

---

# Stochastic Learning for Sparse Discrete Markov Random Fields with Controlled Gradient Approximation Error

---

Sinong Geng \*  
UW-Madison

Zhaobin Kuang \*  
UW-Madison

Jie Liu  
University of Washington

Stephen Wright  
UW-Madison

David Page  
UW-Madison

## Abstract

We study the  $L_1$ -regularized maximum likelihood estimator/estimation (MLE) problem for discrete Markov random fields (MRFs), where efficient and scalable learning requires both sparse regularization and approximate inference. To address these challenges, we consider a stochastic learning framework called stochastic proximal gradient (SPG; Honorio 2012a, Atchade et al. 2014, Miasojedow and Rejchel 2016). SPG is an *inexact* proximal gradient algorithm [Schmidt et al., 2011], whose inexactness stems from the stochastic oracle (Gibbs sampling) for gradient approximation – exact gradient evaluation is infeasible in general due to the NP-hard inference problem for discrete MRFs [Koller and Friedman, 2009]. Theoretically, we provide novel *verifiable* bounds to inspect and control the quality of gradient approximation. Empirically, we propose the *tighten asymptotically* (TAY) learning strategy based on the verifiable bounds to boost the performance of SPG.

## 1 INTRODUCTION

Markov random fields (MRFs, a.k.a. Markov networks, undirected graphical models) are a compact representation of the joint distribution among multiple variables, with each variable being a node and an edge between two nodes indicating conditional dependence between the two corresponding variables. Sparse discrete MRF learning is proposed in the seminal work of

Lee et al. [2006]. By considering an  $L_1$ -regularized MLE problem, many components of the parameterization are driven to zero, yielding a sparse solution to structure learning. However, in general, solving an  $L_1$ -regularized MLE problem exactly for a discrete MRF is infamously difficult due to the NP-hard inference problem posed by exact gradient evaluation [Koller and Friedman, 2009]. We hence inevitably have to compromise accuracy for the gain of efficiency and scalability via *inexact* learning techniques [Liu and Page, 2013, Liu et al., 2014b, 2016, Geng et al., 2018].

In this paper, we consider stochastic proximal gradient (SPG; Honorio 2012a, Atchade et al. 2014, Miasojedow and Rejchel 2016), a stochastic learning framework for  $L_1$ -regularized discrete MRFs. SPG hinges on a stochastic oracle for gradient approximation of the log-likelihood function (inexact inference). However, both the theoretical guarantees and the practical performances of existing algorithms are unsatisfactory.

The stochastic oracle behind SPG is Gibbs sampling [Levin et al., 2009], which is an effective approach to draw samples from an intractable probability distribution. With enough samples, the intractable distribution can be approximated effectively by the empirical distribution, and hence many quantities (e.g., the gradient of the log-likelihood function) related to the intractable distribution can be estimated efficiently. Since SPG uses Gibbs sampling for gradient approximation, it can be viewed as an inexact proximal gradient method [Schmidt et al., 2011], whose success depends on whether the gradient approximation error can be effectively controlled. While previous works [Honorio, 2012a, Atchade et al., 2014, Miasojedow and Rejchel, 2016] have shown that the quality of the gradient approximation can be improved *in the long run* with increasingly demanding computational resources, such long term guarantees might not translate to satisfactory performance in practice (see Section 7). Therefore, it is desirable to estimate and control the

---

\* Sinong Geng and Zhaobin Kuang contribute equally. Their names are listed in alphabetical order. Corresponds to: sgeng2@wisc.edu.

gradient approximation error of SPG meticulously in each iteration so that a more refined approximation to the exact gradient will be rewarded with a higher gain of efficiency and accuracy in practice.

