Approximating the Log-Partition Function

by

Romain Cosson

Diplôme d'ingénieur, Ecole polytechnique

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author . . . . . .
Department of Electrical Engineering and Computer Science
May 20, 2021

Certified by . . . . . .
Devavrat Shah
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Approximating the Log-Partition Function

by

Romain Cosson

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

Graphical Models are used to represent structural information on a high-dimensional joint probability distribution. Their expressiveness offers simple reductions from a large number of NP-hard problems to inference tasks such as computing the partition function (exact inference) or approximating the log-partition function (approximate inference). In this master thesis, we will motivate the need for constant-factor approximations of the log-partition function and prove that a variant of the well studied tree-reweighted algorithm [1] achieves constant factor guarantees. We will express the corresponding approximation ratio $\kappa(G)$ solely as a function of the graph structure G.

Thesis Supervisor: Devavrat Shah
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

This thesis would not have been possible without the support and friendship of many people.

My advisor, Devavrat Shah has guided me throughout the way with amazing intuitions, vision and a lot of patience.

I also wish to thank Anish Agarwal, Dennis Shen and Laurent Massoulié who developped my taste for research and encouraged me with great kindness.

I was fortunate to have very inspiring roommates: Vincent who's lasagna's are an everlasting source of hope, and Moïse who was a devoted linear programming instructor during the winter. Adrien, Antoine, and all my friends who will stop reading after page 4.

Of course, all of this originates and will continue with an amazing family, Papa, Maman, Pauline, Maxime, Marion, Victor et bientôt Maxence.

Contents

Chapter 1

Introduction

Summary of background Graphical models have frequently been applied by the literature [2, 3] to solve combinatorial problems with inference-inspired algorithms. For instance [2] proposes to use a message passing algorithm to compute the maximum weight matching of a graph. This poses the general question of the hardness of inference in graphical models.

It was shown by [4] that the complexity of *exact inference* in graphical models is super-polynomial with respect to k, where k is the tree-width of the underlying graph G. The notion of tree-width is a graph invariant that was introduced by [5] and which is equal to 1 if G has no cycle and to n if G is the complete graph on n vertices. Conversely, a problem that can be reduced to computing the partition function of a graph with bounded tree-width can be solved in polynomial time. For instance, the chromatic number of a graph with bounded tree-width can be computed efficiently, as well as a maximum independent set [6].

Regarding *approximate inference*, no guarantee that would only depend on the graph structure has been provided by the literature to assess feasibility - though variational methods have been widely and successfully deployed for graphical models [7]. Variational methods, such as Mean Field and Bethe typically relax a characterization of the log-partition function as an infimum over the space of distributions. On another line, the tree-reweighted approach developed in [1] uses the convexity of the log-partition function with respect to the weight of potentials to provide an upper bound of the log-partition function that only requires to do inference on spanning trees of G. In Chapter 2, we will show how this upper-bound can be transformed in a constant factor approximation of the log-partition function where the approximation factor $\kappa(G)$ only depends on the graph structure.

Organization of the thesis Chapter 1 will introduce notations, definitions and motivate the problem assuming no prior knowledge of graphical models, Chapter 2 will consist in an analysis of the tree-reweighted variant yielding a constant factor approximation of the log-partiton function, Chapter 3 will briefly describe open directions for future research.

1.1 Graphical models and the partiton function

1.1.1 Undirected graphical models

Undirected graphical models, also known as Markov random fields (MRF) allow to represent the conditional independence relations satisfied by a probability distribution on a (high dimensional) space $\mathcal{X}^n$, where $\mathcal{X}$ is the *alphabet* (assumed finite) and n is the *dimension* or number of random variables. More precisely, when a random vector $\mathbf{x}$ has coordinates $(\mathsf{x}_i)_{i\in[n]}$ satisfy certain conditional independence relations of the form $\mathsf{x}_i \perp\!\!\!\perp \mathsf{x}_j \mid (\mathsf{x}_u)_{u\notin\{i,j\}}$, it is natural to define the set of *edges* $E \subset [n] \times [n]$ such that:

$$\forall (i,j) \in E^c : \mathsf{x}_i \perp\!\!\!\perp \mathsf{x}_j \mid (\mathsf{x}_u)_{u\notin\{i,j\}}. \tag{1.1}$$

where $E^c = ([n] \times [n]) \setminus E$ denotes the complementary set of E. This defines a graph $G = ([n], E)$ that in turn carries information about the induced conditional independence relations and factorization by the Hammersley Clifford theorem. Canonical example of distributions that can be represented by a graphical models are discrete time Markov Chains, $\mathbf{x} = (\mathsf{x}_t)_{t\in[n]}$, that correspond to a line graph with n nodes (aligned in chronological order). For more precision and alternative definitions of graphical models, refer to [8] (lectures notes of 6.438).

1.1.2 Pairwise graphical models

Pairwise graphical models (or pairwise Markov random fields) are a particular case of graphical models in that their distribution factorizes as a product on the edge set $E \subset [n] \times [n]$. They represent distributions $p \in \mathcal{P}(\mathcal{X}^n)$ of $(\mathsf{x}_i)_{i\in[n]}$ that factorize as follows:

$$\forall \boldsymbol{x} \in \mathcal{X}^n : p_{\mathsf{x}}(\boldsymbol{x}) \propto \prod_{i\in[n]} \psi_i(x_i) \prod_{e\in E} \psi_e(x_e), \tag{1.2}$$

where $\psi_i : \mathcal{X} \to \mathbb{R}^+$ and $\psi_e : \mathcal{X}^2 \to \mathbb{R}^+$ are functions called *node potentials* and *edge potentials* and where for $e = (i,j)$, x_e denotes the pair (x_i, x_j). Note that this distribution satisfies the conditional independence relations induced by the graphical model $G = (V, E)$. Indeed, if $i, j \in E^c$, $p_{\mathsf{x}}(\boldsymbol{x})$ can be decomposed in a product of two terms, the first of which will depend on x_i (and not on x_j) and the latter on x_j (and not on x_i), which gives the desired conditional independence property. Also note that the existence of node potentials is superficial, as they no not add expressivity if every node of G is connected to at least one edge. Therefore we will often write as follows:

$$\forall \boldsymbol{x} \in \mathcal{X}^n : p_{\mathsf{x}}(\boldsymbol{x}) \propto \prod_{e\in E} \psi_e(x_e). \tag{1.3}$$

Although pairwise graphical models are simple to express with their edge potentials, their power of expression remains important and comparable with general graphical models (see 3.3 in [7]).

1.1.3 The partition function Z

We have defined distributions by their potentials: that is up to a multiplicative constant. We call this multiplicative constant the *partition function* and denote it Z. Its expression is naturally as follows:

$$Z = \sum_{x \in \mathcal{X}^n} \left(\prod_{e \in E} \psi_e(x_e) \right) \tag{1.4}$$

This expression can be hard to compute because it consists of a sum of $|\mathcal{X}|^n$ terms, therefore a naive computation will always take exponential time. It is actually not possible to compute Z efficiently in general. Indeed assume $\mathcal{X} = \{0, 1, 2\}$ and $\forall e : \psi_e(x_i, x_j) = \mathbb{1}(x_i \neq x_j)$; it appears that $Z > 0$ if and only if $G = (V, E)$ admits a three-coloring, and this decision problem is famously NP-hard [9]. More precisely, Z counts the number of such colorings and its computation therefore belongs to the (harder) class of complexity $\#P$ [10].

Computing sums of products (partition functions) appear in many inference tasks. For instance, if we wish to express the marginal distribution p_{x_i} of some variable x_i in a graphical model, we would write as follows:

$$\forall z \in \mathcal{X} : p_{\mathsf{x}_i}(z) = \frac{1}{Z} \sum_{\boldsymbol{x} \in \mathcal{X}^{n-1}} \left(\prod_{e \in E^{\setminus i}} \psi_e(x_e) \prod_{u \in \mathcal{N}(i)} \psi_{(u,i)}(x_u, z) \right) \tag{1.5}$$

where $E^{\setminus i}$ denote the set of edges that do not involve i and $\mathcal{N}(i) \subset V$ denotes the neighbours of i. Note that the denominator Z is the partition function on G and that the numerator also takes the form of a partition function on the graph $G \setminus \{i\}$ where i has been removed from G and where the potentials of variables of $\mathsf{x}_u : u \in \mathcal{N}(i)$ have been slightly updated. Therefore, finding a efficient scheme for computing partition functions allows to perform marginalization and other inference tasks (for more details, see [8]). This is why the tractability of the partition function is often considered as the baseline for assessing the feasibility of inference in a Markov random field [4].

1.2 Hardness of exact inference

1.2.1 Tractable partition functions

For some families of potentials like *Gaussian families* (where $\mathbf{x}$ is a Gaussian vector), the partition function is always tractable. As we saw above, this guarantees the efficiency of other inference tasks like computing the distribution of a marginal x_i or finding the most likely configuration $\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x}\in\mathcal{X}^n} p_{\mathsf{x}}(\boldsymbol{x})$ (see Gaussian inference in [8]).

Another option to guarantee the tractability of the partition function is to make strong assumption on the graph structure, typically asking for very sparse structures. If a graphical model is defined on a tree, its partition function can be computed efficiently in $O(n|\mathcal{X}|^2)$ with the *sum-product algorithm* and this whatever the choice of potentials. To describe this algorithm, we assume that $G = (V, E)$ is a tree and select a root $r \in V$. Each non-root node i has a unique parent $p(i)$, and possibly several children in set $\mathcal{C}(i)$. Notice that because G is assumed to be a tree, $E = \{(i, p(i)) \mid i \in V \setminus \{r\}\}$ and $|E| = n - 1$. The sum-product algorithm defines $|E|$ *messages* denoted $m_{i\to p(i)}$ that are functions of $\mathcal{X}$ ($m_{i\to p(i)} : \mathcal{X} \to \mathbb{R}^+$) and that are defined by recursion from the leaves to the root:

$$\forall i \neq r : m_{i\to p(i)}(z) = \sum_{y\in\mathcal{X}} \left(\psi_{(i,p(i))}(y, z) \prod_{u\in\mathcal{C}(i)} m_{u\to i}(y) \right) \tag{1.6}$$

and we have:

$$Z = \prod_{u\in\mathcal{C}(r)} m_{u\to r}(x_r). \tag{1.7}$$

Note that the runtime of this algorithm is in $O(n|\mathcal{X}|^2)$ because each of message takes $|\mathcal{X}|^2$ time to compute and there are $|E| = n - 1$ messages. The validity of this algorithm and (1.7) can be shown by induction.

1.2.2 Tree-width and complexity of exact inference

The sum-product algorithm is generalizable to *partial-k-trees* in what is known as the *elimination algorithm* that we will now describe. In the algorithm above, we have recursively eliminated the leaves of the tree until the messages reached the root and their product would be equal to the partition function. Observe that the difficulty of a generalization lies in the fact that, when a node has several neighbours, their elimination will result in affecting the potentials between these neighbours. Therefore, one should attempt to minimize the number of neighbour each node has when it is being eliminated (while this number could be kept equal to 1 for trees).

More precisely, the *elimination of a node* $i \in V$ from a graph $G = (V, E)$ will result in a graph $G' = (V', E')$ where $V' = V \setminus \{i\}$ and $E' = E^{\setminus\{i\}} \cup \{(u, v) \mid u, v \in \mathcal{N}(i)\}$. Note that eliminating a node of degree d may result in adding up to $d(d-1)/2$ edges because its neighbours must form a clique in E'. We define an *elimination order* $\sigma \in S_n$ which is a permutation on the nodes, and consider the elimination sequence of graphs $G_0, G_1, ..., G_n$ where $G_0 = G$ and G_{i+1} is the graph where $\sigma(i)$ has been *eliminated* from G_i. A *partial-k-tree* is a graph such that there exists an elimination order that never eliminates a node of degree more than k. As an example, regular trees are partial-1-trees, and the corresponding elimination orders starts by eliminating the leaves. The *tree-width* k of a graph is the smallest integer such that this graph is a partial-k-tree.

Once an elimination order has been identified for a graph of treewidth k, a direct generalisation of the sum-product algorithm will allow a $O(n|\mathcal{X}|^k)$ computation of the partition function. Other equivalent views exists on this algorithm, one of which consists in constructing a *junction tree*, that is a tree-based graphical model on alphabet $|\mathcal{X}|^k$ and that has the same partition function Z (see [7] for more details).

The idea that the tree-width k is a structural criteria that assesses the general complexity of exact inference on a graph G was consolidated by [4] who proved that the computational complexity of inference is at least super-polynomial in the treewidth if $P \neq NP$. This result relies on a combinatorial hypothesis due to [11] that allows to encode a MAX-2-SAT problem on g variables (NP hard in g) in *any* graph of treewidth greater than poly(g). This hypothesis was recently proven in [12].

1.3 Hardness of approximate inference

Since exact inference is not feasible for a lot of canonical graphical models (see [7] on general Boltzmann machines). Another option is to turn to approximate inference where the objective is to approximate quantities, like the partition function Z. This will require additional assumptions; in what follows, edge potentials are assumed to be positive. Consequently, one can write a pairwise graphical model in its *exponential form*:

$$\forall \boldsymbol{x} \in \mathcal{X}^n : p_{\mathsf{x}}(\boldsymbol{x}) \propto \prod_{e \in E} \psi_e(x_e) = \exp\left(\sum_{e \in E} \phi_e(x_e)\right). \tag{1.8}$$

where $\phi_e = \log(\psi_e)$ denote the *log-potentials*. This positivity assumption will be useful for approximate methods that we will now present. To understand why there is little hope to achieve significant approximations of the log-partition function without this assumption, observe again that if we choose $\mathcal{X} = \{1, 2, 3\}$ and $\psi_{e=(u,v)}(x_e) = \mathbb{1}(x_u \neq x_v)$ (which is not always positive) then $Z > 0$ if and only if G is 3-colorable. Since 3-colorability

is an NP-hard problem, any multiplicative approximation guarantee on Z or $\log(Z)$ is not achievable in polynomial time.

In what follows, we will make the stronger assumption that log-potentials ϕ_e are positive or equivalently that potentials ψ_e satisfy $\forall \boldsymbol{x} \in \mathcal{X}^n, \forall e \in E : \psi_e(x_e) \geq 1$. Under this assumption, we will attempt to approximate $\log(Z) \geq 0$ up to a constant factor $\alpha \geq 1$, i.e. to produce $\widehat{\log(Z)}$ such that:

$$\frac{1}{\alpha} \log(Z) \leq \widehat{\log(Z)} \leq \alpha \log(Z). \tag{1.9}$$

1.3.1 Relation to constraint satisfaction problems

Although we restricted our task only to achieving a constant factor approximation of $\log(Z)$ (therefore an exponential approximation of Z), this is likely to remain hard for most general graphs for any $\alpha > 1$. To explain why this is the case, we recall the Unique Games Conjecture from [13].

Definition 1.3.1 (Unique Game, [13])**.** *A unique game $\mathcal{U} = (G, \mathcal{X}, \{\pi_e \mid e \in E\})$ is a constraint satisfaction problem defined as follows: $G = (V, E)$ is a graph whose vertices represent variables and edges represent constraints. The goal is to assign to each vertex $i \in V$ a label $x_i \in \mathcal{X}$ where $\mathcal{X}$ is a finite set of labels. An edge $e = (i, j)$ is satisfied by the labelling if $(x_i, x_j) \in \pi_e$ where π_e denotes a permutation/matching on the labels. Let OPT($\mathcal{U}$) denote the maximum fraction of constraints that can be satisfied by any labeling:*

$$\textit{OPT}(\mathcal{U}) = \max_{\boldsymbol{x} \in \mathcal{X}^n} \frac{1}{|E|} \left| \{e = (i, j) \in E \mid (x_i, x_j) \in \pi_e\} \right| \tag{1.10}$$

Note that the definition from [13] uses directed graph but can easily be reduced to the definition above by dedoubling all the vertices. Also note that MAXCUT is a Unique Game with $\mathcal{X} = \{0, 1\}$ and $\forall e \in E : \pi_e = \{(0, 1), (1, 0)\}$.

