraudolph & Kamenetsky, 2008).

If no special property of the underlying structure can be exploited, evaluating $Z(\boldsymbol{\theta})$ is hard—in fact, **#P**-complete (Valiant, 1979; Bulatov & Grohe, 2004). The junction tree (JT) algorithm (Lauritzen & Spiegelhalter, 1988; Wainwright & Jordan, 2008) can be applied to compute the exact value $Z(\boldsymbol{\theta})$ with runtime exponential in the size of the largest clique of the triangulation of graph G. Nevertheless, several approximate methods arose in the last decades. The Bethe approximation (Bethe, 1935; Wainwright & Jordan, 2008), where the main underlying idea is to treat a general graph like a tree, is maybe the most popular, computed by loopy belief propagation (Pearl, 1988; Frey, 2000; Kschischang et al., 2001). Its runtime depends on the number of message passing iterations until convergence. This number is in general unknown and the algorithm might not even converge. If no further assumptions are made, the Bethe approximation delivers an estimate of unknown quality. It has been shown quite recently, that in case of log-supermodular potential functions (Ruozzi, 2012; Weller & Jebara, 2014), the Bethe approximation is a lower bound on the partition function. Also the naive mean field (MF) technique (Weiss, 2001; Wainwright & Jordan, 2008) is known to provide a lower bound on $Z(\boldsymbol{\theta})$ (Weiss, 2001). The best known general upper bound is based on convex combinations of exponential family parameters $\boldsymbol{\theta}(T)$ which correspond to spanning-trees $T \in \mathcal{T}(G)$ of the original graph G, known as tree-reweighted belief propagation (TRW) (Wainwright et al., 2005). An overview is provided in Table 1.

[1] Cliques are fully connected subgraphs. A clique is maximal if it is not contained in any other clique.

2.2. Quadrature

All of the existing approaches mentioned above are based on an approximation of the structure. In contrast, we propose a numerical approximation of the integration, based on the general quadrature. If integrating a function is not tractable, one has to resort to numerical methods in order to approximate the definite integral. The basic idea of a *quadrature rule* is to replace the integrand f by an approximation $h \approx f$ that admits tractable integration. It turns out that choosing $h = h_k$ to be a degree-k Chebyshev polynomial approximation of f has outstanding properties like rapidly decreasing and individually converging coefficients (Gautschi, 1985). The general quadrature procedure can be summarized as

$$\int_l^u f(x)\mathrm{d}x \approx \int_l^u h_k(x)\mathrm{d}x = \sum_{i=0}^{k} w_i f(x_i) \tag{6}$$

where $x \in \mathbb{R}$, w_i are certain coefficients and x_i are certain abscissae in $[l, u]$ (all to be determined) (Mason & Handscomb, 2002). Depending on the choice of interpolation points and different kinds of orthogonality properties, Chebyshev polynomial based quadrature rules are termed Gauss-Chebyshev quadrature, Fejér quadrature or Clenshaw-Curtis quadrature (Clenshaw & Curtis, 1960). We make use of discrete orthogonality properties and initialize our approximation at the zeros of second kind Chebyshev polynomials, which is a *Clenshaw-Curtis quadrature*.

Chebyshev Polynomials. In order to construct a quadrature rule that is numerically well behaved, *Chebyshev Polynomials* $T_k(x)$ are chosen as a basis. The fundamental recurrence relation is a convenient representation:

$$T_0(x) = 1, T_1(x) = x, T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x). \tag{7}$$

Table 1. State-of-the-art methods to compute/approximate the partition function of an undirected model with graph $G = (V, E)$, $n = |V|$, $m = |E|$. MCMC-based methods are omitted. Here, I is the number of iterations until convergence. $L = \max_{v \in V} |\mathcal{X}_v|$ and $\Delta = \max_{v \in V} |\mathcal{N}_v|$ are the largest vertex domain and neighborhood size, respectively. w is the tree-width of G and d is the dimension of the parameter space. k_ε is the polynomial degree (implied by ε) as specified by Theorem 5. $m_\zeta = \max_i \boldsymbol{m}_i$ is the number of samples (implied by ζ) from the distribution (11) as specified by Theorem 7.

Algorithm	**Complexity**	**Quality**
JT (Lauritzen & Spiegelhalter, 1988)	$\mathcal{O}(L^w)$	Exact
MF (Weiss, 2001)	$\mathcal{O}(InL\Delta)$	Lower bound
LBP (Heskes, 2002; Yedidia et al., 2003)	$\mathcal{O}(ImL^2\Delta)$	Local minimum of Bethe free energy
TRW (Wainwright et al., 2005)	$\mathcal{O}(ImL^2\Delta + m \log n)$	Upper bound
WISH (Ermon et al., 2013)	$\mathcal{O}(n \ln(n/\zeta)) \times \text{Time(MAP)}$	$(16, \zeta)$-approx
DCCQ (Alg. 1)	$\mathcal{O}(k_\varepsilon^2 d^{2k_\varepsilon})$	ε-approx (Theorem 5)
SDCCQ (Alg. 2)	$\mathcal{O}(k_\varepsilon^2 d^{2k_\varepsilon}) + \mathcal{O}(k_\varepsilon^2 m_\zeta)$	(ε, ζ)-approx (Theorem 7)

It can be easily verified that each T_k is a polynomial of degree-k. The degree-k *Chebyshev interpolation* h_k of a continuous function f is $h_k(x) = \sum_{i=0}^k \boldsymbol{c}_i T_i(x)$ with

$$\boldsymbol{c}_i = \frac{2}{k+1} \sum_{j=1}^{k+1} f(x_j) T_i(x_j). \tag{8}$$

The interpolation points $x_j = \cos \frac{j\pi}{k}$ are given by the extrema of the corresponding Chebyshev polynomial which are, at the same time, the zero of the Chebyshev polynomials of the second kind.

