Sampling GMRFs by Subgraph Correction

Ying Liu
Department of EECS
Massachusetts Institute of Technology
liu_ying@mit.edu

Oliver Kosut
School of Electrical, Computer and Energy Engineering
Arizona State University
okosut@asu.edu

Alan S. Willsky
Department of EECS
Massachusetts Institute of Technology
willsky@mit.edu

Abstract

The problem of efficiently drawing samples from a Gaussian Markov random field is studied. We introduce the subgraph correction sampling algorithm, which makes use of any pre-existing tractable sampling algorithm for a subgraph by perturbing this algorithm so as to yield asymptotically exact samples for the intended distribution. The subgraph can have any structure for which efficient sampling algorithms exist: for example, tree-structured, with low tree-width, or with a small feedback vertex set. Preliminary experimental results demonstrate that the subgraph correction algorithm yields accurate samples much faster than many traditional sampling methods—such as Gibbs sampling—for many graph topologies.

1 Introduction and Background

An important family of Markov random fields (MRFs) is the family of Gaussian graphical models or Gaussian Markov random fields (GMRFs). GMRFs are often used to directly parametrize probabilistic networks and used as approximate models to circumvent the computational complexity inherent in many discrete models. This paper is devoted to developing efficient algorithms for drawing samples from a GMRF.

Samples from a GMRF can be drawn exactly by conducting a Cholesky decomposition of the covariance or inverse covariance matrix and then performing a linear transformation on *i.i.d.* Gaussian samples. However, the cubic complexity of this direct method precludes its use on large-scale models, and thus iterative MCMC-based methods or methods that exploit the internal structures of the GMRFs are often used. Graphical models with particular topologies (for example, tree-structured models) have well-known efficient sampling algorithms, but possess limited modeling power [1]. A popular sampling algorithm for general loopy graphs is Gibbs sampling, an MCMC algorithm in which variables are sequentially drawn conditioned on the most recent sample of all other variables (or of the variables in the Markov blankets in the MRF setting) [2]. However, the Gibbs sampler can have slow mixing rate in many situations, including sampling from some tree-structured MRFs, which have efficient alternative sampling algorithms. In [3], both exact methods and iterative sampling methods using blocking or divide-and-conquer strategies are studied. In a recent paper [4], an exact method using local perturbation is proposed for GMRFs with a sparse decomposition (sparse filters) and an efficient linear system solver for the entire model.

In this paper, we introduce the *subgraph correction algorithm*, which leverages on any pre-existing efficient algorithm to sample from a spanning subgraph[1] and randomly perturbs parameters so as to generate correct samples in the long run. With a tree-structured subgraph, our algorithm is a randomized extension of the embedded-tree algorithm [5], which has been shown to have excellent convergence properties [5]. As our algorithm produces an exact sample from the target distribution asymptotically, the number of iterations required to generate a true sample depends on the convergence rate. We provide theoretical analysis guaranteeing

[1] A spanning subgraph contains all the nodes and a subset of the edges.

convergence for the subgraph correction algorithm. Furthermore, we characterize the convergence rate, both exactly and via tractable bounds. We run experiments using GMRFs of various structures and different sizes to demonstrate that the algorithm converges quickly on a wide variety of graphs. As our research is ongoing, we are currently working on experiments on very large-scale models.

2 Sampling GMRFs by Subgraph Correction

Consider a GMRF with probability density function $p(\mathbf{x}) \propto \exp\{-\frac{1}{2}\mathbf{x}^T J\mathbf{x} + \mathbf{h}^T\mathbf{x}\}$. The subgraph correction algorithm builds on graphical splitting preconditioning algorithms for solving large linear systems [6]. In our context, determining the means of the graphical model corresponds to solving the equation $J\boldsymbol{\mu} = \mathbf{h}$. A matrix splitting for solving this equation would correspond to writing $J = J_{\mathcal{T}} - K$ and using the relationship $J_{\mathcal{T}}\boldsymbol{\mu} = K\boldsymbol{\mu} + \mathbf{h}$ as the basis for an iterative algorithm. An underlying assumption here is that the structure of $J_{\mathcal{T}}$ is such that solving equations of the form $J_{\mathcal{T}}\mathbf{x} = \boldsymbol{b}$ is easy.

