Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.438 Algorithms for Inference
Fall 2020

17 Inference via Graph Partitioning

In this installment of the notes, we discuss a generic procedure for approximate inference on graphs, based in *graph partitioning.* Suppose we have a quantity, e.g., the log partition function, that we want to approximate. The key steps in the graph partitioning approach are:

1. Partition the graph into small disjoint sets.
2. Estimate the quantity for each partition independently.
3. Add the quantities for the subproblems to form a global estimate.

Because of its simplicity, this algorithm seems almost too good to be true and in fact given a specific partition on a graph, we can choose the clique potentials so that this algorithm will give a poor approximation to the quantity. The key is to exploit randomness! Instead of choosing a single partition, we'll define a distribution on partitions and when we select a partition from this distribution, we can guarantee that on average the algorithm will do well.

When a clever distribution on partitions of the graph exists, this produces a linear time algorithm and can be quite accurate. Unfortunately, not all graphs admit a good distribution on partitions, but in this case, we can produce a bound on the approximation error. In the following section, we'll precisely define the algorithm and derive bounds on the approximation error. At the end, we'll explore how to find a clever distribution on partitions.

To make the notion of a "good" distribution on partitions precise, we'll define

Definition 1. *An (ϵ, k)-partitioning of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a distribution on finite partitions of $\mathcal{V}$ such that for any partition $\{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$ with non-zero probability, $|\mathcal{V}_m| \leq k$ for all $1 \leq m \leq M$. Furthermore, we require that for any $e \in \mathcal{E}$, $\mathbb{P}\left(e \in \mathcal{E}^{\mathrm{c}}\right) \leq \epsilon$ where $\mathcal{E}^{\mathrm{c}} = \mathcal{E} \backslash \cup_m (\mathcal{V}_m \times \mathcal{V}_m)$ is the set of cut edges and the probability is with respect to the distribution on partitions.*

Intuitively, an (ϵ, k)-partitioning is a weighted set of partitions, such that in every partition all of the $\mathcal{V}_m$ are small and the set $\mathcal{E}^{\mathrm{c}}$ of cut edges is small. This aligns well with our algorithm because it means that the subproblems will be small because $\mathcal{V}_m$ is small, so our algorithm will be efficient. Because our algorithm evaluates our quantity of interest for each partition independently, it misses out on the information contained on the cut edges. However, as long as the set of cut edges is small, we do not miss much by ignoring them.

Let us consider a simple example of an $\sqrt{N} \times \sqrt{N}$ grid graph $\mathcal{G}$. We'll show that it's possible to find an $(\epsilon, \frac{1}{\epsilon^2})$-partitioning for $\mathcal{G}$ for any $\epsilon > 0$. In this case, $k = \frac{1}{\epsilon^2}$.

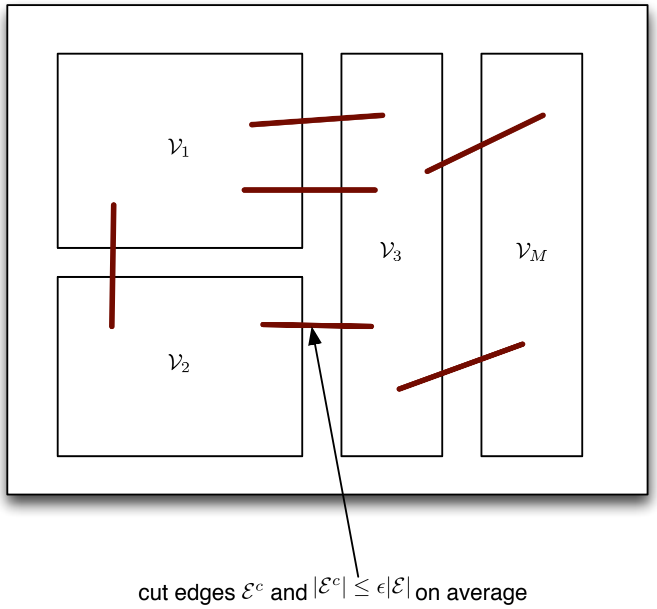
$\mathcal{V}_1$
$\mathcal{V}_2$
$\mathcal{V}_3$
$\mathcal{V}_M$
cut edges $\mathcal{E}^c$ and $|\mathcal{E}^c| \leq \epsilon|\mathcal{E}|$ on average

Figure 1: The nodes are partitioned into subsets $\mathcal{V}_1, \ldots, \mathcal{V}_M$ and the red edges correspond to cut edges.