The major distinguishing properties of T_k are (i) discrete orthogonality, which is required for the derivation of the coefficients (8) and (ii) the fact that $2^{1-k} T_k$ is the minimax approximation to the 0-function on $[-1, 1]$ (Mason & Handscomb, 2002). Let $h \in \mathcal{H}$ be an approximation to f on the domain $[l, u]$; h is called *minimax approximation* to f iff

$$\forall h' \in \mathcal{H} : \|f - h\|_\infty = \sup_{x \in [l,u]} |f(x) - h(x)| \leq \|f - h'\|_\infty$$

Chebyshev showed that an outstanding property of any minimax approximation is an oscillating error curve.

The coefficients given by (8) result in an approximation to the theoretically optimal (minimax) approximation (Mason & Handscomb, 2002) and allow for a rather fast computation in time $\mathcal{O}(k \log k)$ via discrete cosine transformation (DCT). Coefficients which deliver an approximation that is even closer to the optimal one can be obtained by the Remez exchange algorithm (Fraser, 1965). Error functions for DCT and Remez coefficients are shown in Figure 1 (i). Since both approaches deliver high quality approximations, we use DCT because of its superior efficiency.

Finally, the coefficients $\boldsymbol{c}$, which are coefficients of Chebyshev polynomials, can be converted to *native* coefficients $\tilde{\boldsymbol{c}}$—coefficients of powers of x, say x^i—by summing the corresponding coefficients that appear in the Chebyshev polynomials, weighted by $\boldsymbol{c}_i$. Every Chebyshev interpolation can thus be equivalently expressed as

$$h_k(x) = \sum_{i=0}^{k} \tilde{\boldsymbol{c}}_i x^i. \tag{9}$$

Although the general quadrature is known since long, we, for the first time, exploit it for approximating the partition function.

3. Algorithm

We start with the intuition behind our algorithm for approximating $Z(\boldsymbol{\theta})$ called *Discrete Clenshaw-Curtis Quadrature*. Indeed, we aim at approximating (5) by a quadrature rule (6). In contrast to (6), the integration in (5) has to be carried out over the n-dimensional set $\mathcal{X}$. Note, however, that evaluating the potential at $\boldsymbol{x}$ is equivalent to computing $\exp(r)$ for $r = \langle \boldsymbol{\theta}, \phi(\boldsymbol{x}) \rangle$. Integrating ψ over $\mathcal{X}$ is thus equivalent to integrating $\exp$ over $W = \{r \in \mathbb{R} : r = \langle \boldsymbol{\theta}, \phi(\boldsymbol{x}) \rangle, \boldsymbol{x} \in \mathcal{X}\}$. But without any further assumption on W, the definite integral $\int_W \exp(z) \mathrm{d}z$ continues to have no closed-form expression. We hence take h_k to be the Chebyshev interpolation of $\exp$ on $[l, u]$ with $l = \min W$ and $u = \max W$. Moreover, h_k may be interpreted as approximation $\hat{\psi}_k(\boldsymbol{x}) \approx \psi(\boldsymbol{x})$ over $\mathcal{X}$. Computing $\max W$ is **NP**-hard, since it is equivalent to computing the maximum-a-posteriori assignment of the underlying graphical model.

We may use an upper bound on $\max W$ instead: Remember that $\mathcal{C}(G)$ is the number of (maximal) cliques in G. For binary sufficient statistics (2), (3) and (4), it holds

Algorithm 1 DCCQ

Input: $\boldsymbol{\theta} \in \mathbb{R}^d$, $k \in \mathbb{N}$, ϕ
Output: Approximate partition function $\hat{Z}_k(\boldsymbol{\theta})$
1: $[l, u] \leftarrow \text{interval}(\boldsymbol{\theta})$
2: $\tilde{\boldsymbol{c}} \leftarrow \text{coefficients}(k, [l, u])$ // Eq. (8)
3: $\hat{Z}_k(\boldsymbol{\theta}) \leftarrow \sum_{i=0}^{k} \tilde{\boldsymbol{c}}_i \sum_{\boldsymbol{j} \in [d]^i} \chi_\phi(\boldsymbol{j}) \prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{j}_l}$

that $\forall \boldsymbol{x} : \|\phi(\boldsymbol{x})\|_2 = \sqrt{|\mathcal{C}(G)|}$, since each clique is in exactly one state. Therefore, $\max W \leq \|\boldsymbol{\theta}\|_2 B$ which follows directly from the Cauchy-Schwarz inequality with $B = \sqrt{|\mathcal{C}(G)|}$.

Now, we can approximate ψ on $[l, u]$ by Chebyshev polynomials, and hence:

$$\begin{aligned} Z(\boldsymbol{\theta}) &= \int_{\mathcal{X}} \psi(\boldsymbol{x}) \mathrm{d}\nu(\boldsymbol{x}) \approx \int_{\mathcal{X}} \hat{\psi}_k(\boldsymbol{x}) \mathrm{d}\nu(\boldsymbol{x}) \\ &= \int_{\mathcal{X}} h_k(\langle \boldsymbol{\theta}, \phi(\boldsymbol{x}) \rangle) \mathrm{d}\nu(\boldsymbol{x}) \\ &= \int_{\mathcal{X}} \sum_{i=0}^{k} \tilde{\boldsymbol{c}}_i \langle \boldsymbol{\theta}, \phi(\boldsymbol{x}) \rangle^i \mathrm{d}\nu(\boldsymbol{x}) \\ &= \sum_{i=0}^{k} \tilde{\boldsymbol{c}}_i \sum_{\boldsymbol{j} \in [d]^i} \left(\prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{j}_l} \right) \chi_\phi(\boldsymbol{j}) =: \hat{Z}_k(\boldsymbol{\theta}). \end{aligned} \tag{10}$$

Evaluating the last line has an asymptotic runtime of $\mathcal{O}(k^2 n^{2k}) \times \text{Time}(\chi_\phi)$ which is polynomial in the size of the graph. The procedure is summarized in Algorithm 1. We will investigate in Section 4 which k is required to achieve a reasonable accuracy.