Careful analysis and control of the quality of the gradient approximation of SPG call for the cross-fertilization of theoretical and empirical insights from stochastic approximate inference [Bengio and Delalleau, 2009, Fischer and Igel, 2011], inexact proximal methods [Schmidt et al., 2011], and statistical sampling [Mitliagkas and Mackey, 2017]. Our contributions are hence both theoretical and empirical. Theoretically, we provide novel *verifiable* bounds (Section 4) to inspect and control the gradient approximation error induced by Gibbs sampling. Also, we provide a proof sketch for the main results in Section 5. Empirically, we propose the *tighten asymptotically* (TAY) learning strategy (Section 6) based on the verifiable bounds to boost the performance of SPG.

## 2 BACKGROUND

We first introduce  $L_1$ -regularized discrete MRFs in Section 2.1. We then briefly review SPG as a combination of proximal gradient for sparse statistical learning and Gibbs sampling for addressing the intractable exact gradient evaluation problem.

### 2.1 $L_1$ -Regularized Discrete MRF

For the derivation, we focus on the binary pairwise case and we illustrate that our framework can be generalized to other models in Section 6. Let  $\mathbf{X} = [X_1, X_2, \dots, X_p]^\top \in \{0, 1\}^p$  be a  $p \times 1$  binary random vector. We use an uppercase letter such as  $X$  to denote a random variable and the corresponding lowercase letter to denote a particular *assignment* of the random variable, i.e.,  $X = x$ . We also use boldface letters to represent vectors and matrices and regular letters to represent scalars. We define the function  $\psi : \{0, 1\}^p \rightarrow \{0, 1\}^m$ ,  $\mathbf{x} \rightarrow \psi(\mathbf{x})$  to represent the *sufficient statistics* (a.k.a. *features*) whose values depend on the assignment  $\mathbf{x}$  and compose an  $m \times 1$  vector  $\psi(\mathbf{x})$ , with its  $j^{\text{th}}$  component denoted as  $\psi_j(\mathbf{x})$ . We use  $\mathbb{X}$  to represent a dataset with  $n$  independent and identically distributed (i.i.d.) samples.

With the notation introduced above, the  $L_1$ -regularized discrete MRF problem can be formulated as the following convex optimization problem:

$$\hat{\theta} = \arg \min_{\theta \in \Theta} -\frac{1}{n} \sum_{\mathbf{x} \in \mathbb{X}} \theta^\top \psi(\mathbf{x}) + A(\theta) + \lambda \|\theta\|_1, \quad (1)$$

---

### Algorithm 1 Gibbs Sampling (Gibbs-1)

---

**Require:** initial samples  $\mathbb{S}_0$  and  $\theta$ .

**Ensure:**  $\mathbb{S}$ .

```

1: function GIBBS-1( $\mathbb{S}_0, \theta$ )
2:    $\mathbb{S} \leftarrow \mathbb{S}_0$ , and decide  $p$  from  $\mathbb{S}_0$ .
3:   for  $i \in \{1, \dots, p\}$  do
4:     for  $\mathbf{x} \in \mathbb{S}$  do
5:       Compute  $P_\theta(X_i | \mathbf{x}_{-i})$  according to (5).
6:       Update  $x_i$  by  $P_\theta(X_i | \mathbf{x}_{-i})$ .
7:     end for
8:   end for
9:   return  $\mathbb{S}$ .
10: end function

```

---



---

### Algorithm 2 Gradient Approximation (GRAD)

---

**Require:**  $\theta$ ,  $\mathbb{E}_{\mathbb{X}} \psi(\mathbf{x})$ , and  $q$ .

**Ensure:**  $\Delta f(\theta)$ .

```

1: function GRAD( $\theta, \mathbb{E}_{\mathbb{X}} \psi(\mathbf{x}), q$ )
2:   Initialize  $\mathbb{S}$  with  $q$  samples.
3:   while true do
4:      $\mathbb{S} \leftarrow \text{GIBBS-1}(\mathbb{S}, \theta)$ .
5:     if stopping criteria met then
6:       Compute  $\mathbb{E}_{\mathbb{S}} \psi(\mathbf{x})$  according to (6).
7:        $\Delta f(\theta) \leftarrow \mathbb{E}_{\mathbb{S}} \psi(\mathbf{x}) - \mathbb{E}_{\mathbb{X}} \psi(\mathbf{x})$ .
8:       break.
9:     end if
10:  end while
11:  return  $\Delta f(\theta)$ .
12: end function