Definition 1.3.2 (Unique Games Conjecture, [13])**.** *For any $\delta > 0$, there is a constant $c \in \mathbb{N}$ that only depends on δ, such that given a unique game $\mathcal{U} = (G, [c], \{\pi_e \mid e \in E\})$ it is NP-hard to distinguish from these two cases:*

- *YES Case: OPT($\mathcal{U}$) $\geq 1 - \delta$*
- *NO Case: OPT($\mathcal{U}$) $\leq \delta$*

Observe that any algorithm producing a $(1 - \delta)/\delta$-approximation of OPT($\mathcal{U}$), would immediately solve the decision problem above. This yields the following observation:

Theorem 1.3.1. *If the Unique Games Conjecture holds, the task of approximating the log-partition function of a pairwise Markov random field with non-negative log-potentials up to any constant factor $\alpha > 1$ cannot be solved by a generic algorithm in polynomial time.*

To confirm this statement, observe that there is a simple reduction from maximum constraint satisfaction to approximations of the log-partition function. Given a unique game $\mathcal{U} = (G, [c], \{\pi_e \mid e \in E\})$, consider the graphical model on G with alphabet $\mathcal{X} = [c]$ and potentials $\forall e = (u, v) : \phi_e(x_e) = \exp(\beta \mathbb{1}((x_u, x_v) \in \pi_e)) > 1$ for some $\beta > 0$. The partition function then writes as follows:

$$Z(\beta) = \sum_{\boldsymbol{x} \in [c]^n} \left(\prod_{e=(u,v) \in E} \exp(\beta \mathbb{1}((x_u, x_v) \in \pi_e)) \right), \tag{1.11}$$

$$= \sum_{\boldsymbol{x} \in [c]^n} \exp(\beta q(\mathcal{U}, \boldsymbol{x})), \tag{1.12}$$

$$= \sum_{q \in [|E|]} N(\mathcal{U}, q) \exp(\beta q), \tag{1.13}$$

where $q(\mathcal{U}, \boldsymbol{x}) \in \mathbb{N}$ denotes the number of constraints satisfied by $\boldsymbol{x}$, an instance of $\mathcal{U}$, and $N(\mathcal{U}, q)$ denotes the number of instances $\boldsymbol{x} \in [c]^n$ satisfying exactly q constraints of $\mathcal{U}$. Notice that as $\beta \to \infty$, $Z(\beta)$ behaves as its dominant term $\exp(\beta q_{\mathcal{U}})$ where $q_{\mathcal{U}} = |E|\text{OPT}(\mathcal{U})$. More precisely, for all β :

$$\exp(\beta q_{\mathcal{U}}) \leq Z(\beta) \leq c^n \exp(\beta q_{\mathcal{U}}), \tag{1.14}$$

$$\beta q_{\mathcal{U}} \leq \log(Z(\beta)) \leq n \log(c) + \beta q_{\mathcal{U}}. \tag{1.15}$$

By choosing, $\beta_\epsilon = \frac{1}{\epsilon} n \log(c)$, we get that $\frac{1}{\beta_\epsilon |E|} \log(Z(\beta_\epsilon))$ is an ϵ-approximation for $\text{OPT}(\mathcal{U})$.

$$(1 - \epsilon) \frac{1}{\beta_\epsilon |E|} \log(Z(\beta_\epsilon)) \leq \text{OPT}(\mathcal{U}) \leq \frac{1}{\beta_\epsilon |E|} \log(Z(\beta_\epsilon)). \tag{1.16}$$

Therefore any generic polynomial time algorithm yielding a constant factor approximation the log-partition function $\log(Z(\beta_\epsilon))$ would result in contradicting the unique games conjecture for some $\delta > 0$.

1.3.2 Approximate inference by variational methods

Variational methods typically translate the inference task (such as computing Z), to an optimization task that is then approximately solved. As an example we recall the variational characterization of $\log(Z)$ from [8, 14]:

$$\log(Z) = \sup_{q \in \mathcal{P}(\mathcal{X}^N)} \left[H(q) - \mathbb{E}_{\mathbf{x} \sim q} \left(\sum_{e \in E} \psi_e(\mathsf{x}_e) \right) \right] \tag{1.17}$$

where $H(q)$ is the entropy of the distribution q (maximal for the uniform distribution) and $\mathbb{E}_{\mathbf{x} \sim q} \left(\sum_{e \in E} \psi_e(\mathsf{x}_e) \right)$ represents the expected log-potential if x is sampled following

distribution q. The supremum of this expression is attained for p, the latent probability distribution defined as in (1.3) which is optimal for this entropy / energy tradeoff.

This expression can easily be proven by observing that $\inf_{q\in\mathcal{P}(\mathcal{X}^N)} KL(q \mid\mid p) = 0$, where equality is attained if and only if $q = p$. The following derivations are all equal to zero and prove the formula above.

$$\inf_{q\in\mathcal{P}(\mathcal{X}^N)} KL(q \mid\mid p) \tag{1.18}$$

$$\inf_{q\in\mathcal{P}(\mathcal{X}^N)} \int_{\mathcal{X}^N} q(x) \log\left(\frac{q(x)}{p(x)}\right) \quad \text{where} \quad p(x) = \frac{1}{Z}\exp\left(\sum_{e\in E}\psi_e(x_e)\right) \tag{1.19}$$

$$\inf_{q\in\mathcal{P}(\mathcal{X}^N)} \left[-H(q) + \mathbb{E}_{\mathbf{x}\sim q}\left(\sum_{e\in E}\psi_e(\mathsf{x}_e)\right) + \log(Z)\right] \tag{1.20}$$

where $H(q) = \mathbb{E}_{\mathbf{x}\sim q}\left(\log\left(\frac{1}{q(\mathbf{x})}\right)\right)$. This variational characterization of Z opens the way to approximate solutions such as that given by mean field (restricting the optimization to the space of distributions with independent coordinates).

The technique described above typically provide lower bounds on the partition function by restricting the space of optimization. Another variational method was deployed in [1] to obtain upper bounds on the log-partition function. It consists of parametrizing the partition function with edge weights $\boldsymbol{\theta} = (\theta_e)_{e\in E} \in \mathbb{R}^E$ as follows:

$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}\in\mathcal{X}^n} \exp\left(\sum_{e\in E}\theta_e\psi_e(x_e)\right). \tag{1.21}$$

Note that the partition function Z that we considered from the begining corresponds to the edge weights $\boldsymbol{\theta} = \mathbf{1} = (1, ..., 1)$.

$$Z = Z(\mathbf{1}) \tag{1.22}$$

The observation that $\Phi : \boldsymbol{\theta} \to \log(Z(\boldsymbol{\theta}))$ is convex then allows to write $\log(Z)$ as an infimum over weighted combinations of partition functions.

$$\log(Z) = \inf_{\substack{(\rho^t):\sum_t \rho^t = 1 \\ (\boldsymbol{\theta}^t):\sum_t \rho^t\boldsymbol{\theta}^t = \mathbf{1}}} \sum_t \rho^t \log(Z(\boldsymbol{\theta}^t)) \tag{1.23}$$

Though this equation may appear trivial because equality is attained for $\rho^1 = 1$ and $\boldsymbol{\theta}^1 = \boldsymbol{\theta}$, its interest is that the right hand side that is combination of $\log(Z(\boldsymbol{\theta}^t))$ may happen to be tractable if $\boldsymbol{\theta}^t \in \mathbb{R}^E$ are very sparse. More formally, if the support of $\boldsymbol{\theta}^t \in \mathbb{R}^E$, defined as $H^t = \{e \in E \mid \boldsymbol{\theta}^t_e \neq 0\} \subset E$ forms a tree, then $\log(Z(\boldsymbol{\theta}^t))$ is tractable with the sum-product algorithm, yielding a tractable upper bound on the partition function.

We conclude this section by proving that the function $\boldsymbol{\theta} \to \log(Z(\boldsymbol{\theta}))$ is convex in $\boldsymbol{\theta}$ because its Hessian $H(\boldsymbol{\theta}) = \left(\frac{\partial \log(Z(\boldsymbol{\theta}))}{\partial \theta_{e_1} \partial \theta_{e_2}}\right)_{e_1, e_2 \in E}$ is positive semidefinite. Let $e_1, e_2 \in E$ be two edges. We can write as follows:

$$\frac{\partial \log(Z(\boldsymbol{\theta}))}{\partial \theta_{e_1}} = \frac{\sum_{\boldsymbol{x} \in \mathcal{X}^n} \psi_{e_1}(x_{e_1}) \exp\left(\sum_{e \in E} \theta_e \psi_e(x_e)\right)}{Z(\boldsymbol{\theta})} = \sum_{\boldsymbol{x} \in \mathcal{X}^n} p_{\boldsymbol{\theta}}(\boldsymbol{x}) \psi_{e_1}(x_{e_1}), \quad (1.24)$$

$$= \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\theta}}}(\psi_{e_1}(\mathsf{x}_{e_1})). \quad (1.25)$$

where $p_{\boldsymbol{\theta}}$ denotes the distribution associated to the re-weighted potentials $(\theta_e \psi_e)_{e \in E}$ and the partition function $Z(\boldsymbol{\theta})$. This gives,

$$\frac{\partial \log(Z(\boldsymbol{\theta}))}{\partial \theta_{e_1} \partial \theta_{e_2}} = \frac{\sum_{\boldsymbol{x} \in \mathcal{X}^n} \psi_{e_1}(x_{e_1}) \psi_{e_2}(x_{e_2}) \exp\left(\sum_{e \in E} \theta_e \psi_e(x_e)\right)}{Z(\boldsymbol{\theta})} \quad (1.26)$$

$$- \frac{\left(\sum_{\boldsymbol{x} \in \mathcal{X}^n} \psi_{e_1}(x_{e_1}) \exp\left(\sum_{e \in E} \theta_e \psi_e(x_e)\right)\right) \left(\sum_{\boldsymbol{x} \in \mathcal{X}^n} \psi_{e_2}(x_{e_2}) \exp\left(\sum_{e \in E} \theta_e \psi_e(x_e)\right)\right)}{Z(\boldsymbol{\theta})^2} \quad (1.27)$$

$$= \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\theta}}}(\psi_{e_1}(\mathsf{x}_{e_1}) \psi_{e_2}(\mathsf{x}_{e_2})) - \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\theta}}}(\psi_{e_1}(\mathsf{x}_{e_1})) \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\theta}}}(\psi_{e_2}(\mathsf{x}_{e_2})) \quad (1.28)$$

$$= \mathrm{Cov}_{\mathbf{x} \sim p_{\boldsymbol{\theta}}}(\psi_{e_1}(x_{e_1}), \psi_{e_2}(x_{e_2})). \quad (1.29)$$

Therefore we can conclude that $H(\boldsymbol{\theta}) = \mathrm{Cov}_{\mathbf{x} \sim p_{\boldsymbol{\theta}}}(\psi(\mathbf{x}))$ is the covariance matrix of the random vector $(\psi_e(\mathsf{x}_e))_{e \in E}$ and is therefore positive semidefinite.

1.3.3 Bounding a sum of products

In the previous section, we described typical variational methods used to obtain lower and upper bounds on the partition function. Before using those to obtain an approximation of the log-partition function, we notice here that our derivations are not specific to the notion of partition function and would actually hold for any sum of products in a general form. Let $\mathcal{W}$ be a class of weights on E all greater or equal than one, $\forall \boldsymbol{w} \in \mathcal{W}, \forall e \in E : w_e \geq 1$. We wish to estimate the sum of the product of the weights $\sum_{\boldsymbol{w} \in \mathcal{W}} \prod_{e \in E} w_e$. For any distribution on subsets of E, i.e. $\boldsymbol{\rho} = (\rho_H)_{H \subset E}$ s.t. $\rho_H \geq 0$ and $\sum_{H \subset E} \rho_H = 1$, we have that:

$$\sum_{H \subset E} \rho_H \log\left(\sum_{\boldsymbol{w} \in \mathcal{W}} \prod_{e \in H} w_e\right) \leq \log\left(\sum_{\boldsymbol{w} \in \mathcal{W}} \prod_{e \in E} w_e\right) \leq \sum_{H \subset E} \rho_H \log\left(\sum_{\boldsymbol{w} \in \mathcal{W}} \prod_{e \in H} w_e^{\frac{1}{\rho_e}}\right) \quad (1.30)$$

where $\rho_e = \sum_{H \ni e} \rho_H$ denotes the probability that $e \in E$ appears in a $H \subset E$ sampled from ρ,

$$\rho_e = \mathbb{P}_{H \sim \rho}(e \in H). \quad (1.31)$$

This gives an upper and lower bound on the quantity on interest, that only depends on the distribution $\boldsymbol{\rho}$. Note that if the distribution is concentrated on E (i.e. $\rho_E = 1$) the upper and lower bounds are tight whereas if the distribution is uniform on the singletons (i.e. $\forall e : \rho_{\{e\}} = \frac{1}{|E|}$), then the upper and lower bounds can differ from a $\frac{1}{|E|}$ multiplicative factor. Importantly, we can prove that the upper an lower bound always differ by at most by the following multiplicative constant:

$$\kappa_{\boldsymbol{\rho}} = \min_{e \in E} \rho_e \tag{1.32}$$

We will now give some brief explanations on these bounds that relate them to the previous section. The lower bound is straightforward: all the terms of the weighted sum are below the quantity of interest. The upper bound can be obtained by convexity just like in the previous section (observe that $(\rho_e)_{e \in E} = \sum_{H \subset G} \mathbb{1}(e \in H)\rho_H$). Finally, the fact that the upper an lower bound only differ by a multiplicative constant $\kappa(\rho)$ comes from the following inequality: for any $\mathbf{s} = (s_i) \in \mathbb{R}_+^n$ and $\lambda \geq 1$,

$$\sum_{i=1}^{n} s_i^{\lambda} \leq \big(\sum_{i=1}^{n} s_i\big)^{\lambda}.$$

1.3.4 Balanced covering of a graph with its trees

The interest in the bound presented above is that it directly applies to bounding the log-partition function.

$$\sum_{H \subset G} \rho_H \log \left(\sum_{x \in \mathcal{X}^n} \prod_{e \in H} \psi_e(x_e) \right) \leq \log \left(\sum_{x \in \mathcal{X}^n} \prod_{e \in E} \psi_e(x_e) \right) \leq \sum_{H \subset G} \rho_H \log \left(\sum_{x \in \mathcal{X}^n} \prod_{e \in H} \psi_e(x_e)^{\frac{1}{\rho_e}} \right) \tag{1.33}$$

Note that the upper and lower bounds can themselves be seen as averages of log-partition functions on edge-induced subgraphs $H \subset E$. If $\boldsymbol{\rho}$ has support on graph structures for which inference is tractable - like trees or any graph with bounded tree-width - these bounds turn out to be tractable. Recall from the previous section that the quality of the approximation depends on how well $\boldsymbol{\rho}$ spans all edges through the quantity $\kappa_{\boldsymbol{\rho}}$. This advocates for the following question: *how to cover a graph G with its low tree-width subgraphs?* Formally, the question translates in finding $\boldsymbol{\rho}$ such that

$$\boldsymbol{\rho} \in \operatorname*{arg\,max}_{\boldsymbol{\rho} \in \mathcal{T}_k(G)} \min_{e \in E} \rho_e \tag{1.34}$$

where $\mathcal{T}_k(G)$ denotes the edge-induced sub-graphs of G of tree-width less than k. Once this covering is known (and assuming $\boldsymbol{\rho}$ has polynomial support), we achieve an approximation

by the following multiplicative constant:

$$\kappa_k(G) = \max_{\boldsymbol{\rho} \in \mathcal{T}_k(G)} \min_{e \in E} \rho_e \tag{1.35}$$

Chapter 2 will provide a full study of $\kappa_1(G)$, therefore solving the problem of obtaining a balanced covering of a graph from its spanning trees. In particular we will prove that the corresponding distribution $\boldsymbol{\rho}_1^*$ is tractable and that

$$\kappa_1(G) = \min_{S \subset V} \frac{|S| - 1}{|E(S)|} \tag{1.36}$$

where $E(S)$ denotes the subset of edges of G with both endpoints in S. In Figure 1-1 we give some visual examples of attainable balanced coverings for small graphs that were designed by hand. In Figure 1-2, more examples are given but their decomposition is not explicited.

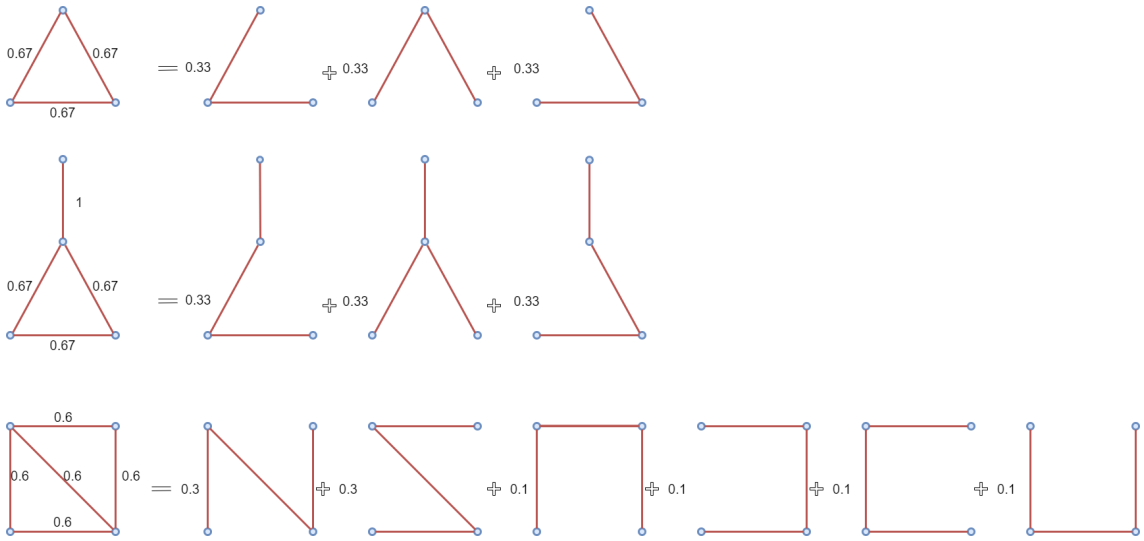

Figure 1-1: Three examples of balanced coverings of graphs. Notice that for the triangle, the most connected subgraph is the entire graph itself which yields $\kappa_1(G) = \frac{3-1}{3} = \frac{2}{3} = 0.67$ just like for the third example for which $\kappa_1(G) = \frac{4-1}{5} = \frac{3}{5} = 0.6$.