3.1. Complexity-decoupling via Randomization.

$\mathcal{O}(k^2 n^{2k})$ is lower than the time that it takes to enumerate the full state space $\mathcal{X}$, but the partition function has to be recomputed every time the model changes. This happens frequently during iterative parameter learning procedures. Moreover, it is well known that the complexity of inference is mainly influenced by the structure (Lauritzen & Spiegelhalter, 1988; Wainwright & Jordan, 2008). So when this structure does not change, is it necessary to redo this costly computation during learning? It turns out that if we combine our Algorithm 1 with a probabilistic sampling scheme, it is possible to *decouple* the most costly computation from the rest. Surprisingly, the costly part depends only on the structure while the feasible part combines it with the parameters. To see this, notice that the function $\chi_\phi : [d]^k \to \mathbb{R}$ is non-negative, hence

$$\mathbb{P}_\phi(\boldsymbol{j} \mid k) = Q_\phi(k)^{-1} \chi_\phi(\boldsymbol{j}) \tag{11}$$

with $Q_\phi(k) = \sum_{\boldsymbol{j}' \in [d]^k} \chi_\phi(\boldsymbol{j}')$ defines a proper probability mass function (pmf) on $[d]^k$. $\hat{Z}_k(\boldsymbol{\theta})$, as given by (10), may now be rewritten as follows:

$$\begin{aligned} \hat{Z}_k(\boldsymbol{\theta}) &= \sum_{i=0}^{k} \tilde{\boldsymbol{c}}_i \sum_{\boldsymbol{j} \in [d]^k} \chi_\phi(\boldsymbol{j}) \left(\prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{j}_l} \right) \\ &= \sum_{i=0}^{k} \tilde{\boldsymbol{c}}_i Q_\phi(i) \sum_{\boldsymbol{j} \in [d]^k} \mathbb{P}_\phi(\boldsymbol{j} \mid i) \left(\prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{j}_l} \right) \\ &= \sum_{i=0}^{k} w_i \mathbb{E}_{\boldsymbol{J}} \left[\prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{J}_l} \mid i \right] \end{aligned} \tag{12}$$

where $w_i = \tilde{\boldsymbol{c}}_i Q_\phi(i)$ and the expectation is taken with respect to the random variable $\boldsymbol{J}$ with pmf $\mathbb{P}_\phi(\boldsymbol{J} = \boldsymbol{j} \mid i)$ as defined in (11). The runtime has not yet improved compared to (10). However, notice that the expectation $\mathbb{E}_{\boldsymbol{J}} \left[\prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{J}_l} \mid i \right]$ may be approximated via $\hat{\mathbb{E}}_{\boldsymbol{J}} \left[\prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{J}_l} \mid i \right] = \frac{1}{m} \sum_{r=1}^{m} \prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{j}_l^{(r)}}$ by drawing m samples from $\mathbb{P}_\phi(\boldsymbol{J} \mid i)$. Most important, $Q_\phi(i)$ will not change as long as the structure G does not change. This procedure is summarized in Algorithm 2.

Algorithm 2 SDCCQ

Input: $\boldsymbol{\theta} \in \mathbb{R}^d$, $k \in \mathbb{N}$, $\boldsymbol{m} \in \mathbb{N}^k$, ϕ
Output: Approximate partition function $\tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta})$
1: $[l, u] \leftarrow \text{interval}(\boldsymbol{\theta})$
2: $\tilde{\boldsymbol{c}} \leftarrow \text{coefficients}(k, [l, u])$
3: **for** i = 1 to k **do**
4: **if** $Q_\phi(i)$ not cached **then**
5: $Q_\phi(i) \leftarrow \sum_{\boldsymbol{j} \in [d]^i} \chi_\phi(\boldsymbol{j})$
6: cache $Q_\phi(i)$
7: $\tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta}) \leftarrow 0$
8: **for** i = 1 to k **do**
9: sum $\leftarrow 0$
10: **for** r = 1 to $\boldsymbol{m}_i$ **do**
11: sample $\boldsymbol{j} \sim \mathbb{P}_\phi(\boldsymbol{J} \mid i)$
12: sum $\leftarrow$ sum $+ \prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{j}_l}$
13: $\tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta}) \leftarrow \tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta}) + \tilde{\boldsymbol{c}}_i \times Q_\phi(i) \times \frac{1}{\boldsymbol{m}_i} \times$ sum

It is clear from line 4 that the $Q_\phi(i)$ are only computed once. If they are cached, the runtime is $\mathcal{O}(k^2 \max_i \boldsymbol{m}_i)$. In Section 4 it is shown which $\boldsymbol{m}$ suffice to guarantee a small approximation error. However, we can anticipate that $\boldsymbol{m}_i \ll d^k$. We like to stress again that the $Q_\phi(i)$ will not change as long as the graphical structure and the state space $\mathcal{X}$ are fixed, moreover the $Q_\phi(i)$ need not be computed on the same machine. It is reasonable to assume that the $Q_\phi(i)$ are computed on a large cluster while the inference is eventually carried out on a small device with low computational resources or energy constraints.