```

---



---

### Algorithm 3 Stochastic Proximal Gradient (SPG)

---

**Require:**  $\mathbb{X}$ ,  $\lambda$ , and  $q$ .

**Ensure:**  $\tilde{\theta}$ .

```

1: function SPG( $\mathbb{X}, \lambda, q$ )
2:   Compute  $\mathbb{E}_{\mathbb{X}} \psi(\mathbf{x})$  according to (4).
3:   Initialize  $\theta^{(0)}$  randomly and  $k \leftarrow 0$ .
4:   Choose step length  $\alpha$ .
5:   while true do
6:      $\Delta f(\theta^{(k)}) \leftarrow \text{GRAD}(\theta^{(k)}, \mathbb{E}_{\mathbb{X}} \psi(\mathbf{x}), q)$ .
7:      $\theta^{(k+1)} \leftarrow \mathcal{S}_{\alpha\lambda}(\theta^{(k)} - \alpha \Delta f(\theta^{(k)}))$ .
8:     if Stopping criteria met then
9:        $\tilde{\theta} = \theta^{(k+1)}$ , return  $\tilde{\theta}$ .
10:    end if
11:     $k \leftarrow k + 1$ 
12:  end while
13: end function

```

---

with

$$A(\theta) = \log \sum_{\mathbf{x} \in \{0, 1\}^p} \exp(\theta^\top \psi(\mathbf{x})),$$

where  $\Theta \subseteq \mathbb{R}^m$  is the parameter space of  $\theta$ 's,  $\lambda \geq 0$ ,

and  $A(\boldsymbol{\theta})$  is the *log partition function*. We denote the differentiable part of (1) as

$$f(\boldsymbol{\theta}) = -\frac{1}{n} \sum_{\mathbf{x} \in \mathbb{X}} \boldsymbol{\theta}^\top \boldsymbol{\psi}(\mathbf{x}) + A(\boldsymbol{\theta}). \quad (2)$$

Solving (1) requires evaluating the gradient of  $f(\boldsymbol{\theta})$ , which is given by:

$$\nabla f(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}} \boldsymbol{\psi}(\mathbf{x}) - \mathbb{E}_{\mathbb{X}} \boldsymbol{\psi}(\mathbf{x}), \quad (3)$$

with

$$\mathbb{E}_{\boldsymbol{\theta}} \boldsymbol{\psi}(\mathbf{x}) = \sum_{\mathbf{x} \in \{0,1\}^p} P_{\boldsymbol{\theta}}(\mathbf{x}) \boldsymbol{\psi}(\mathbf{x}), \quad \mathbb{E}_{\mathbb{X}} \boldsymbol{\psi}(\mathbf{x}) = \frac{1}{n} \sum_{\mathbf{x} \in \mathbb{X}} \boldsymbol{\psi}(\mathbf{x}). \quad (4)$$

$\mathbb{E}_{\boldsymbol{\theta}} \boldsymbol{\psi}(\mathbf{x})$  represents the expectation of the sufficient statistics under  $P_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{\exp(\boldsymbol{\theta}^\top \boldsymbol{\psi}(\mathbf{x}))}{\exp(A(\boldsymbol{\theta}))}$ , which is a discrete MRF probability distribution parameterized by  $\boldsymbol{\theta}$ .  $\mathbb{E}_{\mathbb{X}} \boldsymbol{\psi}(\mathbf{x})$  represents the expectation of the sufficient statistics under the empirical distribution. Computing  $\mathbb{E}_{\mathbb{X}} \boldsymbol{\psi}(\mathbf{x})$  is straightforward, but computing  $\mathbb{E}_{\boldsymbol{\theta}} \boldsymbol{\psi}(\mathbf{x})$  exactly is intractable due to the entanglement of  $A(\boldsymbol{\theta})$ . As a result, various approximations have been made [Wainwright et al., 2007, Höfling and Tibshirani, 2009, Viallon et al., 2014].