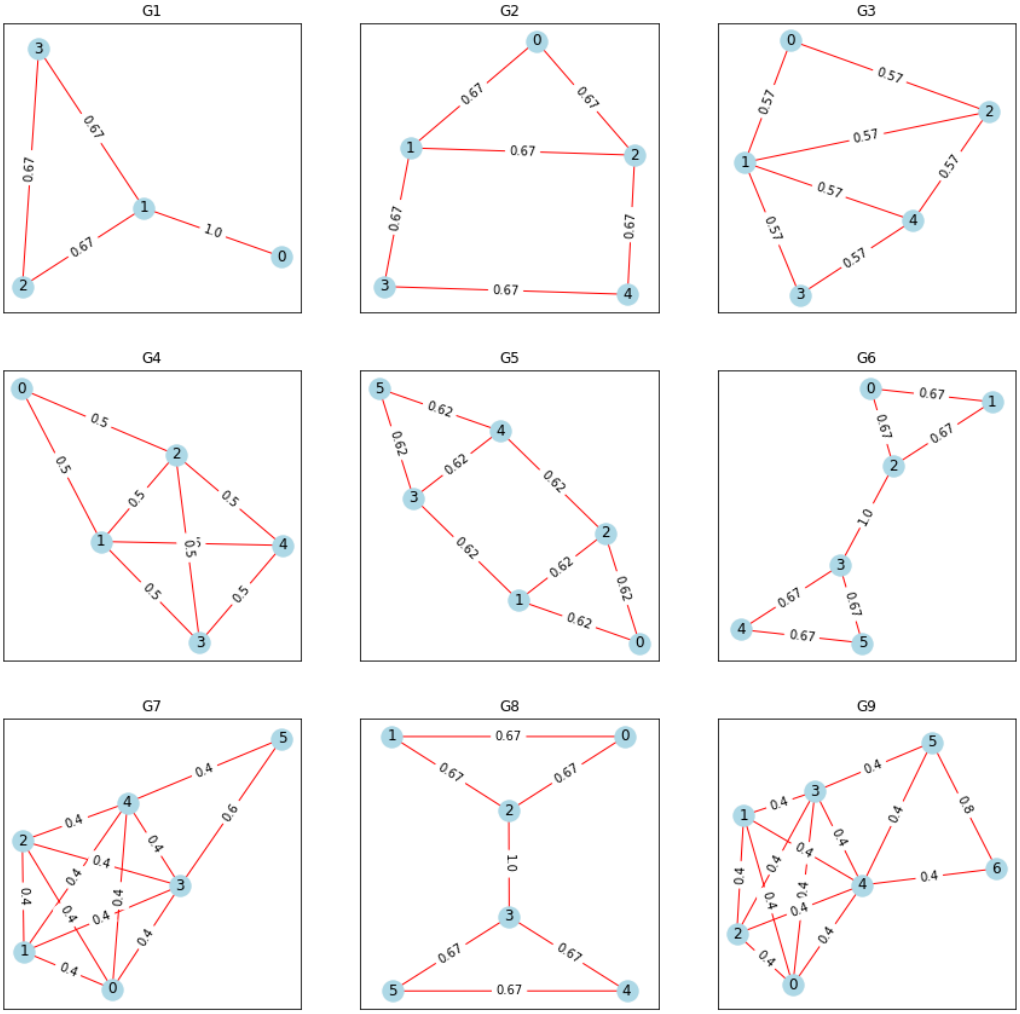

Figure 1-2: More examples of balanced covering of graphs, obtained with a LP solver. Note that for some graphs like $G7$, some of the edge probability for the $(3, 5)$ edge could be redistributed to the $(4, 5)$ edge for symmetry but this would not improve the optimum.

Chapter 2

Analysis of the tree-reweighted method

This chapter is an extraction from Romain Cosson, and Devavrat Shah. "Approximating the Log-Partition Function." *arXiv preprint arXiv:2102.10196* (2021), published in [15].

Abstract

Variational approximation, such as mean-field (MF) and tree-reweighted (TRW), provide a computationally efficient approximation of the log-partition function for a generic graphical model. TRW provably provides an upper bound, but the approximation ratio is generally not quantified. As the primary contribution of this work, we provide an approach to quantify the approximation ratio through the property of the underlying graph structure. Specifically, we argue that (a variant of) TRW produces an estimate that is within factor $\frac{1}{\sqrt{\kappa(G)}}$ of the true log-partition function for any discrete pairwise graphical model over graph G, where $\kappa(G) \in (0, 1]$ captures how far G is from tree structure with $\kappa(G) = 1$ for trees and $2/N$ for the complete graph over N vertices. As a consequence, the approximation ratio is 1 for trees, $\sqrt{(d+1)/2}$ for any graph with maximum average degree d, and $\overset{\beta\to\infty}{\approx} 1 + 1/(2\beta)$ for graphs with girth (shortest cycle) at least $\beta \log N$. In general, $\kappa(G)$ is the solution of a max-min problem associated with G that can be evaluated in polynomial time for any graph. Using samples from the uniform distribution over the spanning trees of G, we provide a near linear-time variant that achieves an approximation ratio equal to the inverse of square-root of minimal (across edges) effective resistance of the graph. We connect our results to the graph partition-based approximation method and thus provide a unified perspective.

Keywords: variational inference, log-partition function, spanning tree polytope, minimum effective resistance, min-max spanning tree, local inference

2.1 Introduction

The Setup. We consider a collection of N discrete valued random variables, $\mathbf{X} = (X_1, \dots, X_N)$, whose joint distribution is modeled as a pair-wise graphical model. Let $G = (V, E)$ rep-

resent the associated graph with vertices $V = \{1, \ldots, N\}$ representing N variables and $E \subset V \times V$ representing edges. Let each variable take value in a discrete set $\mathcal{X} \subset \mathbb{R}_+$. For $e \in E$, let $\phi_e : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ denote the edge potential and let $\theta_e \in \mathbb{R}_+$ denote the associated parameter. This leads to joint distribution with probability mass function

$$\mathbb{P}(\mathbf{X} = \mathbf{x}; \boldsymbol{\theta}) \propto \exp\Big(\sum_{e \in E} \theta_e \phi_e(x_e)\Big) \;=\; \frac{1}{Z(\boldsymbol{\theta})} \exp\Big(\sum_{e \in E} \theta_e \phi_e(x_e)\Big) \tag{2.1}$$

where $\mathbf{x} = (x_1, \ldots, x_N) \in \mathcal{X}^N$, x_e is short hand for (x_s, x_t) if $e = (s, t) \in E$, $\boldsymbol{\theta} = (\theta_e : e \in E) \in \mathbb{R}_+^{|E|}$ and normalizing constant or partition function $Z(\boldsymbol{\theta})$ is defined as

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{x} \in \mathcal{X}^N} \exp\Big(\sum_{e \in E} \theta_e \phi_e(x_e)\Big). \tag{2.2}$$

Such pairwise graphical models provide succinct description for complicated joint distributions. However, the key challenge in utilizing them (e.g. for inference) arises in estimating the partition function $Z(\boldsymbol{\theta})$. In this work, our interest is in computing logarithm of $Z(\boldsymbol{\theta})$, precisely

$$\Phi(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta}) \;=\; \log\Bigg[\sum_{\mathbf{x} \in \mathcal{X}^N} \exp\Big(\sum_{e \in E} \theta_e \phi_e(x_e)\Big)\Bigg]. \tag{2.3}$$

Computing $Z(\boldsymbol{\theta})$ is known to be computationally hard in general, i.e. #P-complete due to relation to counting discrete objects such as independent sets cf. [16, 17]. Due to reductions from discrete optimization problems to log-partition function computation, approximating $\Phi(\boldsymbol{\theta})$, even up to a multiplicative error, can be NP-hard cf. [18, 19, 20]. Therefore, the goal is to develop polynomial time (in N) approximation method for computing $\Phi(\boldsymbol{\theta})$ or $Z(\boldsymbol{\theta})$ with provable guarantees on the approximation error. Specifically, let ALG denote such an approximation method that takes problem description $(G, (\phi_e)_{e \in E}, \mathcal{X})$ as input and produces estimate $\widehat{\Phi}^{\mathrm{ALG}}(\boldsymbol{\theta})$ for $\Phi(\boldsymbol{\theta})$ for any given $\boldsymbol{\theta} \in \mathbb{R}_+^{|E|}$. Then, we define approximation ratio associated with ALG as $\alpha(G, \mathrm{ALG}) \geq 1$ as

$$\alpha(G, \mathrm{ALG}) = \sup_{\boldsymbol{\theta} \in \mathbb{R}_+} \max\Big(\frac{\Phi(\boldsymbol{\theta})}{\widehat{\Phi}^{\mathrm{ALG}}(\boldsymbol{\theta})}, \frac{\widehat{\Phi}^{\mathrm{ALG}}(\boldsymbol{\theta})}{\Phi(\boldsymbol{\theta})}\Big). \tag{2.4}$$

Prior Work. There is a long literature on developing computationally efficient approximation method for log-partition function with significant progress in the past two decades. We recall few relevant prior works here.

A collection of methods, classified as variational approximations, utilize the (Gibbs) variational characterization of the log-partition function when distribution (2.1) is viewed as a member of an exponential family, cf. [21, 19]. Specifically, $\Phi(\boldsymbol{\theta})$ can be viewed as a solution of a high-dimensional constrained maximization problem. By solving the prob-

lem with additional constraints, one obtains a valid lower bound such as that given by Mean-Field methods. By utilizing the convexity of $\Phi(\cdot)$ and restricting it to tree-structured sub-graphs of G, one obtains a valid upper bound such as that given by the tree-reweighted (TRW) method. By relaxing the constraints and adapting the objective to allow for pairwise pseudo-marginals, one obtains heuristics such as Belief Propagation (BP) via Bethe approximation [22, 23]. While BP does not provide provable upper or lower bound in general, for graphs with large-girth such as sparse random graphs and distributions with spatial decay of correlation, it provides an excellent approximation cf. [24]. The spatial decay of correlation property has been further exploited to obtain deterministic Fully Polynomial Time Approximation Schemes (FPTAS) for various counting problems, i.e. computing partition functions cf. [18, 25, 26, 27]. The approximation error of belief propagation for computing log-partition function has been studied through connection to loop calculus as well cf. [28, 29].

In another line of works, graph partitioning based methods have been proposed to provide Polynomial Time Approximation Schemes (PTAS) for a class of graphs that satisfy certain graph partitioning properties which includes minor-excluded graphs [30] or graphs with polynomial growth [31].

In summary, despite the progress, the approximation ratio $\alpha(G, \text{ALG})$ for any of the known variational approximation methods ALG remains undetermined.

Summary of Contributions. As the main contribution, for a simple variant of tree-reweighted (TRW) method, denoted as TRW′, we quantify $\alpha(G, \text{TRW}')$ for any G. The TRW′ is described in Section 2.3 and produces an estimate of $\Phi(\cdot)$ in polynomial time. Specifically, we establish

Theorem 2.1.1. *For any graph G, the approximation ratio of TRW′ is such that*

$$\alpha(G, \textit{TRW}') \leq 1/\sqrt{\kappa(G)} \quad \textit{where} \quad \kappa(G) = \min_{S \subset V} \frac{|S| - 1}{|E(S)|}, \tag{2.5}$$

with $E(S) = E \cap (S \times S)$ for any $S \subset V$.

The term $\kappa(G)$ captures the proximity of G with respect to the tree structure across all of its induced sub-graphs: for $S \subset V$, the induced subgraph $(S, E(S))$ would have at most $|S|-1$ if it were cycle free, but it has $|E(S)|$ edges. Therefore, the ratio of $(|S|-1)/|E(S)|$ measures how far it is from tree – 1 if connected tree and $2/|S|$ if complete graph. The minimum over all possible $S \subset V$ of this ratio captures how far G is from a tree structure.

Using this characterization, we provide bounds on $\alpha(G, \text{TRW}')$ in terms of various simpler graph properties in Section 2.4.4. Specifically, we show that for any graph with maximum average vertex degree $d \geq 1$, $\alpha(G, \text{TRW}') \leq \sqrt{(d+1)/2}$. And for graphs with girth (i.e. length of shortest cycle) $g > 3$, $\alpha(G, \text{TRW}') \leq \sqrt{\frac{1+N^{2/(g-3)}}{2(1-1/g)}}$: for $g \geq \beta \log N$, it is $\left(1 + \frac{1}{2\beta} + o(\frac{1}{\beta})\right)$ for large β. This means that for *any* G with large ($\gg \log N$) girth, $\alpha(G, \text{TRW}') \approx 1$.

In general, we establish that $\kappa(G)$ can be evaluated in polynomial time for any graph G by solving an appropriate linear program on the (polynomially-)extended spanning tree polytope. This is explained in Section 2.4.

The tree-reweighted variant TRW$'$ considered here requires solving a certain optimization problem over the tree polytope of the graph G. Though it can be computed in polynomial time, it can be quite involved. With an eye towards near linear-time (in $|E|$) computation, a variant that instead of optimizing over the tree polytope simply considers a feasible point in the tree polytope that corresponds to the uniform distribution over spanning trees of G. Using the near-linear time sampling of spanning tree from [32], we provide a randomized approximation method. It's approximation ratio $\alpha(G)$ is bounded above by $1/\sqrt{\min_{e\in E} r_e}$ where $r_e \geq 0$ is the effective resistance of $e \in E$ for the graph $G = (V, E)$ (see (2.39) for precise definition). While in general, this provides a weaker approximation guarantee than that of TRW', for graphs with vertex degree bounded by d it leads to a similar guarantee of $\alpha(G) \leq \sqrt{(d+1)/2}$.

We show that the results based on graph partitioning cf. [30, 31] can be recovered as a natural extension of the variant of TRW introduced in this work by allowing for general graphs with bounded tree-width beyond trees.

We take note of the fact that though results discussed in this work are primarily for the variant of TRW described in Section 2.3, as an immediate consequence of our results, $\alpha(G, \text{TRW}) \leq 1/\kappa(G)$, i.e. it is bounded by the square of that derived in Theorem 2.1.1. As discussed in Section 2.7, understanding the tightness of this characterization especially for TRW remains an important open direction.

Outline of Paper. In Section 2.2, we provide some preliminaries including recalling the tree-reweighted (TRW) method. In Section 2.3, we provide a modification of TRW and characterize its approximation guarantee. In Section 2.4, we provide a linear optimization characterization of the approximation guarantee which leads to the proof of Theorem 2.1.1. We discuss implications of Theorem 2.1.1 for various classes of graphs as well. In Section 2.5, we present a near linear-time variant of modified TRW based on sampling from the uniform distribution of spanning tree over G. We derive approximation guarantees for the resulting method in terms of the effective resistance of the graph and derive its implications. In Section 2.6, we discuss connection with graph partitioning methods by extending the modified TRW of Section 2.3 to allow for bounded tree-width subgraphs beyond trees. We argue how results of [30, 31] follow naturally. Section 2.7 discusses directions for future work.

2.2 Preliminaries and Background

2.2.1 Variational Characterization, Mean-Field Approximation and Belief Propagation

We start by recalling the variational characterization of the log-partition function $\Phi(\cdot)$. Let $\mathcal{P}(\mathcal{X}^N)$ denote the space of all probability distributions over $\mathcal{X}^N$. Then, the Gibbs variational characterization states that

$$\Phi(\boldsymbol{\theta}) = \sup_{q \in \mathcal{P}(\mathcal{X}^N)} \mathbb{E}_{\mathsf{x} \sim q}\Big(\sum_e \theta_e \phi_e(\mathsf{x}_e)\Big) + H(q), \tag{2.6}$$

where $H(q) = -\mathbb{E}_{\mathsf{x}\sim q}(\log(q(\mathsf{x})))$ is the entropy of q. While computationally (2.6) does not provide tractable solution for evaluating $\Phi(\cdot)$, it provides a framework to develop approximation methods – such methods, inspired by this characterization, are called *variational approximations*.

As mentioned earlier, the classical mean-field consists in relaxing $\mathcal{P}(\mathcal{X}^N)$ to the space of independent distributions over $\mathcal{X}^N$ denoted as $\mathcal{I}(\mathcal{X}^N)$, i.e. $\mathcal{I}(\mathcal{X}^N) = \{q \in \mathcal{P}(\mathcal{X}^N) : q(X_1, \ldots, X_N) = \prod_{i=1}^N (X_i)\}$. By restricting optimization in (2.6) to $\mathcal{I}(\mathcal{X}^N)$, the resulting answer is a lower bound on $\Phi(\boldsymbol{\theta})$. And mean-field method precisely attempts to solve such a lower-bound.