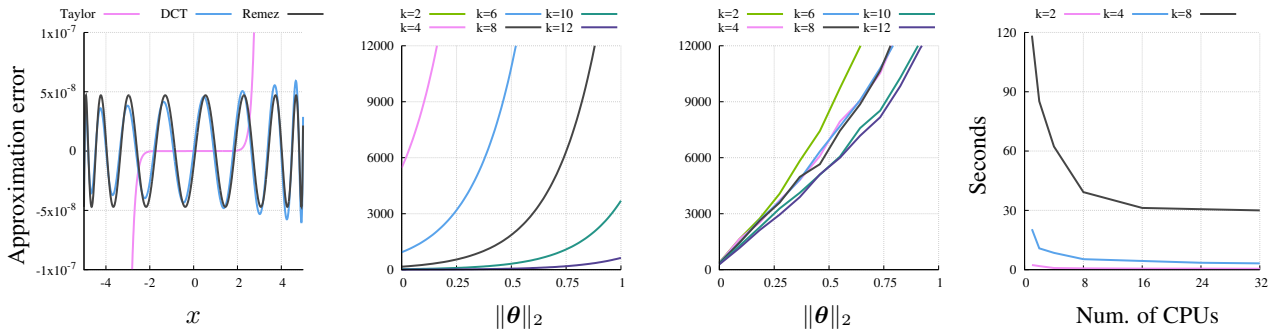

Figure 1. From left to right: (i): Absolute error of degree-16 Taylor, Remez and DCT approximations of $\exp(x)$ on the interval $[-5, 5]$. Only Remez and DCT have oscillating error curves. (ii): Theoretical bounds for the absolute error $|Z(\boldsymbol{\theta}) - \hat{Z}_k(\boldsymbol{\theta})|$ of DCCQ (Alg. 1) as a function of the parameter norm for a 4×4 Ising grid model, predicted by Lemma 3. (iii): Absolute error $|Z(\boldsymbol{\theta}) - \tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta})|$ of SDCCQ (Alg. 2) as a function of the parameter norm for a 4×4 Ising grid model. $\forall i : \boldsymbol{m}_i = 10^4$. (iv): Scalability of SDCCQ (Alg. 2). Runtime in seconds as a function of the number of CPU cores for different polynomial degrees. Best viewed in color.

4. Analysis

The most important prerequisite for our algorithms is χ-integrability of the sufficient statistic.

Lemma 2. *The sufficient statistics of a discrete state space model, constructed from binary indicator functions,* (2)*,* (3) *and* (4)*, is χ-integrable.*

Proof. The state space is discrete and thus the definition of χ-integrability becomes

$$\chi_\phi(\boldsymbol{j}) = \sum_{\boldsymbol{x} \in \mathcal{X}} \left(\prod_{i=1}^{k} \phi(\boldsymbol{x})_{\boldsymbol{j}_i} \right), \forall k \in \mathbb{N}, \forall \boldsymbol{j} \in [d]^k$$

Since $\phi(\boldsymbol{x})_{\boldsymbol{j}_i}$ is binary, the product $\prod_{i=1}^{k} \phi(\boldsymbol{x})_{\boldsymbol{j}_i}$ is binary too. The summation therefore equals the number of instances $\boldsymbol{x} \in \mathcal{X}$ for which the product evaluates to 1 for an arbitrary but fixed $\boldsymbol{j}$. We will derive this number now: As described in Section 2, each index of the vector $\phi(\boldsymbol{x})$ corresponds to a particular assignment of a value $\boldsymbol{y}$ to a clique C. Since $\boldsymbol{j}$ may be any k-dimensional vector with elements from $\{1, 2, \ldots, d\}$, it might happen that it contains *incompatible* indices. Two incompatible indices correspond to two different assignments to the same vertex, which is not possible. Since the summation is over all *possible* instances, the product can never evaluate to 1 for vectors $\boldsymbol{j}$ which contain incompatible indices, and hence $\chi_\phi(\boldsymbol{j}) = 0$. Checking if a vector contains a pair of incompatible indices can be done with $o(k^2)$ comparisons. Any $\boldsymbol{j}$ that contains only compatible indices, corresponds to a possible assignment $\boldsymbol{y}_{U(\boldsymbol{j})}$ to an induced subset $U(\boldsymbol{j}) \subset V$ of vertices. The product will evaluate to 1 for every instance $\boldsymbol{x}$ that matches this particular assignment, i.e., $\boldsymbol{x}_U(\boldsymbol{j}) = \boldsymbol{y}_U(\boldsymbol{j})$. Fixing the values of the vertices in $U(\boldsymbol{j})$ results in a remaining state space of size $|\mathcal{X}|/|\mathcal{X}_{U(\boldsymbol{j})}|$, which is exactly the number of times that the product will evaluate to 1. This finally implies that iff $\boldsymbol{j}$ contains no incompatible indices, then $\chi_\phi(\boldsymbol{j}) = |\mathcal{X}|/|\mathcal{X}_{U(\boldsymbol{j})}|$ and 0 otherwise. This can of course be computed in polynomial time and ϕ is thus χ-integrable. ■

In the rest of this section, we present our main theorems which provide bounds on the error of Algorithms 1 and 2. Formally, the following error bound is known for Chebyshev approximations (Xiang et al., 2010).

Lemma 3. *Suppose f is analytic in the region bounded by the ellipse $\mathcal{E}_\rho = \{z \in \mathbb{C} : |z + \sqrt{z^2 - 1}| = \rho\}$ with major and minor semi-axis lengths summing to $\rho > 1$, foci at ± 1, and $\max_{z \in \mathcal{E}_\rho} |f(z)| \leq M$. Let g_k denote the degree-k interpolant of f according to Eq.* (9)*, then for each $k \geq 0$,*

$$\max_{z \in \{-1,1\}} |f(z) - g_k(z)| \leq \frac{4M}{(\rho - 1)\rho^k}.$$

To apply this Lemma in the sequel, we will consider the function $f = \exp \circ \gamma$ with $\gamma : [l, u] \to [-1, 1]$ and $\gamma(x) = \frac{l+u}{2} + x\frac{u-l}{2}$ (Bernstein, 1912; Mason & Handscomb, 2002). Any choice of $\rho > 1$ would suffice, but increasing ρ increases the imaginary axis of $\mathcal{E}$, and, at the same time, both numerator and denominator of the error bound. On the other hand, for $\rho = 1$, $\mathcal{E}_\rho$ collapses to the real line, which is not compatible with the Lemma. We set $\rho = 1 + \sqrt{2}$ as suggested in (Xiang et al., 2010).