Our iterative algorithm is quite simple to describe: At each iteration, rather than solving $J_{\mathcal{T}}\boldsymbol{\mu}^{(t+1)} = K\boldsymbol{\mu}^{(t)} + \mathbf{h}$—equivalent to finding the mean of a distribution with information matrix $J_{\mathcal{T}}$ and potential vector $K\boldsymbol{\mu}^{(t)} + \mathbf{h}$—instead we draw a sample from a Gaussian with information matrix $J_{\mathcal{T}}$ and potential vector $K\mathbf{x}^{(t)} + \mathbf{h} + \mathbf{e}^{(t+1)}$. The vector $\mathbf{e}^{(t+1)}$ is Gaussian with zero mean and covariance K. It represents a perturbation to the potential vector that exactly compensates for the discrepancy between the spanning subgraph and the full graph. Moreover, $\mathbf{e}^{(t+1)}$ can be constructed locally for distributed computation. The matrix splitting is given by

$$K_{ij} = \begin{cases} \sum_{k:(i,k)\in\mathcal{E}\setminus\mathcal{E}_{\mathcal{T}}} |J_{ik}| & \text{if } i = j \\ -J_{ij} & \text{if } i \neq j,\ (i,j) \in \mathcal{E} \setminus \mathcal{E}_{\mathcal{T}} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$(J_{\mathcal{T}})_{ij} = \begin{cases} J_{ii} + \sum_{k:(i,k)\in\mathcal{E}\setminus\boldsymbol{\mathcal{E}_{\mathcal{T}}}} |J_{ik}| & \text{if } i = j \\ J_{ij} & \text{if } i \neq j,\ (i,j) \in \mathcal{E}_{\mathcal{T}} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

It is easy to see that, as required, $J_{\mathcal{T}} - K = J$, and that both $J_{\mathcal{T}}$ and K are positive semi-definite. In particular, K can be decomposed into a sum of rank-one matrices, so it is quite easy to draw samples from a Gaussian distribution with covariance K, as follows: For each $(i,j) \in \mathcal{E}\setminus\mathcal{E}_{\mathcal{T}}$, let $\mathbf{e}^{(i,j)} = [\ |J_{ij}| \quad J_{ij}]^T u_{ij}$, where u_{ij} is drawn independently from an zero-mean unit-variance Gaussian distribution. We can obtain $\mathbf{e}^{(t+1)}$ by computing

$$\mathbf{e}^{(t+1)} = \sum_{(i,j)\in\mathcal{E}\setminus\mathcal{E}_{\mathcal{T}}} \left[\mathbf{e}^{(i,j)}\right]_n, \tag{3}$$

where $\left[\mathbf{e}^{(i,j)}\right]_n$ is an n-dimensional vector zero-padded from $\mathbf{e}^{(i,j)}$, i.e., the i-th and j-th entries of $\left[\mathbf{e}^{(i,j)}\right]_n$ take the two entries of $\mathbf{e}^{(i,j)}$and all other entries of $\left[\mathbf{e}^{(i,j)}\right]_n$ are zero.

The subgraph correction algorithm is summarized in Algorithm 1. As we can easily see from Algorithm 1, the computational complexity of one iteration is $C_{\mathcal{T}} + \mathcal{O}(|\mathcal{E}_K|)$, where $C_{\mathcal{T}}$ is the complexity of drawing a sample from the tractable subgraph $\mathcal{T}$ and $|\mathcal{E}_K| = |\mathcal{E} - \mathcal{E}_{\mathcal{T}}|$ is the number of edges missing from $J_{\mathcal{T}}$.