## 2.2 Stochastic Proximal Gradient

To efficiently solve (1), many efforts have been made in combining Gibbs sampling [Levin et al., 2009] and proximal gradient descent [Parikh et al., 2014] into SPG, a method that adopts the proximal gradient framework to update iterates, but uses Gibbs sampling as a stochastic oracle to approximate the gradient when the gradient information is needed [Honorio, 2012a, Atchade et al., 2014, Miasojedow and Rejchel, 2016].

Specifically, Gibbs sampling with  $q$  chains running  $\tau$  steps (Gibbs- $\tau$ ) can generate  $q$  samples for  $P_{\boldsymbol{\theta}}(\mathbf{x})$ . Gibbs- $\tau$  is achieved by iteratively applying Gibbs-1 for  $\tau$  times. Gibbs-1 is summarized in Algorithm 1, where

$$P_{\boldsymbol{\theta}}(X_i | \mathbf{x}_{-i}) = P_{\boldsymbol{\theta}}(x_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p) \quad (5)$$

represents the conditional distribution of  $X_i$  given the assignment of the remaining variables  $\mathbf{x}_{-i}$  under the parameterization  $\boldsymbol{\theta}$ . Denoting the set of these  $q$  (potentially repetitive) samples as  $\mathbb{S}$ , we can approximate  $\mathbb{E}_{\boldsymbol{\theta}} \boldsymbol{\psi}(\mathbf{x})$  by the easily computable

$$\mathbb{E}_{\mathbb{S}} \boldsymbol{\psi}(\mathbf{x}) = \frac{1}{q} \sum_{\mathbf{x} \in \mathbb{S}} \boldsymbol{\psi}(\mathbf{x}) \quad (6)$$

and thus reach the approximated gradient  $\Delta f(\boldsymbol{\theta}) = \mathbb{E}_{\mathbb{S}} \boldsymbol{\psi}(\mathbf{x}) - \mathbb{E}_{\mathbb{X}} \boldsymbol{\psi}(\mathbf{x})$  with the gradient

approximation error:

$$\delta(\boldsymbol{\theta}) = \Delta f(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\theta}).$$

By replacing  $\nabla f(\boldsymbol{\theta})$  with  $\Delta f(\boldsymbol{\theta})$  in proximal gradient, the update rule for SPG can be derived as  $\boldsymbol{\theta}^{(k+1)} = \mathcal{S}_{\alpha\lambda}(\boldsymbol{\theta}^{(k)} - \alpha \Delta f(\boldsymbol{\theta}^{(k)}))$ , where  $\alpha > 0$  is the step length and  $\mathcal{S}_{\lambda}(\mathbf{a})$  is the soft-thresholding operator whose value is also an  $m \times 1$  vector, with its  $i^{\text{th}}$  component defined as  $\mathcal{S}_{\lambda}(\mathbf{a})_i = \text{sgn}(a_i) \max(0, |a_i| - \lambda)$  and  $\text{sgn}(a_i)$  is the sign function.

By defining

$$\begin{aligned} \mathbf{G}_{\alpha}(\boldsymbol{\theta}^{(k)}) &:= \frac{1}{\alpha} \left( \boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^{(k+1)} \right) \\ &= \frac{1}{\alpha} \left( \boldsymbol{\theta}^{(k)} - \mathcal{S}_{\alpha\lambda} \left( \boldsymbol{\theta}^{(k)} - \alpha \Delta f(\boldsymbol{\theta}^{(k)}) \right) \right), \end{aligned} \quad (7)$$

we can rewrite the previous update rule in a form analogous to the update rule of a standard gradient descent, resulting in the update rule of a *generalized gradient descent* algorithm:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha \mathbf{G}_{\alpha}(\boldsymbol{\theta}^{(k)}). \quad (8)$$

SPG is summarized in Algorithm 3. Its gradient evaluation procedure based on Algorithm 1 is given in Algorithm 2.