Lemma 4. *Mean field lower bound. Any mean parameter $\boldsymbol{\mu} \in \mathcal{M}^\circ$ with $\boldsymbol{\mu} = \sum_{\boldsymbol{x}} \mathbb{P}(\boldsymbol{x})\phi(\boldsymbol{x})$ yields a lower bound on the partition function: $A(\boldsymbol{\theta}) \geq \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle - A^*(\boldsymbol{\mu})$, where A^* is the convex conjugate of $A = \ln Z$. (Wainwright & Jordan, 2008)*

It can be shown that $-A^*$ is equal to the Shannon entropy $\mathcal{H}(\mathbb{P}) = -\sum_{\boldsymbol{x}} \mathbb{P}(\boldsymbol{x}) \ln \mathbb{P}(\boldsymbol{x})$.

Theorem 5. *Deterministic approximation error. Let ϕ be χ-integrable, $\varepsilon > 0$, $k \geq (\ln 8 + 2 \ln M - \ln(\varepsilon(\rho - 1)))(\ln \rho)^{-1}$ and $\forall \boldsymbol{x} : \|\phi(\boldsymbol{x})\|_2 \leq B$. Then, the*

output $\hat{Z}_k(\boldsymbol{\theta})$ of Algorithm 1 satisfies

$$|Z(\boldsymbol{\theta}) - \hat{Z}_k(\boldsymbol{\theta})| \leq \frac{\varepsilon}{2}|Z(\boldsymbol{\theta})|.$$

Proof. The idea is to show that the lower bound from Lemma 4 is also an upper bound for the right-hand-side in Lemma 3. To see this, construct the naive mean field variational lower bound, based on a fully factored joint probability $\mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) = \prod_{v \in V} \mathbb{P}_v(\boldsymbol{X}_v)$ with uniform vertex marginals $\mathbb{P}_v(\boldsymbol{X}_v) = \mathcal{U}(\mathcal{X}_v)$ (here we assume that the uniform distribution on $\mathcal{X}_v$ exists) for all $v \in V$. This implies for edge marginals that $\mathbb{P}_{vu}(\boldsymbol{X}_v, \boldsymbol{X}_u) = \mathbb{P}_v(\boldsymbol{X}_v)\mathbb{P}_u(\boldsymbol{X}_u)$. Plugging this into Lemma 4, we get

$$\begin{aligned} A(\boldsymbol{\theta}) &\geq \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle + \mathcal{H}(\boldsymbol{\mu}) \\ &\geq -\|\boldsymbol{\theta}\|_2 \|\boldsymbol{\mu}\|_2 + \mathcal{H}(\boldsymbol{\mu}) \\ &\geq -\|\boldsymbol{\theta}\|_2 B - \sum_{v \in V} \sum_{x \in \mathcal{X}_v} \frac{1}{|\mathcal{X}_v|} \ln \frac{1}{|\mathcal{X}_v|} \\ &> -\ln M + \sum_{v \in V} \ln |\mathcal{X}_v| \end{aligned}$$

since $\|\phi(\boldsymbol{x})\|_2 \leq B \Rightarrow \|\boldsymbol{\mu}\|_2 \leq B$. Now, let $M = \max_{z \in \mathcal{E}_\rho} |\exp(\gamma(z))| = \exp(\gamma(\sqrt{2})) = \exp((l+u)/2 + \sqrt{2}(u-l)/2) > \exp((l+u)/2 + (u-l)/2) = \exp(\|\boldsymbol{\theta}\|_2 B) \geq \max_{\boldsymbol{x}} \psi(\boldsymbol{x})$ be an upper bound on the potential, where we set $u = \|\boldsymbol{\theta}\|_2 B$ as suggested in Section 3. By applying Lemma 3, it follows that

$$\begin{aligned} |Z(\boldsymbol{\theta}) - \hat{Z}_k(\boldsymbol{\theta})| &= |\sum_{\boldsymbol{x}} \psi(\boldsymbol{x}) - \sum_{\boldsymbol{x}} \hat{\psi}_k(\boldsymbol{x})| \\ &\leq \sum_{\boldsymbol{x}} |\psi(\boldsymbol{x}) - \hat{\psi}(\boldsymbol{x})| \leq |\mathcal{X}| \frac{4M}{(\rho-1)\rho^k} \\ &\leq |\mathcal{X}| \frac{4\varepsilon}{8M} \leq \frac{\varepsilon}{2} Z(\boldsymbol{\theta}) \end{aligned}$$ ■

Notice that the occurrence of $\|\boldsymbol{\theta}\|_2$ is an artifact of the Cauchy-Schwarz inequality. One may conduct the same derivation based on Hölder's inequality, which in turn would replace the term $\|\boldsymbol{\theta}\|_2 B$ by $\|\boldsymbol{\theta}\|_1 B'$.

Now, we analyze the error that we have to tolerate, if we want to apply complexity-decoupling as explained in Section 3.1. Therein, additional error is introduced by approximating the expected value $\mathbb{E}_{\boldsymbol{J}}[\prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{J}_l} \mid i]$ with the sample mean $\hat{\mathbb{E}}_m[\prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{J}_l} \mid i] := \frac{1}{m} \sum_{r=1}^{m} \prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{j}_l^{(r)}}$, where each $\boldsymbol{j}^{(r)}$ is sampled from the distribution with pmf (11). Indeed, $\hat{\mathbb{E}}_m$ is unbiased and converges to $\mathbb{E}_{\boldsymbol{J}}$. However, we would like to quantify the amount of samples that is required to achieve a desired accuracy.