Our strategy will be to first construct a single partition has $|\mathcal{V}_m| \leq k$ and a small $|\mathcal{E}^c|$. Then we will construct a distribution on partitions that satisfies the constraint that for any $e \in \mathcal{E}$, $p(e \in \mathcal{E}^c)$.

Sub-divide the grid into $\sqrt{k} \times \sqrt{k}$ squares, each containing k nodes (Figure 2). There are $M \triangleq \frac{N}{k}$ such sub-squares; call them $\mathcal{V}_1, \ldots, \mathcal{V}_M$. By construction $|\mathcal{V}_m| \leq k$. The edges in $\mathcal{E}^c$ are the ones that cross between sub-squares. The number of edges crossing out of each such square is at most $4\sqrt{k}$, so the total number of such edges are at most $4M\sqrt{k}\frac{1}{2}$ where $\frac{1}{2}$ is for doubling counting edges. The total number of edges in the grid is roughly $2N$. Therefore, the fraction of cut edges is $2\sqrt{k}\frac{N}{k}\frac{1}{2N} = \frac{1}{\sqrt{k}} = \epsilon$.

Thinking of the sub-division into $\sqrt{k} \times \sqrt{k}$ squares as a coarse grid, we could shift the grid to the right and/or down to create a new partition. If we randomly shift the entire sub-grid uniformly $0, \ldots, \sqrt{k} - 1$ to the right and then uniformly $0, \ldots, \sqrt{k} - 1$ down this gives a distribution on partitions. By symmetry, it ensures the distributional guarantee that $p(e \in \mathcal{E}^c) \leq \epsilon$. Thus the grid graph admits an $(\epsilon, \frac{1}{\epsilon^2})$-partitioning for any $\epsilon > 0$.

17.1 Approximation using (ϵ, k)-Partitioning

In this section, we'll prove a bound on the approximation error when we have an (ϵ, k)-partitioning. For our analysis we will restrict our attention to factorizations

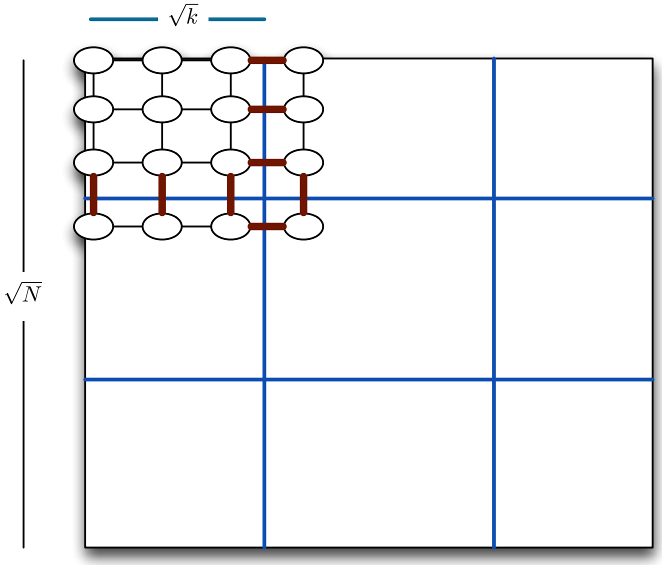

Figure 2: The original grid is sub-divided into a grid of $\sqrt{k} \times \sqrt{k}$ squares.

with respect undirected graphs with up to only pairwise potentials that have non-negative potentials, so p takes the form

$$p_{\mathbf{x}}(\mathbf{x}) \propto \exp \underbrace{\left(\sum_{i \in \mathcal{V}} \tilde{\phi}_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \tilde{\psi}_{ij}(x_i, x_j) \right)}_{\triangleq -E(\mathbf{x})} \tag{1}$$

for a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and where $\tilde{\psi}_i, \tilde{\phi}_{ij} \geq 0$.