It turns out that (2.6) is solvable efficiently for tree-structured graph. Specifically, if G is a connected tree, i.e. G is connected with $|E| = N - 1$, then any distribution satisfying (2.1) can be re-parametrized as

$$\mathbb{P}(\mathbf{x}; \boldsymbol{\theta}) = \prod_{u \in V} \mathbb{P}_{X_u}(x_u) \prod_{(u,v)\in E} \frac{\mathbb{P}_{X_u,X_v}(x_u, x_v)}{\mathbb{P}_{X_u}(x_u)\mathbb{P}_{X_v}(x_v)}. \tag{2.7}$$

In the expression above, $\mathbb{P}_{X_u}(\cdot)$ denotes the marginal distribution of $X_u, u \in V$ and $\mathbb{P}_{X_u,X_v}(\cdot,\cdot)$ denotes the pairwise marginal distribution of (X_u, X_v) for any edge $e = (u, v) \in E$. The Belief Propagation (or sum-product) algorithm can compute these marginal distributions efficiently for tree graphs using only knowledge of $\boldsymbol{\theta}$ and $\phi_e, e \in E$ but not requiring $\Phi(\boldsymbol{\theta})$. It utilizes $O(|\mathcal{X}|^2 N)$ computation time, when implemented efficiently. Therefore, $Z(\boldsymbol{\theta})$ and hence $\Phi(\boldsymbol{\theta})$ can be computed for tree graphs using $O(|\mathcal{X}|^2 N)$ computations.

Indeed, the re-parametrization of the form (2.7) was a basis for the Belief Propagation (BP) algorithm for generic graphical models and also led to the so called Bethe Approximation of (2.6), cf. [22]. However, it does not result in a provably upper or lower bound in general (with few exceptions).

To obtain an upper bound on $\Phi(\cdot)$, its convexity was exploited in [1] along with the fact that (2.6) is solvable efficiently for tree-structured graph. This resulted into tree-reweighted (TRW) algorithm which we describe next.

2.2.2 Tree-Reweighted (TRW): An Upper Bound on $\Phi(\cdot)$

Recall that a spanning tree T is a subgraph of G that contains all vertices V and a subset of edges E so that the resulting subgraph is a tree, i.e. does not have a cycle. Let $\mathcal{T}(G)$ be the set of all spanning trees of G. We shall denote a distribution on $\mathcal{T}(G)$ as $\boldsymbol{\rho} = (\rho^T)_{T\in\mathcal{T}(G)}$ where $\rho^T \geq 0$ for all $T \in \mathcal{T}(G)$, $\sum_{T\in\mathcal{T}(G)} \rho^T = 1$. The space of all distributions on $\mathcal{T}(G)$ is denoted by $\mathcal{P}(\mathcal{T}(G))$. For simplicity, we shall drop notation of G at times when it is clear from the context and denote it simply as $\mathcal{P}(\mathcal{T})$. A distribution $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T})$ induces for all edge $e \in E$ a probability ρ_e that this edge will appear in a tree selected from ρ,

$$\rho_e = \mathbb{P}_{\mathsf{T}\sim\boldsymbol{\rho}}\big(e \in \mathsf{T}\big) \;=\; \sum_{T\in\mathcal{T}(G)} \rho^T \mathbf{1}(e \in T). \tag{2.8}$$

Note that in the above, we have abused notation using T as a spanning tree as well as the set of edges constituting it. We shall continue using this notation since the all spanning trees have the same set of vertices, V and only the edges differ (among subsets of E). Also note another convenient abuse of notation: given $\boldsymbol{\rho}$, ρ^T denotes probability of $T \in \mathcal{T}(G)$ while ρ_e is the marginal probability of edge $e \in E$ being present in tree as per $\boldsymbol{\rho}$ and satisfies $\sum_{e\in E} \rho_e = N - 1$. Given $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T}(G))$, we now define $\kappa_{\boldsymbol{\rho}}$ as

$$\kappa_{\boldsymbol{\rho}} = \min_{e\in E} \rho_e. \tag{2.9}$$

For any $\boldsymbol{\theta} \in \mathbb{R}_+^{|E|}$, define its support as $s(\boldsymbol{\theta}) = \{e \in E : \theta_e \neq 0\}$. Given a spanning tree $T \in \mathcal{T}(G)$, let $\theta^T \in \mathbb{R}_+^{|E|}$ be such that $s(\theta^T) \subset T$. Let $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T})$ along with $(\theta^T)_{T\in\mathcal{T}}$ be such that $\sum_{T\in\mathcal{T}} \rho^T\theta^T = \theta$. That is, $\mathbb{E}_{\mathsf{T}\sim\boldsymbol{\rho}}\big[\boldsymbol{\theta}^{\mathsf{T}}\big] = \boldsymbol{\theta}$. Therefore, we can write

$$\Phi(\boldsymbol{\theta}) = \Phi\big(\mathbb{E}_{\mathsf{T}\sim\boldsymbol{\rho}}\big[\theta^{\mathsf{T}}\big]\big). \tag{2.10}$$

It has been well established that $\Phi : \mathbb{R}_+^{|E|} \to \mathbb{R}$ is a convex function. Precisely, for any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}_+^{|E|}$ and $\gamma \in [0, 1]$

$$\Phi(\gamma\boldsymbol{\theta}_1 + (1-\gamma)\boldsymbol{\theta}_2) \leq \gamma\Phi(\boldsymbol{\theta}_1) + (1-\gamma)\Phi(\boldsymbol{\theta}_2). \tag{2.11}$$

From (2.10) and (2.11), it follows from Jensen's inequality that

$$\Phi(\boldsymbol{\theta}) \leq \mathbb{E}_{\mathsf{T}\sim\boldsymbol{\rho}}\big[\Phi(\theta^{\mathsf{T}})\big] \;=\; \sum_{T\in\mathcal{T}} \rho^T\Phi(\theta^T). \tag{2.12}$$

Since the upper bound (A.6) holds for any $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T})$ and $(\theta^T)_{T\in\mathcal{T}}$ such that $\sum_{T\in\mathcal{T}} \rho^T\theta^T =$

$\boldsymbol{\theta}$ we can optimize on these two parameters to obtain

$$\Phi(\boldsymbol{\theta}) \leq \inf_{\sum_{T\in\mathcal{T}} \rho^T\theta^T=\boldsymbol{\theta}} \left(\sum_{T\in\mathcal{T}} \rho^T \Phi(\theta^T)\right) \equiv U^{\text{TRW}}(\boldsymbol{\theta}). \tag{2.13}$$

As established in [1], this seemingly complicated optimized bound, $U^{\text{TRW}}(\boldsymbol{\theta})$, can be computed via an iterative *tree-reweighted message-passing* algorithm through the dual of the above optimization problem. While this is a valid upper bound, how tight the upper bound is for a given graphical model is not quantified in the literature. And this is precisely the primary contribution of this work.

2.3 Algorithm and Approximation Guarantee

Modified Tree-Reweighted: TRW′. We describe a simple variant of TRW that enables us to bound the approximation ratio of the estimation of Φ using properties of G. We start with some useful notations. Given $\boldsymbol{\theta} = (\theta_e)_{e\in E} \in \mathbb{R}_+^{|E|}$, $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T}(G))$ and spanning tree $T \in \mathcal{T}(G)$ of graph G, define "projection" operations

$$\begin{aligned} &\Pi^T : \mathbb{R}_+^{|E|} \to \mathbb{R}_+^{|E|} \quad \text{where} \quad \Pi^T(\boldsymbol{\theta}) = \big(\mathbf{1}(e\in T)\theta_e\big)_{e\in E} \\ &\Pi_{\boldsymbol{\rho}}^T : \mathbb{R}_+^{|E|} \to \mathbb{R}_+^{|E|} \quad \text{where} \quad \Pi_{\boldsymbol{\rho}}^T(\boldsymbol{\theta}) = \big(\frac{1}{\rho_e}\mathbf{1}(e\in T)\theta_e\big)_{e\in E}. \end{aligned} \tag{2.14}$$

With these notations, for a given $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T}(G))$ define

$$L_{\boldsymbol{\rho}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathsf{T}\sim\boldsymbol{\rho}}(\Phi(\Pi^{\mathsf{T}}(\boldsymbol{\theta}))) = \sum_{T\in\mathcal{T}(G)} \rho^T \Phi(\Pi^T(\boldsymbol{\theta})), \tag{2.15}$$

$$U_{\boldsymbol{\rho}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathsf{T}\sim\boldsymbol{\rho}}(\Phi(\Pi_{\boldsymbol{\rho}}^{\mathsf{T}}(\boldsymbol{\theta}))) = \sum_{T\in\mathcal{T}(G)} \rho^T \Phi(\Pi_{\boldsymbol{\rho}}^T(\boldsymbol{\theta})). \tag{2.16}$$

For a given $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T}(G))$, one obtains an estimate of $\Phi(\boldsymbol{\theta})$

$$\widehat{\Phi}_{\boldsymbol{\rho}}(\boldsymbol{\theta}) = \sqrt{L_{\boldsymbol{\rho}}(\boldsymbol{\theta}) U_{\boldsymbol{\rho}}(\boldsymbol{\theta})}. \tag{2.17}$$

For reasons that will become clear, TRW′ outputs $\widehat{\Phi}_{\boldsymbol{\rho}^\star}(\boldsymbol{\theta})$ where $\boldsymbol{\rho}^\star = \boldsymbol{\rho}^\star(G)$ defined as

$$\boldsymbol{\rho}^\star(G) \in \operatorname*{arg\,max}_{\boldsymbol{\rho}\in\mathcal{P}(\mathcal{T}(G))} \big(\min_{e\in E} \rho_e\big) \quad \text{and} \quad \kappa_{\boldsymbol{\rho}^\star(G)} = \max_{\boldsymbol{\rho}\in\mathcal{P}(\mathcal{T}(G))} \big(\min_{e\in E} \rho_e\big). \tag{2.18}$$

Guarantee. The lemma below quantifies the approximation ratio for TRW′. It's proof is in Appendix A.1.

Lemma 2.3.1. *Given $\boldsymbol{\theta} \in \mathbb{R}_+^{|E|}$, TRW′ produce $\widehat{\Phi}_{\boldsymbol{\rho}^\star}(\boldsymbol{\theta})$ with $\boldsymbol{\rho}^\star = \boldsymbol{\rho}^\star(G)$ as defined in*

(2.18). *Then,*

$$\alpha(G, \mathit{TRW}') \leq \frac{1}{\sqrt{\kappa_{\boldsymbol{\rho}^\star}}}. \tag{2.19}$$

2.4 $\kappa_{\boldsymbol{\rho}^\star(G)}$: Efficient computation, characterization

Lemma 2.3.1 establishes the approximation guarantee for TRW′ as claimed in Theorem 2.1.1 with caveat that it is in terms of $\kappa_{\boldsymbol{\rho}^\star(G)}$ while Theorem 2.1.1 states it in form of $\kappa(G)$ as defined in (2.5). In this section, we shall establish the characterization of $\kappa_{\boldsymbol{\rho}^\star(G)} = \kappa(G)$ and in the process argue that it can be evaluated in polynomial time for any graph G. This characterization will allow us to bound $\kappa(G)$ for certain classes of graphs to obtain meaningful intuition.

2.4.1 Computing $\boldsymbol{\rho}^\star(G)$ and $\kappa_{\boldsymbol{\rho}^\star(G)}$ efficiently

Spanning Tree Polytope. We define a notion of spanning tree polytope for a given graph G. Recall that $\mathcal{T}(G)$ is the set of all spanning trees of G. For any tree $T \in \mathcal{T}(G)$, we shall utilize the notation of $\chi^T = [\chi^T_e] \in \{0,1\}^E$ to represent the characteristic vector of the tree T defined such that

$$\forall e \in E : \chi^T_e = \mathbf{1}(e \in T). \tag{2.20}$$

Given this notation, we define the polytope of spanning trees of G, denoted $\mathsf{P}^{\text{tree}}(G)$, as the convex hull of their characteristic vectors. That is,

$$\mathsf{P}^{\text{tree}}(G) = \Big\{\mathbf{v} \in [0,1]^E : \mathbf{v} = \sum_{T \in \mathcal{T}(G)} \rho^T \chi^T, \sum_{T \in \mathcal{T}(G)} \rho^T = 1,\ \rho_T \geq 0, \forall T \in \mathcal{T}(G)\Big\}. \tag{2.21}$$

The weights $(\rho^T)_{T \in \mathcal{T}(G)}$ can be viewed as probability distribution on $\mathcal{T}(G)$, i.e. an element of $\mathcal{P}(\mathcal{T}(G))$. Therefore $\mathbf{v} = \sum_{T \in \mathcal{T}(G)} \rho^T \chi^T$ corresponds to a vector representing the probabilities that edges in E will be present in in $\mathsf{T} \sim \boldsymbol{\rho} = (\rho^T)$, i.e. $\mathbf{v} = \mathbb{E}_{\mathsf{T} \sim \boldsymbol{\rho}}[\mathbf{1}(e \in \mathsf{T})]$. That is, $\mathbf{v} = (\rho_e)_{e \in E}$ as defined in (2.8). Therefore, we shall abuse notation and write

$$\mathsf{P}^{\text{tree}}(G) = \big\{(\rho_e)_{e \in E} \mid (\rho^T)_{T \in \mathcal{T}(G)} \in \mathcal{P}(\mathcal{T}(G))\big\}. \tag{2.22}$$

[33] gave the following characterization of the spanning tree polytope:

$$\mathsf{P}^{\text{tree}}(G) = \left\{(v_e)_{e \in E} \in \mathbb{R}^E_+ \;\middle|\; \begin{array}{r} \forall S \subset E : v(E(S)) \leq |S| - 1 \\ v(E) = |V| - 1 \end{array}\right\}, \tag{2.23}$$

where $v(E(S)) = \sum_{e \in E(S)} v_e$.

Efficient Separation Oracle. A polytope $\mathsf{P} \subset \mathbb{R}^n$, defined through a set of linear constraints, is said to have a separation oracle if there exists a polynomial time algorithm in n which for given any $x \in \mathbb{R}^n$ can determine whether $x \in \mathsf{P}$ or not; and output a violated constraint if $x \notin \mathsf{P}$. Edmond's characterization of the spanning tree polytope, though has an exponential number of constraints, admits an efficient separation oracle. Such an efficient separation oracle is defined explicitly via a min-cut reduction, see [34, Chapter 4.1].

Complexity of Linear Programming. Consider a linear program where the goal is to find a minimum of a linear objective function over a polytope defined by finitely many linear constraints. Such a linear program can be solved in polynomial time (in size of problem description) via the Ellipsoid method if the polytope admits an efficient separation oracle, see [35, Theorem 8.5] for example. Given that the spanning tree polytope has an efficient separation oracle, optimizing a linear objective over it can be solved efficiently. Of course, due to the structure of the trees, a greedy algorithm like that of Kruskal's may be a lot more direct for solving such a linear program. Having said that, the benefit of efficient separation oracle becomes apparent as soon as we consider additional linear constraints beyond those described in $\mathsf{P}^{\text{tree}}(G)$. Indeed, such approaches have found utility in solving other problems, liked solving bounded-degree maximum-spanning-tree relaxations like in [36].

Augmented Spanning Tree Polytope. We consider a reformulation of the max-min problem in (2.18). To that end consider the following augmented spanning tree polytope:

$$\mathsf{P}^{\text{tree}}_{\min}(G) = \left\{ (z, (v_e)_{e\in E}) \in \mathbb{R} \times \mathbb{R}_+^{|E|} \;\middle|\; \begin{array}{r} \forall e \in E : z \le v_e \\ \forall S \subset E : v(E(S)) \le |S| - 1 \\ v(E) = |V| - 1 \end{array} \right\}. \tag{2.24}$$

With this notation, we can re-write $\kappa_{\boldsymbol{\rho}^\star(G)}$ as per (2.18) as

$$\kappa_{\boldsymbol{\rho}^\star(G)} = \max_{(v_e)_{e\in E} \in \mathsf{P}^{\text{tree}}} \{\min_{e\in E} v_e\} = \max_{(z,(v_e)_{e\in E}) \in \mathsf{P}^{\text{tree}}_{\min}} z. \tag{2.25}$$

Next, we argue that $\mathsf{P}^{\text{tree}}_{\min}$ admits an efficient separation oracle as follows. The separation oracle for $\mathsf{P}^{\text{tree}}_{\min}$ takes $(z, (v_e)_{e\in E})$ as input. It first checks that all $|E|$ constraints of the form $z \le v_e$ are satisfied. If one is not satisfied, then the oracle outputs this constraint. If all constraints are satisfied, the algorithm runs the separation oracle of P^{tree} on $(v_e)_{e\in E}$ and reproduces its output. Since $|E| \le N^2$ and P^{tree} has an efficient separation oracle, this leads to polynomial time separation oracle for $\mathsf{P}^{\text{tree}}_{\min}$.

Efficient computation of $\boldsymbol{\rho}^\star(G)$ and $\kappa_{\boldsymbol{\rho}^\star(G)}$. From the linear program formulation (2.25) and from the efficient separation oracle as defined above, we can compute $\kappa_{\boldsymbol{\rho}^\star(G)}$ in polyno-

mial time using the Ellipsoid algorithm. Note that this does not directly provides $\boldsymbol{\rho}^\star(G) \in \mathcal{P}(\mathcal{T}(G))$ since the representation in P^{tree} corresponds to the edge probabilities $(\rho^\star(G)_e)_{e\in E}$. However, $(\rho^\star(G)_e)_{e\in E}$ is a convex combination of extreme points of P^{tree}, which correspond to the spanning trees of G. Since P^{tree} has efficient separation oracle, we can recover a decomposition of $(\rho^\star(G)_e)_{e\in E}$ in terms of convex combination of characteristic vectors weighted by $(\rho^\star(G)^T)_{T\in\mathcal{T}(G)}$ and such that at most $|E|$ of these weights are strictly positive, see details in [37, Theorem 3.9].