Theorem 6. *Let $X = X(\boldsymbol{J}) = \prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{J}_l}$, where $\boldsymbol{J}$ is the random variable with pmf (11). Moreover, let $\varepsilon_i > 0$, $\zeta_0 \in (0,1)$ and $m \geq (\ln \zeta_0)/(2\|\boldsymbol{\theta}\|_\infty^i - \varepsilon_i |- \ln M + \ln |\mathcal{X}||)$, where M is an upper bound on the potential, then*

$$\mathbb{P}\left(\left|\mathbb{E}_{\boldsymbol{J}}[X \mid i] - \hat{\mathbb{E}}_m[X \mid i]\right| \geq \varepsilon_i |Z(\boldsymbol{\theta})|\right) \leq \zeta_0.$$

Proof. Applying a Chernoff-like argument, followed by the triangle inequality and Lemma 4, we have

$$\begin{aligned} \mathbb{P}\left(\left|\hat{\mathbb{E}}_m[X \mid i] - \mathbb{E}[X \mid i]\right| \geq \varepsilon_i |Z(\boldsymbol{\theta})|\right) &\leq \\ \mathbb{E}\left[\exp\left(\left|\sum_{l=1}^{m} X_l\right| + m|\mathbb{E}[X]| - m\varepsilon_i |Z(\boldsymbol{\theta})|\right)\right] &\leq \\ \exp\left(m2\|\boldsymbol{\theta}\|_\infty^i - m\varepsilon_i |Z(\boldsymbol{\theta})|\right) &\leq \\ \exp\left(m\left(2\|\boldsymbol{\theta}\|_\infty^i - \varepsilon_i |- \ln M + \ln |\mathcal{X}||\right)\right) &= \zeta_0 \end{aligned}$$ ■

Theorem 7. *Stochastic approximation error. Let the preconditions of Theorem 5 hold. Furthermore, let $\zeta \in (0,1)$. Algorithm 2 with $\boldsymbol{m}_i \geq (\ln \frac{\zeta}{k})/(2\|\boldsymbol{\theta}\|_\infty^i - \frac{\varepsilon}{2k|w_i|}|\ln|\mathcal{X}| - \ln M|)$ delivers an (ε, ζ)-approximation of the partition function. In particular,*

$$\mathbb{P}\left(|Z(\boldsymbol{\theta}) - \tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta})| \geq \varepsilon Z(\boldsymbol{\theta})\right) \leq \zeta.$$

Proof. Let $X = X(\boldsymbol{J}) = [\prod_{l=1}^{i} \boldsymbol{\theta}_{\boldsymbol{J}_l} \mid i]$ and $w_i = \tilde{\boldsymbol{c}}_i Q_\phi(i)$. We conclude, that

$$\begin{aligned} \mathbb{P}\left(|Z(\boldsymbol{\theta}) - \tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta})| \geq \varepsilon Z(\boldsymbol{\theta})\right) &\leq \\ \mathbb{P}\left(\sum_{i=1}^{k} |\mathbb{E}_{\boldsymbol{J}}[X \mid i] - \hat{\mathbb{E}}_{\boldsymbol{m}_i}[X \mid i]| \geq \frac{\varepsilon}{2|w_i|} Z(\boldsymbol{\theta})\right) &\leq \\ \sum_{i=1}^{k} \mathbb{P}\left(|\mathbb{E}_{\boldsymbol{J}}[X \mid i] - \hat{\mathbb{E}}_{\boldsymbol{m}_i}[X \mid i]| \geq \frac{\varepsilon}{2k|w_i|} Z(\boldsymbol{\theta})\right) & \end{aligned}$$

from the triangle inequality and Theorem 5. Now, the claim follows from Theorem 6 with $\varepsilon_i = \frac{\varepsilon}{2k|w_i|}$ and $\zeta_0 = \frac{\zeta}{k}$. ■

Notice that $\tilde{\boldsymbol{c}}$ is required to compute $\boldsymbol{m}$. But this is not restrictive, since $\tilde{\boldsymbol{c}}$ can be computed easily with (8) followed by a conversion to native coefficients. Hence, it is safe to assume that the $\tilde{\boldsymbol{c}}$ are available for computing $\boldsymbol{m}$.

5. Numerical Evaluation

For our experiments, we implemented DCCQ (Alg. 1) and SDCCQ (Alg. 2) and execute them on a machine with 40 E5-2697 Xeon CPU cores. The numerical evaluation should show that SDCCQ achieves highly accurate results whenever the norm of the parameter vector is small. In addition, we investigate cases when the norm is not small and

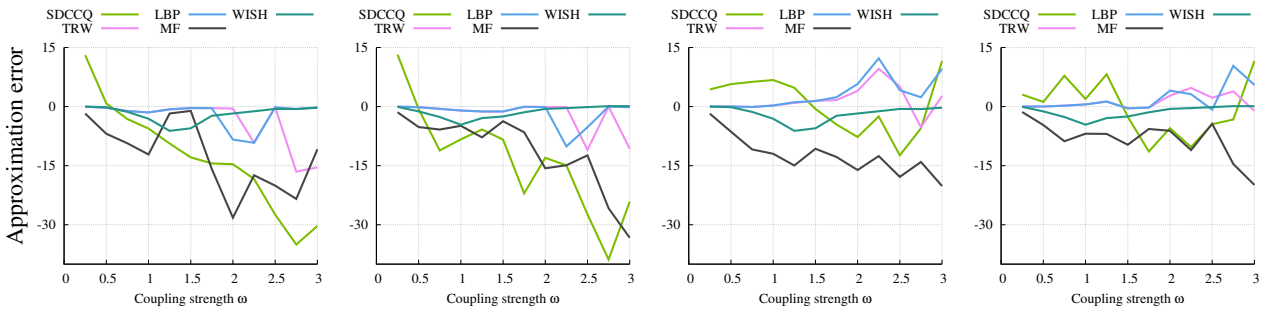

Figure 2. Estimation errors for the log-partition function on 10×10 Ising grids with randomly generated parameter vectors for various coupling strengths. (i): attractive, $\kappa = 0.1$, (ii) attractive, $\kappa = 1.0$, (iii) mixed, $\kappa = 0.1$, (iv) mixed, $\kappa = 1.0$. Best viewed in color.

compare our method to the approaches which are listed in Table 1. Finally, we analyze the scalability of SDCCQ in terms of parallel computing.