3 Theoretical Results

In this section, we state some theoretical results on the convergence of the algorithm. Proofs are omitted due to the page limit. We give both the exact convergence rate[2] and its tractable bounds.

Proposition 1. *For a GMRF with information matrix $J \succ 0$ and potential vector $\mathbf{h}$, the sample distribution generated by Algorithm 1 is guaranteed to converge to the exact distribution and the convergence rate is $\rho(J_{\mathcal{T}}^{-1}K) < 1$.*

[2]Here the convergence rate is defined as $-\limsup_{t\to\infty} \ln\left(||\Sigma^{(t+1)} - \Sigma||/||\Sigma^{(t)} - \Sigma||\right)$, where $\Sigma^{(t)}$ is the sample covariance at iteration t and $\Sigma = J^{-1}$ is the exact target covariance. The larger the convergence rate is, the faster the sample distribution converges to the target distribution.

Algorithm 1 Sampling by Subgraph Correction

Input: J, $\mathbf{h}$, and $\mathcal{T}$, where $\mathcal{T}$ corresponds to a tractable subgraph.
Output: samples with the asymptotic distribution $\mathcal{N}(J^{-1}\mathbf{h}, J^{-1})$.

1. Form $J_{\mathcal{T}}$ and K using (1)–(2).
2. Draw an initial sample $\mathbf{x}^{(0)}$ from a Gaussian distribution with potential vector $\mathbf{h}$ and information matrix being the diagonal of J.
3. In each iteration:
   (a) For each $(i,j) \in \mathcal{E} \setminus \mathcal{E}_{\mathcal{T}}$, draw u_{ij} from $\mathcal{N}(0,1)$.
   (b) Compute $\mathbf{e}^{(t+1)}$ using (3).
   (c) Draw a new sample $\mathbf{x}^{(t+1)}$ from a Gaussian distribution with information matrix $J_{\mathcal{T}}$ and potential vector $\mathbf{h} + Kx^{(t)} + \mathbf{e}^{(t+1)}$.

Proposition 2 and Corollary 1 give bounds on the convergence rate of the subgraph correction algorithm.

Proposition 2. *Consider symmetric matrices J, $J_{\mathcal{T}}$, and K that satisfy $J = J_{\mathcal{T}} - K$. If $J \succ 0$ and $K \succeq 0$, then*

$$\frac{\lambda_{\max}(K)}{\lambda_{\max}(K) + \lambda_{\max}(J)} \leq \rho(J_{\mathcal{T}}^{-1}K) \leq \frac{\lambda_{\max}(K)}{\lambda_{\max}(K) + \lambda_{\min}(J)} < 1.$$

We can further bound $\lambda_{\max}(K)$, the largest eigenvalue of K, using a simple function of its entries. We define the weight of node i in a GMRF with information matrix K as $w(i) = \sum_j |K_{ij}|$ and the weight of the model as $w(K) = \max_i w(i)$. Corollary 1 follows immediately.

Corollary 1. *In the same setting as in Proposition 2, we have $\rho(J_{\mathcal{T}}^{-1}K) \leq \frac{\mathrm{w}(K)}{w(K)+\lambda_{\min}(J)}$*

4 On Selecting Tractable Subgraphs

Our algorithm does not restrict the subgraph to be tree-structured and any subgraph with an existent fast sampling method can be used, such as subgraphs with low tree-width [7], or subgraphs with small feedback vertex sets (FVS[3]) [8]. The computational complexity of one iteration is the complexity of generating one sample from the tractable subgraph, plus a term proportional to the number of missing edges from the subgraph. For many sparse graphs, the complexity is linear with respect to the number of nodes. Although we have presented the algorithm using a single, constant splitting for clarity, using different trees or other tractable structures on different iterations can be very beneficial, as it is when calculating the means in the inference case [9]. By Proposition 1, to speed convergence, $J_{\mathcal{T}}$ should be selected to minimize $\rho(J_{\mathcal{T}}^{-1}K)$. In this section, we give brief references to the selection algorithms for different families of tractable subgraphs.