## 3 MOTIVATION

Both practical performance and theoretical guarantees of SPG are still far from satisfactory. Empirically, there are no convincing schemes for selecting  $\tau$  and  $q$ , which hinders the efficiency and accuracy of SPG. Theoretically, to the best of our knowledge, existing non-asymptotic convergence rate guarantees can only be achieved for SPG with an averaging scheme [Schmidt et al., 2011, Honorio, 2012a, Atchade et al., 2014] (see also Section 3.3), instead of ordinary SPG. In contrast, in the exact proximal gradient descent method, the objective function value is non-decreasing and convergent to the optimal value under some mild assumptions [Parikh et al., 2014]. In Section 3.2, we identify that the absence of non-asymptotic convergence rate guarantee for SPG primarily comes from the existence of gradient approximation error  $\delta(\boldsymbol{\theta})$ . In Section 3.3, we further validate that the objective function value achieved by SPG is also highly dependent on  $\delta(\boldsymbol{\theta})$ . These issues bring about the demand of inspecting and controlling  $\delta(\boldsymbol{\theta})$  in each iteration.

### 3.1 Setup and Assumptions

For the ease of presentation, we rewrite the objective function in (1) as  $g(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) + h(\boldsymbol{\theta})$ , where  $h(\boldsymbol{\theta}) = \lambda \|\boldsymbol{\theta}\|_1$ , and  $f(\boldsymbol{\theta})$  is given in (2). Since  $\nabla f(\boldsymbol{\theta})$  is Lipschitz continuous [Honorio, 2012b], we denote its Lipschitz constant as  $L$ . We also make the same assumption that  $\alpha \leq 1/L$  as Schmidt et al. [2011].

### 3.2 Decreasing Objective

It is well-known that exact proximal gradient enjoys a  $O(\frac{1}{k})$  convergence rate [Parikh et al., 2014]. One premise for this convergence result is that the objective function value decreases in each iteration. However, satisfying the decreasing condition is much more intricate in the context of SPG. Theorem 1 clearly points out that  $\boldsymbol{\delta}(\boldsymbol{\theta})$  is one main factor determining whether the objective function decreases in SPG.

**Theorem 1.** Let  $\boldsymbol{\theta}^{(k)}$  be the iterate of SPG after the  $k^{\text{th}}$  iteration. Let  $\boldsymbol{\theta}^{(k+1)}$  be defined as in (8). With  $\alpha \leq 1/L$ , we have

$$g(\boldsymbol{\theta}^{(k+1)}) - g(\boldsymbol{\theta}^{(k)}) \leq \alpha \boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})^\top \mathbf{G}_\alpha(\boldsymbol{\theta}^{(k)}) - \frac{\alpha}{2} \|\mathbf{G}_\alpha(\boldsymbol{\theta}^{(k)})\|_2^2.$$

Furthermore, a sufficient condition for  $g(\boldsymbol{\theta}^{(k+1)}) < g(\boldsymbol{\theta}^{(k)})$  is

$$\|\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})\|_2 < \frac{1}{2} \|\mathbf{G}_\alpha(\boldsymbol{\theta}^{(k)})\|_2.$$

According to Theorem 1, if the magnitude of the noise, quantified by  $\|\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})\|_2$ , is reasonably small, the objective function value decreases in each iteration. Under this condition, we can further construct a theoretical support for the convergence rate of the objective function value in the Section 3.3.

### 3.3 Convergence Rate

Assuming that  $\boldsymbol{\delta}(\boldsymbol{\theta})$  is small enough in each iteration to generate a decreasing objective value sequence, we can derive Theorem 2 following Proposition 1 in Schmidt et al. [2011]:

**Theorem 2.** Let  $\mathcal{K} = (\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots, \boldsymbol{\theta}^{(\kappa)})$  be the iterates generated by Algorithm 3. Then if  $g(\boldsymbol{\theta}^{(k+1)}) \leq g(\boldsymbol{\theta}^{(k)})$  with  $k \in \{1, 2, \dots, \kappa - 1\}$ , we have

$$g(\boldsymbol{\theta}^{(\kappa)}) - g(\hat{\boldsymbol{\theta}}) \leq \frac{L}{2\kappa} \left( \|\boldsymbol{\theta}^{(0)} - \hat{\boldsymbol{\theta}}\|_2 + \frac{2}{L} \sum_{k=1}^{\kappa} \|\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})\|_2 \right)^2. \quad (9)$$