One of the main benefits of SDCCQ is the complexity-decoupling. This allows us to compute the $Q_\phi(i)$ (see Alg. 2) only once per structure and reuse them in every run[2]. In total, we computed the $Q_\phi(i)$ for Ising grid models with sizes $2 \times 2, \dots, 128 \times 128$ and $i \in \{1, 2, \dots, 16\}$.

Parameters with small norm. First of all, we show that SDCCQ achieves small errors when the norm of $\boldsymbol{\theta}$ is small (≤ 1). Therefore, we conducted a 4×4 Ising grid with random Gaussian parameters, i.e., $\boldsymbol{\theta}_i \sim \mathcal{N}(0, \sigma)$. We varied σ from 10^{-4} to 1 in order to generate models with various norms. The partition function was then estimated by SDCCQ for polynomial degrees in $\{2, 4, 6, 8, 10, 12\}$ with $\boldsymbol{m}_i = 10^3, \forall i$, while the correct value of the partition function was computed by the JT algorithm. In Figure 1 (iii), the average absolute estimation error, $|Z(\boldsymbol{\theta}) - \tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta})|$ (y-axis), is plotted against the norm of $\boldsymbol{\theta}$ (x-axis). For comparison, the theoretical error curves which are implied by Lemma 3 are depicted in Figure 1 (ii). Empirically, the error of the stochastic low-degree approximations is lower than predicted while the error of the high-degree approximations is higher than predicted. Note, however, that the error bound of SDCCQ (Theorem 7) is stochastic, i.e., an ε-approximation is achieved with a certain probability. Further experiments show that the error can indeed be decreased by increasing the $\boldsymbol{m}_i$. In this setting, an absolute error of 12000 corresponds to a relative error of $\varepsilon < 0.2$.

Parameters with large norm. In the second experiment, we investigate the quality of our new method when the norm of the parameter vector is $\gg 1$ and hence, Theorem 7 predicts a high error for small polynomial degrees. Nevertheless, the question is how SDCCQ compares to other approaches. We use the same experimental setup as in (Ermon et al., 2013). Specifically, we have $n = 10 \times 10$ binary variables $\boldsymbol{x} \in \{-1, 1\}^n$ with weights $\boldsymbol{\theta}_{vu=xy} = -w_{vu}$ whenever $x \neq y$ and $\boldsymbol{\theta}_{vu=xy} = w_{vu}$ otherwise. In the *attractive* setting, the w_{vu} are drawn from $[0, \omega]$; in the *mixed* setting, from $[-\omega, \omega]$. Moreover, vertex weights $\boldsymbol{\theta}_{v=1} = -\boldsymbol{\theta}_{v=-1} = w_v$ are sampled from $[-\kappa, \kappa]$ with $\kappa \in \{0.1, 1.0\}$. This setting implies that $\|\boldsymbol{\theta}\|_2$ is in the range $[3.86, 44.53]$, which is far outside of the interval in which we should expect small errors. Figure 2 shows the estimation error for the log-partition function, i.e. $\ln \tilde{Z}_k^{\boldsymbol{m}}(\boldsymbol{\theta}) - \ln Z(\boldsymbol{\theta})$, averaged over 5-folds of SDCCQ, WISH, MF, TRW and LBP. SDCCQ has been run with $\boldsymbol{m}_i = 10^3, \forall i$ and $k \in \{1, 2, 4, 8\}$, where, due to space restrictions, the plot shows the average over all SDCCQ runs. Individual runs are within a standard deviation of 5. Clearly, the error increases for increasing coupling strengths. However, SDCCQ is often close to the MF lower bound or the TRW upper bound. WISH, delivers the most accurate result but has by far the largest complexity per run.

Scalability. Finally, we analyze the empirical runtime on randomly parametrized Ising grids with 128×128 vertices. Since the major computational cost in Algorithms 1 and 2 arise through a large summation and the sampling step, respectively, both are easy to parallelize. We restricted the execution to a subset of the available CPU cores and measured the runtime in seconds. The result is shown in Fig. 1 (iv), where SDCCQ scales well with an increasing number of CPU cores.

Conclusion. For the first time, we exploited quadrature rules to construct an approximation to the partition function of probabilistic models with arbitrary structure. We derived bounds on the error and compared the new algorithm to other methods. It turned out that the new method scales well on multi-core systems and that it delivers superior results, whenever the norm of the model's parameter vector is small. Moreover, complexity-decoupling allows us to run probabilistic inference with error guarantees on small, resource-constrained devices.

[2]Our C++ source code and the precomputed $Q_\phi(i)$ values are available at http://sfb876.tu-dortmund.de/sdccq.

Acknowledgments

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) within the collaborative research center SFB 876, project A1.

References

Bernstein, Sergei Natanovich. Sur la meilleure approximation des fonctions continues par les polynômes du degré donné. i. *Communications de la Société mathématique de Kharkow. 2-ée série*, 13(2–3):49–144, 1912.

Bethe, Hans A. Statistical theory of superlattices. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 150(871):552–575, 1935.