In light of the variation characterization we discussed in previous lectures, we consider the approximation of the log partition function $\log Z$ of $p_{\mathbf{x}}$:

$$\log Z = \sup_{q \in \mathcal{P}} \left\{ -\mathbb{E}_q \left[E(\mathbf{x}) \right] + H(q) \right\}. \tag{2}$$

For this setting, we are going to show that there is a simple randomized approximation algorithm using graph partitioning whose output $\log \hat{Z}$ approximates (in expectation) the true $\log Z$ within a $(1 - \epsilon)$ factor and has $O(\exp(k))$ computational complexity.

The approximation algorithm works as follows. First, from the (ϵ, k)-partitioning of $\mathcal{G}$, sample a partition $\{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$ of $\mathcal{V}$. Next, we “modify” the distribution (1) into M factors $q_1, \ldots, q_M$ defined on each subgraph $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E} \cap (\mathcal{V}_m \times \mathcal{V}_m))$, ignoring the edge potentials of cut edges. In other words, we define

$$q_{\mathsf{x}}(\mathbf{x}) = \prod_{m=1}^{M} q_m(\mathbf{x}_{\mathcal{V}_m}) \propto \prod_{m=1}^{M} \exp\left(-E_m(\mathbf{x})\right), \tag{3}$$
$$\text{where } \; -E_m(\mathbf{x}) = \sum_{i\in\mathcal{V}_m} \tilde{\phi}_i(x_i) + \sum_{\substack{(i,j)\in\mathcal{E}:\\ i,j\in\mathcal{V}_m}} \tilde{\psi}_{ij}(x_i, x_j).$$

Then, we calculate the log partition function of q_{x}, which is our output $\log \hat{Z}$. Note that computing the *exact* partition function of q_{x} takes $O(\exp(k))$ time, because q_{x} is factorized into independent factors q_m that involve at most k random variables ($|\mathcal{V}_m| \le k$).

Now, we prove upper and lower bounds on the expected output of the algorithm.

Theorem 1.

$$(1-\epsilon)\log Z \le \mathbb{E}\left[\log \hat{Z}\right] \le \log Z, \tag{4}$$

where the expectation is taken over the (ϵ, k)*-partitioning.*

Proof. Suppose we sampled a partition $(\mathcal{V}_1, \ldots, \mathcal{V}_M)$ from the (ϵ, k)-partitioning. From the variational characterization (2), we have

$$\begin{aligned}\log Z &= \sup_{q\in\mathcal{P}} \left\{-\mathbb{E}_q\left[E(\mathbf{x})\right] + H(q)\right\} \\ &= \sup_{q\in\mathcal{P}} \left\{-\mathbb{E}_q\left[\sum_{m=1}^{M} E_m(\mathbf{x})\right] + \sum_{(i,j)\in\mathcal{E}^c} \mathbb{E}_q\left[\tilde{\psi}_{ij}(\cdot)\right] + H(q)\right\},\end{aligned}$$

where $\mathcal{E}^c$ is the set of cut edges, defined in Definition 1. Recall from Lecture 15 that the supremum in this variational characterization is attained by the true distribution p_{x} in (1). This leads to

$$\log Z = \underbrace{-\mathbb{E}_{p_{\mathsf{x}}}\left[\sum_{m=1}^{M} E_m(\mathbf{x})\right] + H(p_{\mathsf{x}})}_{=:A} + \sum_{(i,j)\in\mathcal{E}^c} \mathbb{E}_{p_{\mathsf{x}}}\left[\tilde{\psi}_{ij}(\cdot)\right].$$

We first bound A separately. For A, note that, by the definition of supremum,

$$A \le \sup_{q\in\mathcal{P}} \left\{-\mathbb{E}_q\left[\sum_{m=1}^{M} E_m(\mathbf{x})\right] + H(q)\right\}.$$

Note that the right-hand side is precisely the variational characterization for the modified distribution q_{x} defined in (3), which is equal to the log partition function of q_{x}, in other words, the output $\log \hat{Z}$ of the algorithm. Hence, we have

$$\log Z \le \log \hat{Z} + \sum_{(i,j)\in\mathcal{E}^c} \mathbb{E}_{p_{\mathsf{x}}}\left[\tilde{\psi}_{ij}(\cdot)\right].$$