Tree-Structured Subgraphs The idea of using a maximum spanning tree (MST) has been discussed in the support graph preconditioner literature [6] as well as in the studies of the embedded tree algorithm for inference [9], where multiple embedded trees are selected adaptively.

Subgraphs with Low Tree-width Graphical models with low tree-width have efficient inference and sampling algorithms and have been widely studied. We can compute a low tree-width approximation $J_{\mathcal{T}}$ to J using algorithms such as those in [10, 11, 7].

Subgraphs with Small FVS The key step of obtaining the structure of a graph with a small FVS is the selection of the FVS. We first use the heuristic in [8] to select a pseudo-FVS (a subset of an FVS that breaks most crucial cycles, but not all the cycles) of the whole graph. The structure of the subgraph with a small FVS is constructed by combing the nodes in the FVS (with all their incident edges) and the MST of the remaining graph. There is a trade-off in choosing the FVS size (a larger FVS means more computation per iteration but faster convergence).

Spectrally Sparsified Subraphs Many common GMRFs such as thin-membrane or thin-plate models have diagonally dominant information matrices. Some recent studies show that the graph Laplacian of a dense graph can be well-approximated by the graph Laplacian of graphs with nearly-linear number of edges [12]. These spectrally sparsified graphs have efficiently inference and sampling algorithms and can be used as tractable subgraphs.

[3] An FVS is a set of nodes whose removal results in a cycle-free graph.

5 Experimental Results

In this section, we present some preliminary experimental results showing the subgraph correction algorithm on graphs of various structures and compare the performance with the Gibbs sampler. For a given graph structure, the model parameters J and $\mathbf{h}$ are randomly generated as follows: the entries of the potential vector $\mathbf{h}$ are generated $i.i.d.$ from a uniform distribution $U[-1, 1]$; the edge weights of J are also generated *i.i.d.* from $U[-1, 1]$ with a multiple of the identity matrix added to ensure $J \succ 0$. We compute the numbers of iterations needed to achieve an approximating error of $\epsilon = 10^{-5}$, i.e., the minimum t such that $\left\|\Sigma^{(t)} - \Sigma\right\| \leq \epsilon$. For each graph structure and each size, we repeat the algorithm for 100 sets of random model parameters and the results shown are averaged over the 100 runs.

We run the subgraph correction algorithm on cycles of 3 to 100 nodes. The tractable subgraph we use for a cycle is a spanning tree (for a cycle, it is equivalent to removing the edge with the weakest normalized weight from the graph). As shown in Figure 1, our algorithm gives a substantial reduction of the number of iterations needed compared with the Gibbs sampler. For example, for a cycle of length 100, the Gibbs sampler needs more than 600 iterations on average while our algorithm only needs fewer than ten. We also test the algorithm on l-by-l grids with l ranging from 3 to 30. For each grid, two different subgraphs are used: one is a tree-structured subgraph, the other is a graph with an FVS of size $\lceil \log l^2 \rceil$. Figure 2 shows that both kinds of subgraphs give better convergence than the Gibbs sampler while the subgraphs with small FVSs perform the best consistently.

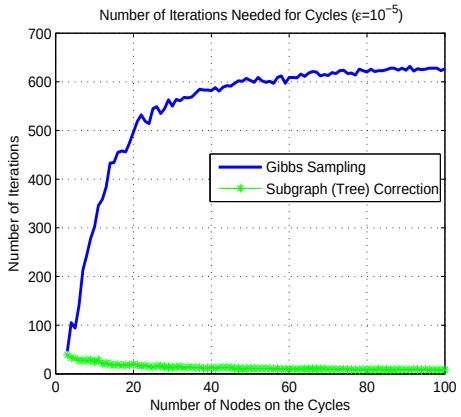

Figure 1: the performance of subgraph correction on cycles of size 3 to 100. The subgraphs used are tree-structured.