2.4.2 Characterizing $\kappa_{\boldsymbol{\rho}^\star(G)} = \kappa(G)$

We wish to establish $\kappa_{\boldsymbol{\rho}^\star(G)} = \kappa(G)$, i.e. we want to establish

$$\kappa_{\boldsymbol{\rho}^\star(G)} = \max_{(v_e)_{e\in E}\in\mathsf{P}^{\text{tree}}} \{\min_{e\in E} v_e\} = \min_{S\subset V} \frac{|S|-1}{|E(S)|}. \tag{2.26}$$

Upper bound: $\kappa_{\boldsymbol{\rho}^\star(G)} \leq \frac{|S|-1}{|E(S)|}$. The upper bound is immediately given by Edmond's characterisation of the spanning tree polytope. For any $(\rho_e)_{e\in E} \in \mathsf{P}^{\text{tree}}$ and any $S \subset V$:

$$|E(S)|\big(\min_{e\in E} \rho_e\big) \leq \big(\sum_{e\in E(S)} \rho_e\big) \;=\; \rho(E(S)) \;\leq\; |S|-1. \tag{2.27}$$

That is, for any $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T}(G))$

$$\kappa_{\boldsymbol{\rho}} \leq \min_{S\subset V} \frac{|S|-1}{|E(S)|}. \tag{2.28}$$

And hence it holds for $\boldsymbol{\rho}^\star(G)$ as well.

Lower bound: $\kappa_{\boldsymbol{\rho}^\star(G)} \geq \frac{|S|-1}{|E(S)|}$. To establish the lower bound, we need a few additional results. To start with, we define a dual of the optimization problem (2.25) to characterize $\kappa_{\boldsymbol{\rho}^\star(G)}$. By strong duality it follows that

$$\kappa_{\boldsymbol{\rho}^\star(G)} = \max_{\boldsymbol{\rho}\in\mathcal{P}(\mathcal{T})} \min_{e\in E} \sum_{T\in\mathcal{T}} \mathbf{1}(e\in T)\rho^T = \min_{\mathbf{w}\in\mathcal{P}(E)} \max_{T\in\mathcal{T}} \sum_{e\in E} \mathbf{1}(e\in T)w_e, \tag{2.29}$$

where $\mathcal{P}(E) = \{\mathbf{w} = (w_e)_{e\in E} : \sum_{e\in E} w_e = 1, w_e \geq 0 \;\forall\; e \in E\}$. Table 2.1 provides the precise primal and dual formulation associated with $\kappa_{\boldsymbol{\rho}^\star(G)}$ justifying (2.29). We state the following Lemma characterizing an optimal solution of **Dual**, whose proof is in Appendix A.2.

Lemma 2.4.1. *There exists an optimal solution of* ***Dual****,*

$$\mathbf{w}^\star \in \arg\min_{\mathbf{w}\in\mathcal{P}(E)} \max_{T\in\mathcal{T}} \sum_{e\in E} \mathbf{1}(e\in T)w_e,$$

	Primal	**Dual**
Objective	$\max\ z$	$\min\ y$
Variables / Constraints	$\begin{aligned} z \quad &\in \mathbb{R} \\ \forall T \in \mathcal{T} : \rho_T \quad &\in \mathbb{R}_+ \end{aligned}$	$\begin{aligned} \textstyle\sum_{e\in E} w_e \quad &= 1 \\ \forall T \in \mathcal{T} : y - \textstyle\sum_{e\in T} w_e \quad &\geq 0 \end{aligned}$
Constraints / Variables	$\begin{aligned} \textstyle\sum_{T\in\mathcal{T}} \rho_T \quad &= 1 \\ \forall e \in E : \textstyle\sum_{T\ni e} \rho_T - z \quad &\geq 0 \end{aligned}$	$\begin{aligned} y \quad &\in \mathbb{R} \\ \forall e \in E : w_e \quad &\in \mathbb{R}_+ \end{aligned}$

Table 2.1: The primal (cf. (2.25)) and dual formulation of $\kappa_{\boldsymbol{\rho}^\star(G)}$.

such that all non-zero components of $\mathbf{w}^\star$ *take the identical values: i.e.* $|\{w_e : w_e \neq 0, e \in E\}| = 1$.

As per Lemma 2.4.1, consider an optimal solution of **Dual**, $\mathbf{w}^\star$, that assigns constant value to a subset $F \subset E$ edges and 0 to edges $E\backslash F$: let $\mathbf{w}^\star = (w_e^\star)_{e\in E}$ with $w_e^\star = \frac{1}{|F|}$ for $e \in F$ and $w_e^\star = 0$ for $e \in E\backslash F$. Let $V(F) \subset V$ be set of all vertices corresponding to the end points of edges in F making a subgraph $(V(F), F)$ of G. Let $c(F) \geq 1$ denote the number of connected components of $(V(F), F)$. Per **Dual**, given $\mathbf{w}^\star$, $\kappa_{\boldsymbol{\rho}^\star(G)}$ equals the weight of the maximum weight spanning tree in G with edges assigned weights as per $\mathbf{w}^\star$. Such a maximum weight spanning tree must select as many edges as possible from F: since it has $c(F)$ connected components and $V(F)$ vertices, it can select at most $|V(F)| - c(F)$ such edges and any each such edge has weight $1/|F|$. The rest of the edges in the maximum weight spanning tree will carry weight 0. Thus, the total weight of such a maximum weight spanning tree is $(|V(F)| - c(F))/|F|$. This gives us an equivalent characterization for $\kappa_{\boldsymbol{\rho}^\star(G)}$ as

$$\kappa_{\boldsymbol{\rho}^\star(G)} = \min_{F\subset E} \frac{|V(F)| - c(F)}{|F|}. \tag{2.30}$$

Now we state a Lemma, whose proof is in Appendix A.3, which relates the characterization of (2.30) with that of (2.5).

Lemma 2.4.2. *For any graph* G,

$$\min_{S\subset V} \frac{|S| - 1}{|E(S)|} = \min_{F\subset E} \frac{|V(F)| - c(F)}{|F|}. \tag{2.31}$$

2.4.3 Proof of Theorem 2.1.1

The primary claim of Theorem 2.1.1 is that $\alpha(G, \text{TRW}') \leq 1/\sqrt{\kappa(G)}$. As per Lemma 2.3.1, we have that $\alpha(G, \text{TRW}') \leq 1/\sqrt{\kappa_{\boldsymbol{\rho}^\star(G)}}$. As per arguments in Section 2.4.2, we have that $\kappa_{\boldsymbol{\rho}^\star(G)} = \kappa(G)$. Therefore, we conclude the proof of Theorem 2.1.1.

2.4.4 Evaluating $\kappa(G)$ For a Class of Graphs

As established in Section 2.4.1, $\kappa(G)$ or $\kappa_{\boldsymbol{\rho}^\star(G)}$ can be computed in polynomial time for any G. Here, we attempt to obtain a (lower) bound on $\kappa(G)$ in terms of simple graph properties. To that end, we obtain the following for graphs with bounded maximum average degree.

Lemma 2.4.3. *For a graph $G = (V, E)$, let $\bar{d} = \max_{S \subset V} \frac{2|E(S)|}{|S|}$ denote the maximum average degree. Then*

$$\kappa(G) \geq \frac{2}{\bar{d}+1}. \tag{2.32}$$

For graphs with large girth, we obtain the following.

Lemma 2.4.4. *For a graph $G = (V, E)$, let $g > 3$ be its girth, i.e. the length of the shortest cycle. Then*

$$\kappa(G) \geq \frac{2}{1 + N^{\frac{2}{g-3}}}(1 - \frac{1}{g}). \tag{2.33}$$

The proofs of Lemmas 2.4.3 and 2.4.4 are presented in Appendix A.4. As per Lemma 2.4.4, for $g = \beta \log N$ for $\beta \gg 1$ and N large enough

$$\kappa(G) \geq \frac{2}{1 + N^{\frac{2}{g-3}}}(1 - \frac{1}{g}). \tag{2.34}$$

Therefore,

$$\alpha(G, \text{TRW}') \leq \frac{1}{\sqrt{\kappa(G)}} \approx 1 + \frac{1}{2\beta}. \tag{2.35}$$

2.5 A Near Linear-Time Variant of TRW

2.5.1 Algorithm

The TRW′ requires finding $\boldsymbol{\rho}^\star(G)$. As discussed in Section 2.4, it can be computed efficiently. However it can be cumbersome and having near-linear (in $|E|$) time variant can be more attractive in practice. With this as a motivation, we propose utilizing uniform distribution on $\mathcal{T}(G)$, denoted as $\mathbf{u} \equiv \mathbf{u}(\mathcal{T}(G))$, in place of $\boldsymbol{\rho}^\star(G)$ in TRW′ . The challenge is it has very large support, $\mathcal{T}(G)$, and hence it is difficult to compute $L_{\mathbf{u}}(\boldsymbol{\theta}), U_{\mathbf{u}}(\boldsymbol{\theta})$. But, both of these quantities are averages, with respect to $\mathbf{u}$, of a certain functional. And it is feasible to sample spanning tree uniformly at random for any G in near-linear time. Therefore, we can draw n samples from the distribution $\mathbf{u}$ and consider the empirical distribution $\hat{\mathbf{u}}^n$ to compute estimates $L_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta}), U_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})$ with few samples. This is precisely the algorithm.

To that end, consider n trees $\mathsf{T}_1, \ldots, \mathsf{T}_n$ sampled uniformly at random from $\mathcal{T}(G)$. Compute

$$\hat{u}_e^n = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(e \in \mathsf{T}_i),\ \forall\, e \in E, \quad L_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta}) \;=\; \frac{1}{n}\sum_{i=1}^{n} \Phi(\Pi^{\mathsf{T}_i}(\boldsymbol{\theta})), \quad U_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \Phi(\Pi^{\mathsf{T}_i}_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})), \tag{2.36}$$

where $\hat{\mathbf{u}}^n = (\hat{u}_e^n)_{e\in E}$. Given this, produce the estimate

$$\widehat{\Phi}_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta}) = \sqrt{L_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta}) U_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})}. \tag{2.37}$$

2.5.2 Guarantees

Given a graph G, remember that $\kappa_{\mathbf{u}}(G) = \min_{e\in E} u_e$ with $\mathbf{u}$ being the uniform distribution on $\mathcal{T}(G)$ and $u_e = \mathbb{E}_{\mathsf{T}\sim\mathbf{u}}[\mathbf{1}(e \in \mathsf{T})]$. We state the following Lemma, whose proof can be found in Appendix A.5.

Lemma 2.5.1. *Given $\epsilon > 0$ and $d > 0$, for $n \geq O\left(\log(\frac{N}{\delta})\kappa_{\mathbf{u}}(G)^{-2}\epsilon^{-2}\right)$ and ϵ sufficiently small, with probability at least $1-\delta$*

$$\max_{\boldsymbol{\theta}\in\mathbb{R}^E}\left(\frac{\Phi(\boldsymbol{\theta})}{\widehat{\Phi}_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})}, \frac{\widehat{\Phi}_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})}{\Phi(\boldsymbol{\theta})}\right) \leq \frac{1+\epsilon}{\sqrt{\kappa_{\mathbf{u}}(G)}}. \tag{2.38}$$

2.5.3 Computation Cost

To sample tree uniformly at random from $\mathcal{T}(G)$, [32] recently proposed a method that has $O(|E|^{1+o(1)})$ runtime using short-cutting method and insights from effective resistance. The earliest polynomial time algorithm has been known since [38]. While we do not recall either of these here, we briefly recall algorithm from [39] due to its elegance even though it is not the optimal (it has $O(N|E|)$ run time): (1) starting with any $u \in V$ run a random walk on G until it covers all vertices, (2) for every vertex $v \neq u$, select the edge through which v was reached for the first time during the walk, and (3) output the $N-1$ edges (which form tree) thus selected.

Given n such samples, to compute $\widehat{\Phi}_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})$, we have to compute $2n$ log-partition functions for tree structured graph. As noted in Section 2.2, each such computation requires $O(N|\mathcal{X}|^2)$ operations.

By Lemma 2.5.1, we need $n \geq O\left((d+\log(N))\kappa_{\mathbf{u}}(G)^{-2}\epsilon^{-2}\right)$ to achieve $(1+\epsilon)/\sqrt{\kappa_{\mathbf{u}}(G)}$ approximation with probability $1-e^{-d}$. That is, in total we need total of $O(|E|^{1+o(1)} + N|\mathcal{X}|^2) \times O(\kappa_{\mathbf{u}}(G)^{-2}\epsilon^{-2}\log 1/\epsilon)$ computation for $(1+\epsilon)/\sqrt{\kappa_{\mathbf{u}}(G)}$ approximation with probability $1-\epsilon$.

2.5.4 $\kappa_{\mathbf{u}}(G)$ and Effective Resistance

The $\kappa_{\mathbf{u}}(G) = \min_{e \in E} u_e$ where $u_e = \mathbb{E}_{\mathsf{T} \sim \mathbf{u}}[\mathbf{1}(e \in \mathsf{T})]$ turns out to be related to the so called "effective resistance" associated with edge $e \in E$ for the graph G. The notion was introduced by [40] and has multiple interpretations. We present one such here. For $e = (s,t) \in E$, the effective resistance u_e is equal to the amount of electric energy dissipated by the network when all edges are seen as electric wire of resistance $R_e = 1$ and a generator guarantees a total current flow ($\iota_{\text{gen}} = 1$) from s to t. The distribution of the current ι across the network must minimize the dissipated energy while respecting the constraints imposed by Kirchoff's laws (also see [41, Chapter 2]). Below we provide variational characterization of it.

$$\forall e = (s,t) \in E: \quad u_e = \min \left\{ \sum_{\{u,v\} \in E} \iota(u,v)^2 \;\middle|\; \begin{array}{c} \forall \{u,v\} \in E : \iota(u,v) + \iota(v,u) = 0 \\ \forall u \in V \setminus \{s,t\} : \sum_{v|(u,v)\in E} \iota(u,v) = 0 \\ \sum_{v|(s,v)\in E} \iota(s,v) = \sum_{u|(u,t)\in E} \iota(u,t) = 1 \end{array} \right\}. \tag{2.39}$$

Lemma 2.5.2. *Given $G = (V,E)$: (a) if d be the maximum vertex degree, then for any $e \in E$, $u_e \geq \frac{2}{d+1}$; (b) if the girth is at least $g > 3$, then for any $e \in E$, $u_e \geq \frac{1}{1+\frac{|E|}{(g-1)^2}}$.*

The proof can be found in Appendix A.6.

2.6 Beyond Trees

This far, we have restricted to approximating $\Phi(\boldsymbol{\theta})$ by decomposing $\boldsymbol{\theta} = \mathbb{E}_{\mathsf{T} \sim \boldsymbol{\rho}}[\Pi^T_{\boldsymbol{\rho}}(\boldsymbol{\theta})]$ and then using convexity, monotonicity and sub-linearity to produce an approximation guarantee. Such arguments would hold even if we can decompose $\boldsymbol{\theta}$ using subgraphs of G beyond trees. The choice of trees was particularly useful since they allow for an efficient computation of Φ. In general, graphs with bounded tree-width lend themselves to efficient computation of Φ, cf. [29].

To that end, let $\mathcal{T}_k(G)$ denote the set of all subgraphs of G that have treewidth bounded by $k \geq 1$. Let $\mathcal{P}(\mathcal{T}_k(G))$ denote the distribution over all such subgraphs. For any $H \in \mathcal{T}_k(G)$ and $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T}_k(G))$, define $\Pi^H(\cdot)$ and $\Pi^H_{\boldsymbol{\rho}}(\cdot)$ similar to that in (2.14) in Section 2.3 in the definition of TRW′,

$$L_{\boldsymbol{\rho}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathsf{H} \sim \boldsymbol{\rho}}(\Phi(\Pi^{\mathsf{H}}(\boldsymbol{\theta}))), \quad U_{\boldsymbol{\rho}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathsf{H} \sim \boldsymbol{\rho}}(\Phi(\Pi^{\mathsf{H}}_{\boldsymbol{\rho}}(\boldsymbol{\theta}))), \quad \text{and} \quad \widehat{\Phi}_{\boldsymbol{\rho}}(\boldsymbol{\theta}) = \sqrt{L_{\boldsymbol{\rho}}(\boldsymbol{\theta}) U_{\boldsymbol{\rho}}(\boldsymbol{\theta})}. \tag{2.40}$$

Using identical arguments as in Theorem 2.1.1, it follows that $\widehat{\Phi}_{\boldsymbol{\rho}}(\boldsymbol{\theta})$ is $\frac{1}{\sqrt{\kappa^k_{\boldsymbol{\rho}}}}$-approximation

where

$$\kappa_{\boldsymbol{\rho}}^{k} = \max_{\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T}_k(G))} \min_{e \in E} \rho_e. \tag{2.41}$$

(ϵ, k)-*partitioning*. While such generality is pleasing its utility is in improved approximation. Indeed, in [30, 31] a seemingly different approach was proposed using graph partitioning. At its core, it was shown that for a large family of graphs including minor-excluded graphs and graphs with polynomial growth, there exists $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T}_k(G))$ which satisfies certain (ϵ, k)-partitioning property (for appropriately chosen ϵ, k). Consider k-partitions of G defined as

$$\mathsf{Part}_k(G) = \{H = (V, \bigcup_{i=1}^{K} E(S_i)) \mid (S_i)_{1 \le i \le k} \text{ is a partition of } V \text{ and } \forall i : |S_i| \le k\}. \tag{2.42}$$

Note that $\mathsf{Part}_k(G) \subset \mathcal{T}_k(G)$. A distribution $\boldsymbol{\rho} \in \mathcal{P}(\mathsf{Part}_k(G)) \subset \mathcal{P}(\mathcal{T}_k(G))$ is called an (ϵ, k)-partitioning of G if

$$\forall e \in E : 1 - \epsilon \le \mathbb{E}_{\mathsf{H} \sim \boldsymbol{\rho}}[\mathbf{1}(e \in \mathsf{H})] \le 1. \tag{2.43}$$

We state the following result whose proof can be found in Appendix A.5.