Now take expectation on both sides, over the (ϵ, k)-partitioning. We then have

$$\begin{aligned}
\log Z &\le \mathbb{E}\left[\log \hat{Z}\right] + \mathbb{E}\left[\sum_{(i,j)\in\mathcal{E}^c} \mathbb{E}_{p_{\mathsf{x}}}\left[\tilde{\psi}_{ij}(\cdot)\right]\right] \\
&\le \mathbb{E}\left[\log \hat{Z}\right] + \epsilon \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{p_{\mathsf{x}}}\left[\tilde{\psi}_{ij}(\cdot)\right] \\
&\le \mathbb{E}\left[\log \hat{Z}\right] + \epsilon(-\mathbb{E}_{p_{\mathsf{x}}}\left[E(\mathbf{x})\right] + H(p_{\mathsf{x}})) \\
&= \mathbb{E}\left[\log \hat{Z}\right] + \epsilon \log Z,
\end{aligned}$$

where the second inequality used the definition of (ϵ, k)-partitioning, and the third inequality used that $\mathbb{E}_{p_{\mathsf{x}}}\left[\phi_i(\cdot)\right]$ and $H(p_{\mathsf{x}})$ terms are all nonnegative. This proves $(1-\epsilon)\log Z \le \mathbb{E}\left[\log \hat{Z}\right]$.

For the other bound, simply note that

$$\log \hat{Z} = \sup_{q\in\mathcal{P}} \left\{ -\mathbb{E}_q\left[\sum_{m=1}^{M} E_m(\mathbf{x})\right] + H(q)\right\} \le \sup_{q\in\mathcal{P}} \left\{-\mathbb{E}_q\left[E(\mathbf{x})\right] + H(q)\right\} = \log Z,$$

so the bound holds after taking expectation over the (ϵ, k)-partitioning. □

17.2 Approximate MAP inference using (ϵ, k)-Partitioning

In the same setting on distribution (1), we can also use graph partitioning to do an approximate MAP inference. Formally, the approximate MAP algorithm is

1. Given an (ϵ, k)-partitioning of $\mathcal{G}$, sample a partition $\{\mathcal{V}_1, \dots, \mathcal{V}_M\}$ of $\mathcal{V}$.
2. For each $1 \le m \le M$: Using max-product on $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E} \cap \mathcal{V}_m \times \mathcal{V}_m)$ find
$$\hat{\mathbf{x}}_m \in \operatorname*{arg\,max}_{\mathbf{y}\in\mathcal{X}^{|\mathcal{V}_m|}} -E_m(\mathbf{y}). \tag{5}$$
3. Set $\hat{\mathbf{x}} = ((\hat{\mathbf{x}}_m)_m)$ as an approximation of the MAP.

We can get a handle on the approximation error by understanding how much error arises from ignoring the edge potentials corresponding to $\mathcal{E}^{\mathrm{c}}$. If we use an (ϵ, k)-partitioning, then we expect $\mathcal{E}^{\mathrm{c}}$ to be small, so we can bound our approximation error. The following theorem from Jung and Shah [5] and Jung, Kohli and Shah [6] makes this intuition rigorous.

Theorem 2.
$$-\mathbb{E}\left[E(\hat{\mathbf{x}})\right] \ge -E(\mathbf{x}^*)(1-\epsilon). \tag{6}$$
where the expectation is taken over the (ϵ, k)-partitioning.

Proof.

$$
\begin{aligned}
-E(\mathbf{x}^*) &= \sum_{i \in \mathcal{V}} \tilde{\phi}_i(x_i^*) + \sum_{(i,j) \in \mathcal{E}} \tilde{\psi}_{ij}(x_i^*, x_j^*) \\
&= \sum_{m=1}^{M} \left[\sum_{i \in \mathcal{V}_m} \tilde{\phi}_i(x_i^*) + \sum_{\substack{(i,j) \in \mathcal{E}: \\ i,j \in \mathcal{V}_m}} \tilde{\psi}_{ij}(x_i^*, x_j^*) \right] + \sum_{(i,j) \in \mathcal{E}^c} \tilde{\psi}_{ij}(x_i^*, x_j^*) \\
&= \sum_{m=1}^{M} -E_m(\mathbf{x}^*) + \sum_{(i,j) \in \mathcal{E}^c} \tilde{\psi}_{ij}(x_i^*, x_j^*) \\
&\leq -E(\hat{\mathbf{x}}) + \sum_{(i,j) \in \mathcal{E}} \mathbb{1}_{(i,j) \in \mathcal{E}^c} \tilde{\psi}_{ij}(x_i^*, x_j^*).
\end{aligned}
$$