Blei, David M., Ng, Andrew Y., and Jordan, Michael I. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Bulatov, Andrei and Grohe, Martin. The complexity of partition functions. In Daz, Josep, Karhumki, Juhani, Lepist, Arto, and Sannella, Donald (eds.), *Automata, Languages and Programming*, volume 3142 of *Lecture Notes in Computer Science*, pp. 294–306. Springer, Heidelberg, Germany, 2004.

Clenshaw, C. W. and Curtis, A. R. A method for numerical integration on an automatic computer. *Numerische Mathematik*, 2(1):197–205, 1960.

Clifford, Peter. Markov random fields in statistics. In *Disorder in physical systems*, Oxford Science Publications, pp. 19–32. Oxford University Press, New York, 1990.

Cover, Thomas M. and Thomas, Joy A. *Elements of Information Theory*. John Wiley & Sons, New York, NY, USA, 2nd edition, 2006. ISBN 978-0471241959.

Ermon, Stefano, Gomes, Carla P., Sabharwal, Ashish, and Selman, Bart. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *30th International Conference on Machine Learning*, pp. 334–342, 2013.

Fraser, W. A survey of methods of computing minimax and near-minimax polynomial approximations for functions of a single independent variable. *Journal of the ACM*, 12 (3):295–314, July 1965.

Frey, Brendan J. Local probability propagation for factor analysis. In Solla, S.A., Leen, T.K., and Müller, K. (eds.), *Advances in Neural Information Processing Systems*, volume 12, pp. 442–448. MIT Press, Cambridge, MA, USA, 2000.

Gautschi, W. Questions of numerical condition related to polynomials. *Studies in Numerical Analysis*, (24):140–177, 1985.

Girolami, Mark and Calderhead, Ben. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

Goldberg, Leslie Ann and Jerrum, Mark. A polynomial-time algorithm for estimating the partition function of the ferromagnetic Ising model on a regular matroid. *SIAM Journal on Computing*, 42(3):1132–1157, 2013.

Grünwald, Peter D. *The Minimum Description Length Principle*. The MIT Press, Cambridge, MA, USA, 2007. ISBN 978-0262072816.

Han, Insu, Malioutov, Dmitry, and Shin, Jinwoo. Large-scale log-determinant computation through stochastic Chebyshev expansions. In *32nd International Conference on Machine Learning*, pp. 908–917, 2015.

Heskes, Tom. Stable fixed points of loopy belief propagation are local minima of the Bethe free energy. In Becker, S., Thrun, S., and Obermayer, K. (eds.), *Advances in Neural Information Processing Systems*, volume 15, pp. 343–350, 2002.

Jaakkola, Tommi S. and Jordan, Michael I. Computing upper and lower bounds on likelihoods in intractable networks. In *12th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 340–348. Morgan Kaufmann Publishers, 1996.

Koller, Daphne and Friedman, Nir. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009. ISBN 978-0262013192.

Kschischang, Frank R., Frey, Brendan J., and Loeliger, Hans-Andrea. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2): 498–519, 2001.

Lauritzen, Steffen L. *Graphical Models*. Oxford University Press, Oxford, UK, 1996. ISBN 978-0198522195.

Lauritzen, Steffen L. and Spiegelhalter, David J. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50 (2):157–224, 1988.

Liu, Qiang, Peng, Jian, Ihler, Alexander, and Fisher III, John. Estimating the partition function by discriminance sampling. In *31st Annual Conference on Uncertainty in Artificial Intelligence*, pp. 514–522. AUAI Press, 2015.

Mason, J.C. and Handscomb, David C. *Chebyshev Polynomials*. Chapman and Hall/CRC, 1st edition, 2002. ISBN 978-0849303555.

Pearl, Judea. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Burlington, MA, USA, 1988. ISBN 978-1558604797.

Pitman, Edwin James George. Sufficient statistics and intrinsic accuracy. *Mathematical Cambridge Philosophical Society*, 32:567–579, 1936.

Ranganath, Rajesh, Tang, Linpeng, Charlin, Laurent, and Blei, David M. Deep exponential families. In Lebanon, Guy and Vishwanathan, S. V. N. (eds.), *18th International Conference on Artificial Intelligence and Statistics*, volume 38 of *JMLR W&CP*. JMLR.org, 2015.

Ruozzi, Nicholas. The Bethe partition function of log-supermodular graphical models. In Pereira, F., Burges, C.J.C., Bottou, L., and Weinberger, Kilian Q. (eds.), *Advances in Neural Information Processing Systems*, volume 25, pp. 117–125. Curran Associates, Inc., 2012.

Schraudolph, Nicol N. and Kamenetsky, Dmitry. Efficient exact inference in planar Ising models. In Koller, Daphne, Schuurmans, Dale, Bengio, Yoshua, and Bottou, Léon (eds.), *Advances in Neural Information Processing Systems*, volume 21, pp. 1417–1424, 2008.

Sutton, Charles and McCallum, Andrew. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2011.

Valiant, Leslie G. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3): 410–421, 1979.

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

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

Weiss, Yair. Comparing the mean field method and belief propagation for approximate inference in MRFs. In Opper, M. and Saad, D. (eds.), *Advanced Mean Field Methods:Theory and Practice*, pp. 229–239. MIT Press, Cambridge, MA, USA, 2001.

Weller, Adrian and Jebara, Tony. Clamping variables and approximate inference. In Ghahramani, Zoubin, Welling, Max, Cortes, Corinna, Lawrence, Neil D., and Weinberger, Kilian Q. (eds.), *Advances in Neural Information Processing Systems*, volume 27, pp. 909–917, 2014.

Xiang, Shuhuang, Chen, Xiaojun, and Wang, Haiyong. Error bounds for approximation in Chebyshev points. *Numerische Mathematik*, 116(3):463–491, 2010.

Yedidia, Jonathan S., Freeman, William T., and Weiss, Yair. Exploring artificial intelligence in the new millennium. chapter Understanding Belief Propagation and its Generalizations, pp. 239–269. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.