Theorem 2.6.1. *Let G be such that there exists $\boldsymbol{\rho} \in \mathcal{P}(\mathsf{Part}_k(G)) \subset \mathcal{P}(\mathcal{T}_k(G))$ that is (ϵ, k) partition of G. Then, for any $\boldsymbol{\theta} \in \mathbb{R}_+^{|E|}$*

$$\sqrt{1-\epsilon} \le \frac{\Phi(\boldsymbol{\theta})}{\widehat{\Phi}_{\rho}(\boldsymbol{\theta})} \le \frac{1}{\sqrt{1-\epsilon}}. \tag{2.44}$$

We note that $\frac{1}{\sqrt{1-\epsilon}} = 1 + \frac{1}{2}\epsilon + o(\epsilon)$ and hence it improves upon the result given in [30, 31] which achieves a $1 + \epsilon$ approximation error.

2.7 Conclusions

We presented a method to quantify the approximation ratio of variational approximation method for est mating the log-partition function of discrete pairwise graphical models. As the main contribution, we quantified the approximation error as a function of the underlying graph properties. In particular, for a variant of the tree-reweighted algorithm, for graphs with bounded degree the approximation ratio is a constant factor (function of degree) and graphs with large ($\gg$ logarithmic) girth, the approximation ratio is close to 1. The method naturally extends beyond trees unifying prior works on graph partitioning based approach.

In this work, we restricted the analysis to non-negative valued potentials and edge parameters. If potentials are bounded, we can transform the general setting into a setting with non-negative potentials. However, the approximation ratio with respect to this transformed

setting may not translate to that of the original setting. This may be interesting direction for future works.

Acknowledgements

This work is supported in parts by projects from NSF and KACST as well as by a Hewlett Packard graduate fellowship. We would like to thank Moïse Blanchard for useful discussions on duality.

Chapter 3

Conclusion and open questions

3.1 Log-potentials taking negative values

In our analysis, we have focused on the case where log-potentials ψ_e take positive values. While we argued in Section 1.3 that approximating the corresponding partition function Z was as hard as unique games, a lot of natural approximation problems cannot be expressed naturally with positive potentials.

For instance, *maximum independent set*, which aims at finding $x \in \{0,1\}^V$ maximizing $\sum_{i \in V} x_i$ under the constraints $\forall e = (i,j) \in E : (x_i, x_j) \neq (1,1)$ naturally corresponds to the following log-potentials:

$$\psi_i(x_i) = \beta \mathbb{1}(x_i = 1) \tag{3.1}$$

$$\psi_{(i,j)}(x_i, x_j), = -\beta \mathbb{1}((x_i, x_j) \neq (1,1)). \tag{3.2}$$

Any constant factor approximation on $Z(\beta)$ would then yield a constant factor approximation on the size of the maximum independent set. In particular a $\kappa(G) \geq \frac{2}{\overline{d}+1}$ approximation factor where $\overline{d}$ is the maximum average degree would be significant in the light of known hardness results [42].

One of the reasons why there is little hope of achieving similar approximation ratio when allowing for potentials with negative values is the feasibility of $Z = 1$ (i.e, $\log(Z) = 0$) which reduces any constant factor approximation to an exact computation. Instead, an option could be to allow for log-potentials taking values in $\mathbb{R}^+ \cup \{-\infty\}$ but again, it is easy to see that testing whether $Z = 0$ (i.e, $\log(Z) = -\infty$) would immediately allow to decide whether a constraint satisfaction problem on G (like 3-colorability) admits a solution.

This leads us to consider the case when log-potentials can only take bounded negative values (i.e. such that $\phi_e \geq -m$). One can then consider the re-scaled potentials $\phi'_e =$

$\phi_e + m \geq 0$ and approximate the corresponding log-partition function $\Phi' = \log(Z')$ with TRW′. This yields the following where for clarity, we drop the θ, and we denote $\kappa(G) = \kappa$:

$$\log(Z') = \log\left(\sum_{x \in \mathcal{X}^N} \Pi_{e \in E} \exp(\phi_e(x_e) + m)\right) = \log(Z) + m|E| \tag{3.3}$$

$$\Phi' = \Phi + m|E|. \tag{3.4}$$

The approximation $\widehat{\Phi}'$ obtained by TRW is provably a $\frac{1}{\sqrt{\kappa}}$-approximation of $\Phi' = \log(Z')$. It is then natural to define $\widehat{\Phi} = \widehat{\Phi}' - m|E|$. The guarantee we obtain on $\widehat{\Phi}$ writes as follows: $\sqrt{\kappa}\Phi + (\sqrt{\kappa} - 1)m|E| \leq \widehat{\Phi} \leq \frac{1}{\sqrt{\kappa}}\Phi + (\frac{1}{\sqrt{\kappa}} - 1)m|E|$ and is obviously not a constant-factor guarantee. Obtaining a constant factor approximation in any setup where there are negative weights would be a significant improvement.

3.2 Relation to graph sparsification

Note that the approximation factor $\kappa(G)$ reflects the local sparsity of the graph G.

$$\kappa_1(G) = \min_{S \subset V} \frac{|S| - 1}{|E(S)|} \tag{3.5}$$

For a graphical model $(G, (\phi_e)_{e \in E})$ with partition function $Z_{G,\phi}$ if there exists some sparser graphical model $(\tilde{G}, (\tilde{\phi}_e)_{e \in E})$ such that $Z_{\tilde{G},\tilde{\phi}} \approx Z_{G,\phi}$, then one can obtain a better approximation guarantee by approximating the sparse problem.

This is specifically the case for MAXCUT for which linear sized spectral sparsifiers [43, 44] produce a $(1 + \epsilon)$ sparsifier of G with $O(\frac{n}{\epsilon^2})$ edges. Unfortunately, this does not allow to beat the trivial factor $1/2$ approximation factor for MAXCUT but similar techniques could perhaps be deployed successfully for other problems.

3.3 Generalization to graphs with bounded tree-width

As we presented in Section 1.3.4, the general question we raised is *how to cover a graph G with its low tree-width subgraphs?* which translates in finding $\boldsymbol{\rho}$ such that

$$\boldsymbol{\rho} \in \operatorname*{arg\,max}_{\boldsymbol{\rho} \in \mathcal{T}_k(G)} \min_{e \in E} \rho_e \tag{3.6}$$

where $\mathcal{T}_k(G)$ denotes the edge-induced sub-graphs of G of tree-width less than k. Once this covering is known (and assuming $\boldsymbol{\rho}$ has polynomial support), we achieve an approximation

by the following multiplicative constant:

$$\kappa_k(G) = \max_{\boldsymbol{\rho} \in \mathcal{T}_k(G)} \min_{e \in E} \rho_e \tag{3.7}$$

Obtaining a closed form for $\kappa_k(G)$ for $k \geq 1$ is beyond the scope of this thesis and would immediately result in a better approximation factor for the log-partition function. This problem is likely to be much harder since the maximum weight partial k-subtree problem is known to be NP-hard [45, 46] when the correctness of Kruskal's algorithm was a key ingredient in our proof.

3.4 A practical algorithm to find $\boldsymbol{\rho}$

In Chapter 2 we defined two dual linear programming problems with solution $\kappa(G)$. We showed that these two problems could be solved in polynomial time by the ellipsoid method, and gave a closed form for the objective. However we observed that these problems are hard to solve efficiently in practice. In our simulations (**Primal** in Figure 1-2 and **Dual** in Figure A-1) we had to restrict the graph to have less than 10 nodes for computational purposes by lack of an existing efficient solver on the spanning tree polytope in Python (though it exists in theory). We wonder whether an effective and natural algorithm can be designed to solve either of these problems. We believe that this problem could be closely related to the computation of the graph density $f(G)$ [47, 48] that is equal to :

$$f(G) = \max_{S \subset V} \frac{|E(S)|}{|S|}, \tag{3.8}$$

and for which there are various efficient algorithmic formulations. We transcribe here the elegant LP formulation given by [48] stating that $f(G)$ is also the result of the following linear programming problem:

$$\max \left(\sum_{ij} x_{ij} \right) \tag{3.9}$$

$$\text{s.t.} \quad \forall ij \in E : x_{ij} \leq y_i \tag{3.10}$$

$$\forall ij \in E : x_{ij} \leq y_j \tag{3.11}$$

$$\sum_i y_i \leq 1 \tag{3.12}$$

$$x_{ij}, y_i \geq 0 \tag{3.13}$$

Note that contrarily to **Primal** or **Dual** this problem only has a polynomial number of variables and a polynomial number of constraints and can therefore be solved much more efficiently in practice (without the need for an oracle). This suggests an efficient LP for-

mulation of:

$$\frac{1}{\kappa(G)} = \max_{S \subset V} \frac{|E(S)|}{|S| - 1}. \tag{3.14}$$

Consider the following linear problem indexed by $i_0 \in V$:

$$\max \left(\sum_{ij} x_{ij} \right) \tag{3.15}$$

$$\text{s.t.} \quad \forall ij \in E : x_{ij} \leq y_i \tag{3.16}$$

$$\forall ij \in E : x_{ij} \leq y_j \tag{3.17}$$

$$y_{i_0} = 1 \tag{3.18}$$

$$\sum_{i \neq i_0} y_i \leq 1 \tag{3.19}$$

$$x_{ij}, y_i \geq 0 \tag{3.20}$$

By the same reasoning as that deployed in [48], this problem has solution:

$$\max_{S \subset V \setminus \{i_0\}} \frac{|E(S \cup \{i_0\})|}{|S|}. \tag{3.21}$$

By change of variable $S' = S \cup \{i_0\}$ and taking the max over all possible $i_0 \in V$ (we solve each of the n associated linear problems), we obtain

$$\frac{1}{\kappa(G)} = \max_{S' \subset V} \frac{|E(S')|}{|S'| - 1}. \tag{3.22}$$

This shows that $\kappa(G)$ can be expressed efficiently by a linear program with polynomial number of constraints and variables. There remains to derive an efficient algorithm for computing $\boldsymbol{\rho}^* \in \arg\max_{\boldsymbol{\rho} \in \mathcal{T}_1(G)} \min_{e \in E} \rho_e$ which also appears to be an interesting open problem.

Bibliography

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

[2] Mohsen Bayati, Devavrat Shah, and Mayank Sharma. Maximum weight matching via max-product belief propagation. In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, pages 1763–1767. IEEE, 2005.

[3] David Gamarnik, David A Goldberg, and Theophane Weber. Correlation decay in random decision networks. *Mathematics of Operations Research*, 39(2):229–261, 2014.

[4] Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of inference in graphical models. *arXiv preprint arXiv:1206.3240*, 2012.

[5] Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.

[6] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for np-hard problems restricted to partial k-trees. *Discrete applied mathematics*, 23(1):11–24, 1989.

[7] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2): 183–233, 1999.

[8] Algorithms for inference, 6.438, mit lecture notes.

[9] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[10] Nadia Creignou and Miki Hermann. *On P completeness of some counting problems*. PhD thesis, INRIA, 1993.

[11] Neil Robertson, Paul Seymour, and Robin Thomas. Quickly excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 62(2):323–348, 1994.

[12] Chandra Chekuri and Julia Chuzhoy. Polynomial bounds for the grid-minor theorem. *Journal of the ACM (JACM)*, 63(5):1–65, 2016.

[13] Subhash Khot and Nisheeth K Vishnoi. On the unique games conjecture. In *FOCS*, volume 5, page 3. Citeseer, 2005.

[14] Stats 375, inference in graphical models, stanford lecture notes.

[15] Romain Cosson and Devavrat Shah. Quantifying variational approximation for log-partition function. In *Conference on Learning Theory*, pages 1330–1357. PMLR, 2021.

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

[17] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM journal on computing*, 18(6):1149–1178, 1989.

[18] Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 140–149, 2006.

[19] Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

[20] Amir Dembo, Andrea Montanari, Nike Sun, et al. Factor models on locally tree-like graphs. *Annals of Probability*, 41(6):4162–4213, 2013.

[21] Hans-Otto Georgii. *Gibbs measures and phase transitions*, volume 9. Walter de Gruyter, 2011.

[22] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Generalized belief propagation. In *Advances in neural information processing systems*, pages 689–695, 2001.

[23] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.

[24] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.

[25] David Gamarnik and Dmitriy Katz. Correlation decay and deterministic fptas for counting colorings of a graph. *Journal of Discrete Algorithms*, 12:29–47, 2012.

[26] Mohsen Bayati, David Gamarnik, Dimitriy Katz, Chandra Nair, and Prasad Tetali. Simple deterministic approximation algorithms for counting matchings. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 122–127, 2007.

[27] David Gamarnik and Dmitriy Katz. Sequential cavity method for computing free energy and surface pressure. *Journal of Statistical Physics*, 137(2):205–232, 2009.

[28] Michael Chertkov and Vladimir Y Chernyak. Loop series for discrete statistical models on graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(06): P06009, 2006.

[29] Venkat Chandrasekaran, Misha Chertkov, David Gamarnik, Devavrat Shah, and Jinwoo Shin. Counting independent sets using the bethe approximation. *SIAM Journal on Discrete Mathematics*, 25(2):1012–1034, 2011.

[30] Kyomin Jung and Devavrat Shah. Local approximate inference algorithms. *arXiv preprint cs/0610111*, 2006.

[31] Kyomin Jung, Pushmeet Kohli, and Devavrat Shah. Local rules for global map: When do they work? In *NIPS*, pages 871–879, 2009.

[32] Aaron Schild. An almost-linear time algorithm for uniform random spanning tree generation. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 214–227, 2018.

[33] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical programming*, 1 (1):127–136, 1971.

[34] Lap Chi Lau, Ramamoorthi Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization*, volume 46. Cambridge University Press, 2011.

[35] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

[36] Michel X Goemans. Minimum bounded degree spanning trees. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 273–282. IEEE, 2006.

[37] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[38] Alain Guenoche. Random spanning tree. *Journal of Algorithms*, 4(3):214–220, 1983.

[39] Andrei Z Broder. Generating random spanning trees. In *FOCS*, volume 89, pages 442–447. Citeseer, 1989.

[40] Douglas J Klein and Milan Randić. Resistance distance. *Journal of mathematical chemistry*, 12(1):81–95, 1993.

[41] Russell Lyons and Yuval Peres. *Probability on trees and networks*, volume 42. Cambridge University Press, 2017.

[42] Per Austrin, Subhash Khot, and Muli Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 74–80. IEEE, 2009.

[43] Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

[44] Yin Tat Lee and He Sun. An sdp-based algorithm for linear-sized spectral sparsification. In *Proceedings of the 49th annual acm sigact symposium on theory of computing*, pages 678–687, 2017.

[45] Shahaf Dafna and Carlos Guestrin. Learning thin junction trees via graph cuts. In *Artificial Intelligence and Statistics*, pages 113–120. PMLR, 2009.

[46] David R Karger and Nathan Srebro. Learning markov networks: maximum bounded tree-width graphs. In *SODA*, pages 392–401, 2001.

[47] Andrew V Goldberg. *Finding a maximum density subgraph*. University of California Berkeley, 1984.

[48] Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 84–95. Springer, 2000.

[49] Noga Alon, Shlomo Hoory, and Nathan Linial. The moore bound for irregular graphs. *Graphs and Combinatorics*, 18(1):53–57, 2002.

Appendix A

Proofs and illustrations

A.1 Proof of Lemma 2.3.1

Proof. We start by observing a few properties of function $\Phi(\cdot)$.