Therefore, by taking expectation with respect to randomness in partitioning and using the fact that $\tilde{\phi}_i, \tilde{\psi}_{i,j} \geq 0$ and $-E(\mathbf{x}^*) \geq \sum_{(i,j) \in \mathcal{E}} \tilde{\psi}_{ij}(x_i^*, x_j^*)$,

$$
-E(\mathbf{x}^*) \leq -\mathbb{E}\left[E(\hat{\mathbf{x}})\right] + \epsilon - E(\mathbf{x}^*). \tag{7}
$$

□

This means that given our choice of ϵ, we can ensure that $-\mathbb{E}\left[E(\hat{\mathbf{x}})\right]$ is close to the correct answer (i.e. the approximation error is small).

17.3 Generating (ϵ, k)-Partitions

We've seen that as long as we have an (ϵ, k)-partitioning for an MRF, then we can make guarantees about the approximation algorithm. Now we will show that a large class of graphs have (ϵ, k)-partitionings. First we'll describe a procedure for generating a **potential** (ϵ, k)-paritioning and then we'll see which class of graphs this realizes an (ϵ, k)-partitioning.

The procedure for generating a potential (ϵ, k)-partitioning on a graph is given by the following method for sampling a partition

1. Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, k, and $\epsilon > 0$. Define the *truncated geometric distribution* with parameter ϵ truncated at k as follows

$$
p(x = l) = \begin{cases} (1-\epsilon)^{l-1}\epsilon & l < k \\ (1-\epsilon)^{k-1} & l = k \end{cases}. \tag{8}
$$

2. Order the nodes $\mathcal{V}$ arbitrarily $1, \ldots, N$. For node i:

   Sample R_i from a truncated geometric distribution with parameter ϵ truncated at k.

Assign all nodes within distance[1] R_i from i color i. If the node is already colored, recolor it to i.

3. All nodes with the same color form a partition.

This gives a partition and defines a distribution on partitions. The questions is: for what graphs $\mathcal{G}$ is this distribution an (ϵ, k)-partitioning?

Intuitively, for any given node, we want the number of nodes within some distance of it not to grow too quickly. Precisely,

Definition 2 (Poly-growth graph)**.** *A graph $\mathcal{G}$ is a poly-growth graph if there exists $\rho > 0, C > 0$ such that for any vertex v in the graph,*

$$|N_v(r)| \leq Cr^\rho, \tag{9}$$

where $N_v(r)$ is the number of nodes within distance r of v in $\mathcal{G}$.

In this case, we know that [6]

Theorem 3. *If $\mathcal{G}$ is a poly-growth graph then by selecting $k = \Theta(\frac{\rho}{\epsilon}\log\frac{\rho}{\epsilon})$, [2] the above procedure results in an (ϵ, Ck^ρ) partition.*[3]

This shows that we have a large class of graphs where we can apply the procedure to generate an (ϵ, k)-partitioning, which guarantees that our approximation error is small and controlled by our choice of ϵ.

References.

[1] Bayati M., D. Shah, and M. Sharma, “Max-Product for Maximum Weight Matching: Convergence, Correctness, and LP Duality,” IEEE Transaction on Information Theory, Volume 54, No. 3, pp. 1241-1251, March 2008.

[2] Sanghavi S., Malioutov D. and Willsky A.“Belief Propagation and LP Relaxation for Weighted Matching in General Graphs”, 2011.

[3] Gamarnik D., D. Shah and Y. Wei, “Belief propagation for min-cost network flow: convergence & correctness,” Operations Research, Volume 60, No. 2, pp. 410-428, 2012.

[4] Weiss Y. and Freeman W. “Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology”, 2001.

[5] Jung K. and D. Shah, “Local approximate inference algorithms for minor excluded graphs,” in Advances in Neural Information Processing Systems, December 2007.

[6] Jung K., P. Kohli and D. Shah, “Local rules for global MAP: when do they work?,” in Advances in Neural Information Processing Systems, pp. 871-879, December 2009.

[1] Where distance is defined as the path length on the graph.

[2] This notation means that k is asymptotically bounded above and below by $\frac{\rho}{\epsilon}\log\frac{\rho}{\epsilon}$.

[3] A similar procedure exists for all planar graphs.