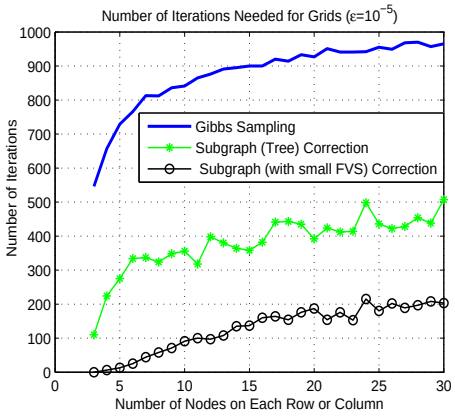

Figure 2: the performance of subgraph correction on grids of size 3-by-3 to 30-by-30. The subgraphs used include tree-structured graphs and graphs with small FVSs.

6 Conclusion

This paper introduced the subgraph correction algorithm for sampling from GMRFs, which takes advantage of a subgraph with an existing fast solvers. We gave theoretical results on the convergence rate and tractable bounds, as well as experimental results. Our analysis found the algorithm to be particularly effective when the model can be decomposed into a tree and a set of edges of small weights. We also found that even for graph as apparently un-tree-like as a grid the algorithm performs quite well as compared to Gibbs sampling. We limited our analysis to a fixed subgraph, but the algorithm naturally extends to allow for different subgraphs to be used at different iterations. This sequence of subgraphs can be selected *a priori* or on the fly. This idea is particularly useful if the model changes over time. Currently, we are working on expanding the experiments on very large-scale models and compare more extensively with other existing methods. Studying the trade-off between the convergence rate and the complexity of one iteration is also an interesting future research direction.

References

[1] D. Batra, A. Gallagher, D. Parikh, and T. Chen, "Beyond trees: MRF inference via outer-planar decomposition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE,

2010, pp. 2496–2503.

[2] C. Bishop, *Pattern recognition and machine learning*. Springer New York:, 2006.

[3] H. Rue, "Fast sampling of Gaussian Markov random fields," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 325–338, 2001.

[4] G. Papandreou and A. Yuille, "Gaussian sampling by local perturbations," in *Proc. NIPS*, 2010.

[5] E. Sudderth, M. Wainwright, and A. Willsky, "Embedded trees: estimation of Gaussian processes on graphs with cycles," *IEEE Transactions on Signal Processing*, vol. 52, no. 11, pp. 3136–3150, 2004.

[6] M. Bern, J. Gilbert, B. Hendrickson, N. Nguyen, and S. Toledo, "Support-graph preconditioners," *SIAM Journal on Matrix Analysis and Applications*, vol. 27, no. 4, pp. 930–951, 2006.

[7] D. Karger and N. Srebro, "Learning Markov networks: Maximum bounded tree-width graphs," in *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2001, pp. 392–401.

[8] Y. Liu, V. Chandrasekaran, A. Anandkumar, and A. Willsky, "Feedback message passing for inference in gaussian graphical models," *Signal Processing, IEEE Transactions on*, vol. 60, no. 8, pp. 4135–4150, 2012.

[9] V. Chandrasekaran, J. Johnson, and A. Willsky, "Estimation in Gaussian graphical models using tractable subgraphs: A walk-sum analysis," *Signal Processing, IEEE Transactions on*, vol. 56, no. 5, pp. 1916–1930, 2008.

[10] N. Srebro, "Maximum likelihood bounded tree-width Markov networks," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 504–511.

[11] D. Shahaf, A. Chechetka, and C. Guestrin, "Learning thin junction trees via graph cuts," in *Conference on Artificial Intelligence and Statistics*, 2009.

[12] D. Spielman and S. Teng, "Spectral Sparsification of Graphs," *SIAM Journal on Computing*, vol. 40, p. 981, 2011.