Property 1. Φ *is non-decreasing.* For $a, b \in \mathbb{R}^n$ let $a \preceq b$ denote that every component of a is less or equal to that of b, i.e. $a_i \leq b_i,\ i \in [n]$. With this, for $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}_+^{|E|}$ such that $\boldsymbol{\theta} \preceq \boldsymbol{\theta}'$, it can be easily verified that

$$\Phi(\boldsymbol{\theta}) \leq \Phi(\boldsymbol{\theta}'). \tag{monotonicity}$$

Since $\Phi(\mathbf{0}) = N \log |\mathcal{X}|$, and $\mathbf{0} \preceq \boldsymbol{\theta} \preceq \boldsymbol{\theta}'$, we have

$$N \log(|\mathcal{X}|) \leq \Phi(\boldsymbol{\theta}) \ \leq \ \Phi(\boldsymbol{\theta}'). \tag{A.1}$$

Property 2. Φ *is sub-linear.* For $\lambda \geq 1$ and $\boldsymbol{\theta} \in \mathbb{R}_+^{|E|}$,

$$\Phi(\lambda \boldsymbol{\theta}) \leq \lambda \Phi(\boldsymbol{\theta}). \tag{sub-linearity}$$

The above follows from the fact that for any $\mathbf{s} = (s_i) \in \mathbb{R}_+^n$,

$$\big(\sum_{i=1}^{n} s_i^{\lambda}\big) \leq \big(\sum_{i=1}^{n} s_i\big)^{\lambda}.$$

Now consider any $\boldsymbol{\rho} \in \mathcal{P}(\mathcal{T}(G))$. For any $T \in \mathcal{T}(G)$ and $\boldsymbol{\theta} \in \mathbb{R}_+^{|E|}$, by definition of Π^T, we have that $\Pi^T(\boldsymbol{\theta}) \preceq \boldsymbol{\theta}$. Therefore, using the monotonicity of the log-partition function it follows that

$$L_{\boldsymbol{\rho}}(\boldsymbol{\theta}) = \sum_{T \in \mathcal{T}(G)} \rho^T \Phi(\Pi^T(\boldsymbol{\theta})) \leq \sum_{T \in \mathcal{T}(G)} \rho^T \Phi(\boldsymbol{\theta}) \leq \Phi(\boldsymbol{\theta}). \tag{A.2}$$

By definition $\boldsymbol{\theta} = \mathbb{E}_{\mathsf{T}\sim\boldsymbol{\rho}}[\Pi^{\mathsf{T}}_{\boldsymbol{\rho}}(\boldsymbol{\theta})]$, and due to convexity of Φ (cf. (2.11)), it follows that

$$\Phi(\boldsymbol{\theta}) = \Phi\big(\mathbb{E}_{\mathsf{T}\sim\boldsymbol{\rho}}[\Pi^{\mathsf{T}}_{\boldsymbol{\rho}}(\boldsymbol{\theta})]\big) \;\leq\; \mathbb{E}_{\mathsf{T}\sim\boldsymbol{\rho}}[\Phi(\Pi^{\mathsf{T}}_{\boldsymbol{\rho}}(\boldsymbol{\theta}))] \;=\; U_{\boldsymbol{\rho}}(\boldsymbol{\theta}). \tag{A.3}$$

By definition of $\kappa_{\boldsymbol{\rho}} = \min_{e\in E}\rho_e$, it follows that

$$\Pi^{T}_{\boldsymbol{\rho}}(\boldsymbol{\theta}) \leq \frac{1}{\kappa_{\boldsymbol{\rho}}}\Pi^{T}(\boldsymbol{\theta}), \;\; \forall\, T \in \mathcal{T}(G). \tag{A.4}$$

And, by definition $\kappa_{\boldsymbol{\rho}} \geq 1$. Therefore by (monotonicity) and (sub-linearity), we have

$$\Phi(\Pi^{T}_{\boldsymbol{\rho}}(\boldsymbol{\theta})) \leq \Phi\big(\frac{1}{\kappa_{\boldsymbol{\rho}}}\Pi^{T}(\boldsymbol{\theta})\big) \;\leq\; \frac{1}{\kappa_{\boldsymbol{\rho}}}\Phi(\Pi^{T}(\boldsymbol{\theta})). \tag{A.5}$$

Therefore,

$$U_{\boldsymbol{\rho}}(\boldsymbol{\theta}) = \sum_{T\in\mathcal{T}(G)} \rho^T \Phi(\Pi^{T}_{\boldsymbol{\rho}}(\boldsymbol{\theta})) \;\leq\; \frac{1}{\kappa_{\boldsymbol{\rho}}}\Big(\sum_{T\in\mathcal{T}(G)} \rho^T \Phi(\Pi^{T}(\boldsymbol{\theta}))\Big) \;=\; \frac{1}{\kappa_{\boldsymbol{\rho}}} L_{\boldsymbol{\rho}}(\boldsymbol{\theta}). \tag{A.6}$$

As a consequence of (A.2), (A.3) and (A.6) we obtain that

$$\Phi(\theta) \leq U_{\boldsymbol{\rho}}(\boldsymbol{\theta}) \leq \frac{1}{\kappa_{\boldsymbol{\rho}}}\Phi(\boldsymbol{\theta}) \quad \text{and} \quad \kappa_{\boldsymbol{\rho}}\Phi(\boldsymbol{\theta}) \leq L_{\boldsymbol{\rho}}(\boldsymbol{\theta}) \leq \Phi(\boldsymbol{\theta}). \tag{A.7}$$

From this, it follows that

$$\sqrt{\kappa_{\boldsymbol{\rho}}}\Phi(\boldsymbol{\theta}) \leq \sqrt{L_{\boldsymbol{\rho}}(\boldsymbol{\theta})U_{\boldsymbol{\rho}}(\boldsymbol{\theta})} \leq \frac{1}{\sqrt{\kappa_{\boldsymbol{\rho}}}}\Phi(\boldsymbol{\theta}). \tag{A.8}$$

Which can be rewritten as

$$\sqrt{\kappa_{\boldsymbol{\rho}}} \leq \frac{\widehat{\Phi}_{\boldsymbol{\rho}}(\boldsymbol{\theta})}{\Phi(\boldsymbol{\theta})} \leq \frac{1}{\sqrt{\kappa_{\boldsymbol{\rho}}}}. \tag{A.9}$$

By optimizing over choice of $\boldsymbol{\rho} = \boldsymbol{\rho}^\star$, we conclude that $\alpha(G, \text{TRW}') \leq \frac{1}{\kappa_{\boldsymbol{\rho}^\star}}$. □

A.2 Proof of Lemma 2.4.1

Proof. (See illustration in Figure A-1) For $\mathbf{w} = (w_e)_{e\in E}$ denote $f(\mathbf{w})$ the number of distinct values in its support:

$$f(\mathbf{w}) = |\{w_e : e \in E, w_e \neq 0\}|. \tag{A.10}$$

To prove the lemma, it suffices to show that there exists an optimal solution of **Dual** such that $f(\mathbf{w}) = 1$. We will prove that if $\mathbf{w}$ is an optimal solution and $f(\mathbf{w}) > 1$ then we

can build $\mathbf{w}'$ of similar objective value such that $f(\mathbf{w}') \leq f(\mathbf{w}) - 1$. By repeating this till $f(\mathbf{w}) = 1$ will conclude the proof.

Let $\mathbf{w}$ be an optimal solution with $f(\mathbf{w}) > 1$. We consider the edges $e_1, e_2, ..., e_{|E|}$ ordered by their weights, i.e.

$$w_{e_1} \geq ... \geq w_{e_{|E|}}. \tag{A.11}$$

In what follows, we will make sure that the ordering on the edges never changes, therefore we allow ourselves to write w_i instead of w_{e_i}. Now the objective of **Dual** achieved by such an optimal $\mathbf{w}$ corresponds to the weight of a maximum weight spanning tree. Let us utilize Kruskal's algorithm to find such an maximum weight spanning tree. Recall that Kruskal's algorithm greedily selects edges from higher to lower weight as long as they do not create a cycle with previously selected edges. We will denote $I_T = \{t_1 < ... < t_{N-1}\}$ the indices of the edges selected by the algorithm to construct tree T and let $I_{E\setminus T} = \cup_{k=1}^{N-1}\{s : t_k < s < t_{k+1}\}$ denote the indices of edges not part of T with notation $t_N = |E| + 1$. The weight of the maximum spanning tree is then $\mathbf{w}(T) = \sum_{k=1}^{N-1} w_{t_k}$. Note that $t_1 = 1$ and $t_2 = 2$ since cycle requires 3 or more edges. By definition $w_{j-1} \geq w_j$ for $2 \leq j \leq |E|$. Now if $w_{j-1} > w_j$ the we claim that $j \in I_T$. This is because for $1 \leq k \leq N-1$ if $(w_{t_k}, ..., w_{t_{k+1}-1})$ are not equal, setting them all to their average decreases w_{t_k} strictly while preserving $\mathbf{w} \in \mathcal{P}(E)$ as well as the order on the edges and therefore contradicting the optimality of $\mathbf{w}$ for **Dual**. Therefore $\mathbf{w}$ is piece-wise constant with discontinuities only appearing for $j \in I_T$.

If $f(\mathbf{w}) = 2$ and all weights are positive, we denote $2 \leq k \leq N-1$ such that $w_{t_k-1} > w_{t_k} > 0$ and we have:

$$w_1 = ... = w_{t_k-1} > w_{t_k} = ... = w_{|E|}. \tag{A.12}$$

In this case, the optimal objective value for **Dual** is equal to $(k-1)w_1 + (N-k)w_{t_k}$. To make $\mathbf{w}$ constant on its support while preserving the order on the weights, there are two possibilities. Either transfer all weight from $(w_{t_k}, ..., w_{|E|})$ to $(w_1, ..., w_{t_k-1})$ until $(w_{t_k}, ..., w_{|E|})$ reaches zero. The objective will then be $w_1 + \frac{|E|-t_k+1}{t_k-1} w_{t_k}$. Or transfer all weight from $(w_1, ..., w_{t_k-1})$ to $(w_{t_k}, ..., w_{|E|})$ until all weights are equal. The objective will be then $w_{t_k} + \frac{|E|-t_k+1}{t_k-1}(w_1 - w_{t_k})$. Because either $\frac{|E|-t_k+1}{t_k-1} \leq 1$ or $\frac{t_k-1}{|E|-t_k+1} \leq 1$, one of these transfers does not increase the objective and yields $f(\mathbf{w}) = 1 < 2$.

If $f(\mathbf{w}) = 2$ and some weights are 0, denote k_0 the smallest index such that $w_{t_{k_0}} = 0$. The method above still holds when replacing $|E| - t_k + 1$ by $t_{k_0} - t_k$.

Now suppose $f(\mathbf{w}) \geq 3$, making sure that the order on the weights is preserved requires extra caution. In addition to k and k_0 (if required), we denote k_1 the index of the

discontinuity that follows k. We have:

$$... = w_{t_{k_1}-1} > w_{t_{k_1}} = ... = w_{t_k-1} > w_{t_k} = ... = w_{t_{k_0}-1} > w_{t_{k_0}} = ... \tag{A.13}$$

In the event when we want to transfer weight from $(w_{t_k}, ..., w_{t_{k_0}-1})$ to $(w_{t_{k_1}}, ..., w_{t_k-1})$, we must make sure that $(w_{t_{k_1}}, ..., w_{t_k-1})$ does not exceed $w_{t_{k_1}-1}$. If $(w_{t_{k_1}}, ..., w_{t_k-1})$ attains $w_{t_{k_1}-1}$ the transfer must stop at equality, and one should observe that we have decreased $f(\mathbf{w})$ strictly by 1 because the discontinuity at $w_{t_{k_1}}$ has disappeared and no new discontinuity was created.

In summary, we have argued that if $\mathbf{w}$ is an optimal solution and $f(\mathbf{w}) > 1$ then we can build $\mathbf{w}'$ of same objective value (optimal) and such that $f(\mathbf{w}') \leq f(\mathbf{w}) - 1$. This completes the proof of Lemma. □

A.3 Proof of Lemma 2.4.2

Proof. We prove the equality by establishing inequalities in both direction.

Establishing $\min_{S\subset V} \frac{|S|-1}{|E(S)|} \geq \min_{F\subset E} \frac{|V(F)|-c(F)}{|F|}$*:* For $S \subset V$ note that $V(E(S)) \subset S$ and $c(E(S)) \geq 1$ and therefore that $\frac{|S|-1}{|E(S)|} \geq \frac{|V(E(S))|-c(E(S))}{|E(S)|}$ with $E(S) \subset E$. Thus, $\min_{S\subset V} \frac{|S|-1}{|E(S)|}$ is minimizing a larger objective function over smaller set compared to $\min_{F\subset E} \frac{|V(F)|-c(F)}{|F|}$. Therefore, inequality follows immediately.

Establishing $\min_{S\subset V} \frac{|S|-1}{|E(S)|} \leq \min_{F\subset E} \frac{|V(F)|-c(F)}{|F|}$*:* Let $F^\star \subset E$ be a minimizer of $\min_{F\subset E} \frac{|V(F)|-c(F)}{|F|}$. Let $H = (V(F^\star), F^\star)$. By optimality, all connected components of H must be vertex-induced subgraphs of G. This is because, if not then it is possible to add edges to H without changing the number of vertices or number of connected components in it, which would contradict optimality. In other words, there exists disjoint subsets $S_i, 1 \leq i \leq c(H)$ of $V(F^\star)$ with $V(F^\star) = \cup_{i=1}^{c(H)} S_i$ and $F^\star = \cup_{i=1}^{c(H)} E(S_i)$. If $c(H) = 1$, then the inequality follows immediately. If $c(H) \geq 2$, denote $H \setminus H_1$ the graph obtained by removing $H_1 = (S_1, E(S_1))$ from H. Note that $c(H \setminus H_1) = c(H) - 1$ and that $c(H_1) = 1$. By Lemma A.3.1, $\forall a, b, c, d \in \mathbb{R}^4_+ : \min(\frac{a}{b}, \frac{c}{d}) \leq \frac{a+c}{b+d}$. Therefore,

$$\min\left(\frac{|V(H_1)| - c(H_1)}{|E(H_1)|}, \frac{|V(H \setminus H_1)| - c(H \setminus H_1)}{|E(H \setminus H_1)|}\right) \leq \frac{|V(H)| - c(H)}{|E(H)|}. \tag{A.14}$$

If H_1 achieves the minimum on the left hand side, then it concludes the proof. If $H \setminus H_1$ achieves the minimum simply iterate the above argument till we are left with single connected component and that would conclude the proof. □

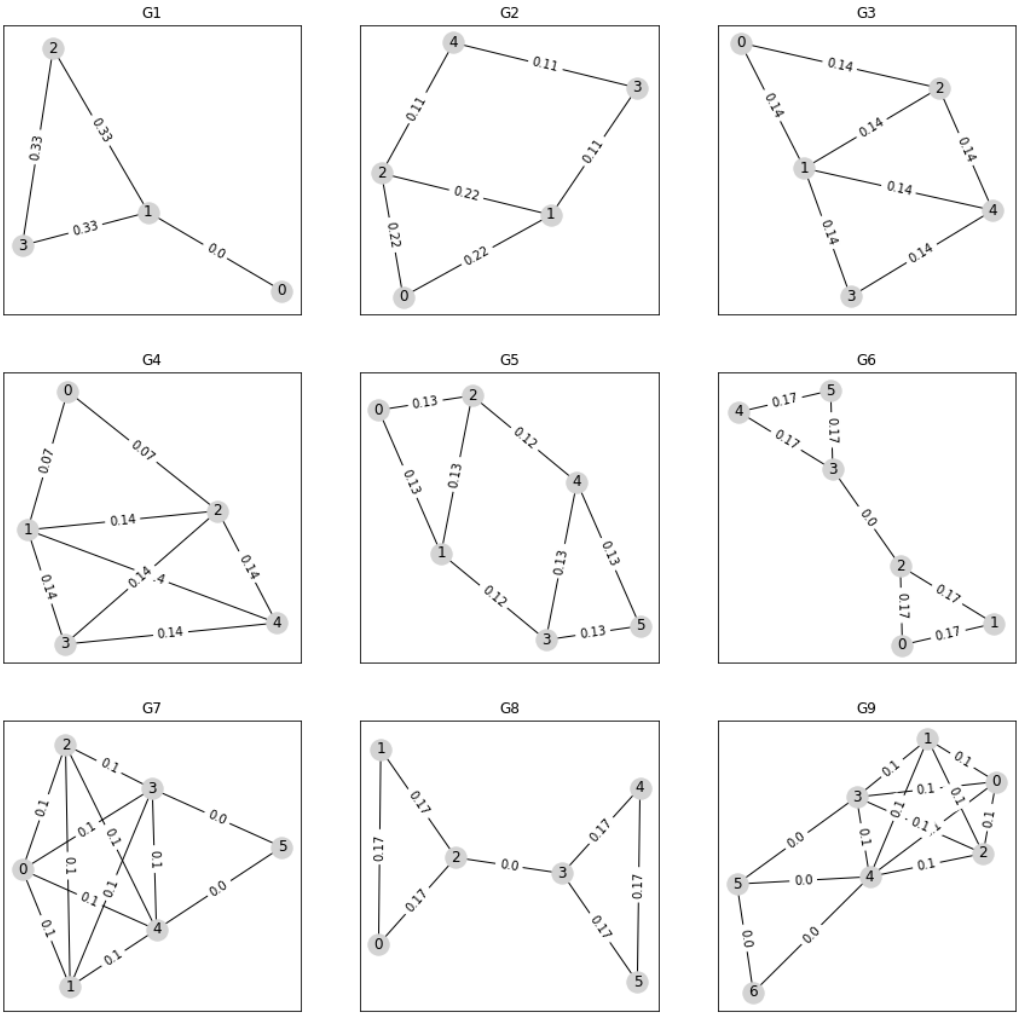

Figure A-1: An example of an optimal weight assignment for the problem **Dual** on nine different graphs. The solution was found by the interior point method using a linear programming solver. Note that for most graphs, the solution reached is already constant on its support. On graphs G_2 and G_4, note that evening out the weights would not increase the weight of the maximum spanning tree.

Lemma A.3.1. *For any* $a, b, c, d \in \mathbb{R}_+$,

$$\min(\frac{a}{b}, \frac{c}{d}) \leq \frac{a+c}{b+d}. \tag{A.15}$$

Proof. Let $a, b, c, d \in \mathbb{R}_+$. Without loss of generality assume that Then the following sequence of statements hold leading to the proof of the claim:

$$ad \leq bc \quad \text{or} \quad bc \leq ad \tag{A.16}$$

$$\min(ad(b+d), cb(b+d)) \leq (a+c)bd \tag{A.17}$$

$$\min(\frac{a}{b}, \frac{c}{d}) \leq \frac{a+c}{b+d}. \tag{A.18}$$

□

A.4 Proofs of Lemmas 2.4.3 and 2.4.4

Proof of Lemma 2.4.3. Assume G has maximum average degree bounded by $\bar{d}$, where by definition

$$\bar{d} = \max_{S \subset V} \frac{2|E(S)|}{|S|}. \tag{A.19}$$

Therefore, for any $S \subset V$, $|E(S)| \leq \frac{\bar{d}}{2}|S|$. And there can be at most $\binom{|S|}{2}$ edges in a graph over vertices S, and hence $|E(S)| \leq \frac{|S|(|S|-1)}{2}$. Therefore, we obtain

$$\frac{|S|-1}{|E(S)|} \geq \frac{2}{\bar{d}}\big(1 - \frac{1}{|S|}\big) = L_1(|S|), \tag{A.20}$$

$$\frac{|S|-1}{|E(S)|} \geq \frac{2}{|S|} = L_2(|S|). \tag{A.21}$$

Therefore

$$\frac{|S|-1}{|E(S)|} \geq \min_{x \in \mathbb{R}_+} \{\max(L_1(x), L_2(x))\}. \tag{A.22}$$

Note that L_1 is increasing and bounded whereas L_2 is decreasing. Therefore, $\max(L_1(x), L_2(x))$ with $x \in \mathbb{R}$ reaches its minimum for x such that $L_1(x) = L_2(x)$ which leads to minima at $x = \bar{d} + 1$. Therefore, we conclude that for all $S \subset V$,

$$\frac{|S|-1}{|E(S)|} \geq \frac{2}{\bar{d}+1}. \tag{A.23}$$

Proof of Lemma 2.4.4. Let G has girth $g > 3$. Therefore, all subgraphs of G have girth at

least g. The generalised Moore bound (obtained by [49]) then gives $\forall S \subset V$:

$$|S| \geq 1 + d_S \sum_{i=0}^{\frac{g-3}{2}} (d_S - 1)^i \qquad \text{if g is odd,} \tag{A.24}$$

$$|S| \geq 2 \sum_{i=0}^{\frac{g-2}{2}} (d_S - 1)^i, \qquad \text{if g is even} \tag{A.25}$$

with $d_S = \frac{2|E(S)|}{|S|}$. We will only keep a weaker version of this bound that does not depend on the parity of g. Specifically, for all $S \subset V$:

$$|S| \geq \big(2\frac{|E(S)|}{|S|} - 1\big)^{\frac{g-3}{2}}. \tag{A.26}$$

Therefore, $|E(S)| \leq \frac{1}{2}(|S|^{\frac{2}{g-3}+1} + |S|)$ for all $S \subset V$. Subsequently, we have

$$\frac{|S|-1}{|E(S)|} \geq 2\frac{1 - \frac{1}{|S|}}{1 + |S|^{\frac{2}{g-3}}} \geq 2\frac{1 - \frac{1}{|S|}}{1 + N^{\frac{2}{g-3}}} \tag{A.27}$$

This bound is clearly increasing with $|S|$. Also note that if $|S| \leq g - 1$, the subgraph $(S, E(S))$ can have no cycle and therefore $\frac{|S|-1}{|E(S)|} = 1$. The worse case is therefore attained for $|S| = g$ where we have:

$$\frac{|S|-1}{|E(S)|} \geq \frac{2}{1 + N^{\frac{2}{g-3}}}\big(1 - \frac{1}{g}\big). \tag{A.28}$$

A.5 Proof of Lemma 2.5.1

Proof. We shall use Hoeffding's inequality: for any bounded random variable $a \leq X \leq b$, the deviation of its n-empirical average $\overline{X}_n$ computed from in dependant samples is such that for any $t > 0$,

$$\mathbb{P}(|\mathbb{E}(X) - \overline{X}_n| \geq t) \leq 2\exp\Big(\frac{-2nt^2}{(b-a)^2}\Big). \tag{A.29}$$

Another version of the equation when $\mathbb{E}(X) > 0$ is as follows, for any $\epsilon > 0$

$$\mathbb{P}\left(1 - \epsilon \leq \frac{\overline{X}_n}{\mathbb{E}(X)} \leq 1 + \epsilon\right) \geq 1 - 2\exp\Big(\frac{-2n\epsilon^2\mathbb{E}(X)^2}{(b-a)^2}\Big). \tag{A.30}$$

An immediate consequence is that $\hat{\mathbf{u}}^n$ is a good approximation for $\mathbf{u}$. For any $e \in E$,

$$\mathbb{P}\left(1 - \epsilon \leq \frac{\hat{u}_e^n}{u_e} \leq 1 + \epsilon\right) \geq 1 - 2\exp\big(-2n\epsilon^2 u_e^2\big), \tag{A.31}$$

fctherefore by union bound,

$$\mathbb{P}\left(\forall e \in E : 1-\epsilon \leq \frac{\hat{u}_e^n}{u_e} \leq 1+\epsilon\right) \geq 1 - 2|E| \exp\left(-2n\epsilon^2\kappa_{\mathbf{u}}\right). \tag{A.32}$$

Another consequence is that $L_{\hat{\mathbf{u}}^n}$ is a good approximation for $L_{\mathbf{u}}$. Indeed, considering the random variable $\Phi(\Pi^{\mathsf{T}}(\boldsymbol{\theta}))$ of mean $L_{\mathbf{u}}(\boldsymbol{\theta})$ and of empirical average $L_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \Phi(\Pi^{\mathsf{T}_i}(\boldsymbol{\theta}))$ and noting that this variable is bounded as follows $0 \leq \Phi(\Pi^{\mathsf{T}}(\boldsymbol{\theta}))\,(\leq \Phi(\boldsymbol{\theta})) \leq \frac{1}{\kappa_{\mathbf{u}}} L_{\mathbf{u}}(\boldsymbol{\theta})$, we have

$$\mathbb{P}\big(1-\epsilon \leq \frac{L_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})}{L_{\mathbf{u}}(\boldsymbol{\theta})} \leq 1+\epsilon\big) \geq 1 - 2\exp\left(-2n\epsilon^2\kappa_{\mathbf{u}}^2\right). \tag{A.33}$$

Regarding $U_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})$, the discussion requires an additional argument because $\frac{1}{n}\sum_{i=1}^n \Phi(\Pi^{\mathsf{T}_i}_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta}))$ is not a sum of independent random variables. Instead, let us focus on the close quantity, $\frac{1}{n}\sum_{i=1}^n \Phi(\Pi^{\mathsf{T}_i}_{\mathbf{u}}(\boldsymbol{\theta}))$ for which we have $0 \leq \Phi(\Pi^{\mathsf{T}}_{\mathbf{u}}(\boldsymbol{\theta}))\left(\leq \frac{1}{\kappa_{\mathbf{u}}}\Phi(\boldsymbol{\theta})\right) \leq \frac{1}{\kappa_{\mathbf{u}}} U_{\mathbf{u}}(\boldsymbol{\theta})$ and therefore,

$$\mathbb{P}\big(1-\epsilon \leq \frac{\frac{1}{n}\sum_{i=1}^n \Phi(\Pi^{\mathsf{T}_i}_{\mathbf{u}}(\boldsymbol{\theta}))}{U_{\mathbf{u}}(\boldsymbol{\theta})} \leq 1+\epsilon\big) \geq 1 - 2\exp\left(-2n\epsilon^2\kappa_{\mathbf{u}}^2\right). \tag{A.34}$$

Fortunately, if (A.32) is satisfied this quantity turns out to be a good approximation of $U_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})$. Indeed, assuming that $\forall e \in E : 1-\epsilon \leq \frac{\hat{u}_e^n}{u_e} \leq 1+\epsilon$ we have that for all $T \in \mathcal{T}(G)$,

$$(1-\epsilon)\Pi^T_{\mathbf{u}}(\boldsymbol{\theta}) \preceq \Pi^T_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta}) \preceq (1+\epsilon)\Pi^T_{\mathbf{u}}(\boldsymbol{\theta}) \tag{A.35}$$

therefore by (monotonicity) and (sub-linearity),

$$(1-\epsilon)\Phi(\Pi^T_{\mathbf{u}}(\boldsymbol{\theta})) \leq \Phi(\Pi^T_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})) \leq (1+\epsilon)\Phi(\Pi^T_{\mathbf{u}}(\boldsymbol{\theta})), \tag{A.36}$$

which shows,

$$(1-\epsilon) \leq \frac{U_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})}{\frac{1}{n}\sum_{i=1}^n \Phi(\Pi^{\mathsf{T}_i}_{\mathbf{u}}(\boldsymbol{\theta}))} \leq (1+\epsilon). \tag{A.37}$$

Therefore by union bound,

$$\mathbb{P}\left((1-\epsilon)^2 \leq \frac{U_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})}{U_{\mathbf{u}}(\boldsymbol{\theta})} \leq (1+\epsilon)^2\right) \geq 1 - (2|E|+2)\exp\left(-2n\kappa_{\mathbf{u}}^2\epsilon^2\right). \tag{A.38}$$

By putting together (A.33) and (A.38), we obtain

$$\mathbb{P}\left((1-\epsilon)^{\frac{3}{2}} \leq \frac{\widehat{\Phi}_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})}{\Phi_{\mathbf{u}}(\boldsymbol{\theta})} \leq (1+\epsilon)^{\frac{3}{2}}\right) \geq 1 - (2|E|+4)\exp\left(-2n\kappa_{\mathbf{u}}^2\epsilon^2\right), \tag{A.39}$$

and by arguments of Lemma 2.3.1, we can conclude that

$$\mathbb{P}\left(\sqrt{\kappa_{\mathbf{u}}}(1-\epsilon)^{\frac{3}{2}} \leq \frac{\widehat{\Phi}_{\hat{\mathbf{u}}^n}(\boldsymbol{\theta})}{\Phi(\boldsymbol{\theta})} \leq \frac{(1+\epsilon)^{\frac{3}{2}}}{\sqrt{\kappa_{\mathbf{u}}}}\right) \geq 1-(2|E|+4)\exp\left(-2n\sqrt{\kappa_{\mathbf{u}}}^2\epsilon^2\right), \quad \text{(A.40)}$$

This completes the proof of Lemma 2.5.1. □

A.6 Proof of Lemma 2.5.2

Bounded degree graph G. First assume that G has maximum degree d. Consider any edge $e = (s,t) \in E$. Denote $\mathcal{N}(s), \mathcal{N}(t) \subset V$ the neighbours of s and t. Consider current $\iota : V \times V \to \mathbb{R}$ which is a solution of optimization problem corresponding to effective resistance as defined in (2.39). By definition, we have that the effective resistance u_e for $e \in E$ is given by

$$\begin{aligned} u_e &= \sum_{(u,v)\in E} \iota(u,v)^2 \\ &\geq \iota(s,t)^2 + \sum_{u\in\mathcal{N}(s)\setminus\{t\}} \iota(s,u)^2 + \sum_{u\in\mathcal{N}(t)\setminus\{s\}} \iota(u,t)^2. \end{aligned} \quad \text{(A.41)}$$

By constraints of the optimization problem, the sum of currents entering source s and leaving sink t is equal to 1 (whereas it is null for isolated vertices). Therefore, focusing on s, we have $\sum_{u\in\mathcal{N}(s)\setminus\{t\}} |\iota(s,u)| \geq 1 - |\iota(s,t)|$. By applying Cauchy Schwarz inequality, we have that

$$\Big(\sum_{u\in\mathcal{N}(s)\setminus\{t\}} \iota(s,u)^2\Big) \times \Big(\sum_{u\in\mathcal{N}(s)\setminus\{t\}} 1^2\Big) \geq (1-|\iota(s,t)|)^2. \quad \text{(A.42)}$$

Recall that G has maximum vertex degree d and therefore $|\mathcal{N}(s)\setminus\{t\}| \leq d-1$. Therefore,

$$\sum_{u\in\mathcal{N}(s)\setminus\{t\}} \iota(s,u)^2 \geq \frac{(1-|\iota(s,t)|)^2}{d-1}. \quad \text{(A.43)}$$

Because the same holds for the term $\sum_{u\in\mathcal{N}(t)\setminus(s)} \iota(u,t)^2$, we obtain from (A.41) that

$$u_e \geq \iota(s,t)^2 + (1-|\iota(s,t)|)^2\frac{2}{d-1}. \quad \text{(A.44)}$$

This expression holds for all possible values of $\iota(s,t)$. We note that for any given $\lambda \in \mathbb{R}_+$,

$$\inf_{x\in\mathbb{R}} x^2 + (1-x)^2\lambda \geq \frac{\lambda}{1+\lambda}. \quad \text{(A.45)}$$

Therefore, we conclude that for graph G with bounded degree d,

$$u_e \geq \frac{2}{d+1}. \tag{A.46}$$

Graph G with girth g. We now assume that G has girth g. As before, let $e = (s,t) \in E$. Denote $G \setminus \{e\} = (V, E \setminus \{e\})$ the graph obtained by removing edge e from G. For $0 \leq k \leq g-2$, we define

$$E_k = \{(u,v) \in E : d_{G\setminus\{e\}}(s,u) = k, d_{G\setminus\{e\}}(s,v) = k+1\}, \tag{A.47}$$

where $d_{G\setminus\{e\}}(s,u)$ denotes the shortest path distance between vertices s,u in graph G excluding edge e. That is, E_k is the set of edges connecting vertices at distance k from s in $G \setminus \{e\}$ to vertices at distance $k+1$ from s in $G \setminus \{e\}$. Since $k \leq g-2$, all E_k are disjoint and hence current ι satisfies

$$u_e \geq \iota(s,t)^2 + \sum_{k=0}^{g-2} \sum_{(u,v)\in E_k} \iota(u,v)^2. \tag{A.48}$$

For $0 \leq k \leq g-2$, note that $E_k \cup \{e\}$ defines a cut of G. Therefore by Kirchoff's law $\sum_{(u,v)\in E_k} |\iota(u,v)| \geq 1 - |\iota(s,t)|$. Using Cauchy-Schwartz inequality, we obtain:

$$\Big(\sum_{(u,v)\in E_k} \iota(u,v)^2\Big) \times \Big(\sum_{(u,v)\in E_k} 1^2\Big) \geq (1-\iota(s,t))^2. \tag{A.49}$$

By summing-up all inequalities, we obtain

$$\Big(\sum_{k=0}^{g-2} \sum_{(u,v)\in E_k} \iota(u,v)^2\Big) \geq (1-|\iota(s,t)|)^2 \Big(\sum_{k=0}^{g-2} \frac{1}{|E_k|}\Big). \tag{A.50}$$

Note that if a sequence $(m_k) \geq 0$ respects $\sum_{k=1}^{l} m_k \leq |E|$ then, $\sum_{k=1}^{l} \frac{1}{m_k} \geq \frac{l^2}{|E|}$. Therefore, because all E_k are disjoint, $\sum_{k=0}^{g-2} \frac{1}{|E_k|} \geq \frac{(g-1)^2}{|E|}$. Inserting this in (A.48), we obtain

$$u_e \geq \iota(s,t)^2 + (1-\iota(s,t))^2 \frac{(g-1)^2}{|E|}. \tag{A.51}$$

Using (A.45), we obtain

$$u_e \geq \frac{1}{1 + \frac{|E|}{(g-1)^2}}. \tag{A.52}$$

This completes the proof of Lemma 2.5.2.

A.7 Proof of Theorem 2.6.1

Proof. The proof follows by establishing that $\kappa^k_{\boldsymbol{\rho}}$ as defined in (2.41) for $\boldsymbol{\rho} \in \mathcal{P}(\mathsf{Part}_k(G))$ is such that

$$\kappa^k_{\boldsymbol{\rho}} \geq 1 - \epsilon, \tag{A.53}$$

if $\boldsymbol{\rho}$ is (ϵ, k) partition. Indeed, by definition of (ϵ, k) partition, we have that for any $e \in E$,

$$\rho_e = \mathbb{E}_{\mathsf{H} \sim \boldsymbol{\rho}}[\mathbf{1}(e \in \mathsf{H})] \geq 1 - \epsilon. \tag{A.54}$$

Therefore,

$$\kappa^k_{\boldsymbol{\rho}} = \min_{e \in E} \rho_e \ \geq 1 - \epsilon. \tag{A.55}$$

Subsequently, using arguments identical to that for proof of Lemma 2.3.1, it follows that $\widehat{\Phi}_{\boldsymbol{\rho}}(\boldsymbol{\theta})$ is $1/\sqrt{\kappa^k_{\boldsymbol{\rho}}}$ approximation. That is,

$$\sqrt{1-\epsilon} \leq \frac{\Phi(\boldsymbol{\theta})}{\widehat{\Phi}_{\boldsymbol{\rho}}(\boldsymbol{\theta})} \leq \frac{1}{\sqrt{1-\epsilon}}. \tag{A.56}$$

This completes the proof of Theorem 2.6.1.

□