Recall that  $\hat{\boldsymbol{\theta}}$  is an optimal solution to the sparse MLE problem defined in (1). From (9), it is obvious that if the gradient approximation error is reasonably small, then during the early iterations of SPG,  $\|\boldsymbol{\theta}^{(0)} - \hat{\boldsymbol{\theta}}\|_2$  dominates  $\frac{2}{L} \sum_{k=1}^{\kappa} \|\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})\|_2$ . Therefore, in the beginning, the convergence rate is  $O(1/\kappa)$ . However, as the iteration proceeds,  $\frac{2}{L} \sum_{k=1}^{\kappa} \|\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})\|_2$  accumulates and hence in practice SPG can only maintain a convergence rate of  $O(1/\kappa)$  up to some noise level that is closely related to  $\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})$ . Therefore,  $\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})$  plays an importance role in the performance of SPG.

Notice that Theorem 2 offers convergence analysis of the objective function value in the last iteration  $g(\boldsymbol{\theta}^{(\kappa)})$ . This result is different from the existing non-asymptotic analysis on  $g(\sum_{k=1}^{\kappa} \boldsymbol{\theta}^{(k)}/\kappa)$ , the objective function evaluated on the average of all the visited solutions [Schmidt et al., 2011, Honorio, 2012a, Atchade et al., 2014]. Theorem 2 is more practical than previous analysis, since  $\sum_{k=1}^{\kappa} \boldsymbol{\theta}^{(k)}/\kappa$  is a dense parameterization not applicable to structure learning.

According to the analysis above, we need to control  $\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})$  in each iteration to achieve a decreasing and  $O(\frac{1}{k})$ -converging objective function value sequence. Therefore, we focus on checkable bounds for gradient approximation error in Section 4.

## 4 MAIN RESULTS

In this section, we derive an asymptotic and a non-asymptotic bound to control the gradient approximation error  $\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})$  in each iteration. For this purpose, we consider an arbitrary  $\boldsymbol{\theta}$ , and perform gradient approximation via Gibbs- $\tau$  using Algorithm 2, given an initial value for the Gibbs sampling algorithm,  $\tilde{\mathbf{x}}_0$ . By bounding  $\boldsymbol{\delta}(\boldsymbol{\theta})$ , we can apply the same technique to address  $\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})$ .

We first provide a bound for the magnitude of the conditional expectation of  $\boldsymbol{\delta}(\boldsymbol{\theta})$ ,  $\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\boldsymbol{\delta}(\boldsymbol{\theta}) \mid \tilde{\mathbf{x}}_0]\|_2$ , in Section 4.1. Based on this result, we further draw a non-asymptotic bound for the magnitude of the gradient approximation error,  $\|\boldsymbol{\delta}(\boldsymbol{\theta})\|_2$ , in Section 4.2. Both results are *verifiable* in each iteration.

For the derivation of the conclusions, we will focus on binary pairwise Markov networks (BPMNs). Let  $\mathbf{x} \in \{0, 1\}^p$  and  $\boldsymbol{\theta}$  be given, a binary pairwise Markov network [Höfling and Tibshirani, 2009, Geng et al., 2017] is defined as:

$$P_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{Z(\boldsymbol{\theta})} \exp \left( \sum_{i=1}^p \sum_{j \geq i}^p \theta_{ij} x_i x_j \right), \quad (10)$$

where  $Z(\boldsymbol{\theta}) = \exp(A(\boldsymbol{\theta}))$  is the partition function.

$\theta_{ij}$  is a component of  $\boldsymbol{\theta}$  that represents the strength of conditional dependence between  $X_i$  and  $X_j$ .

#### 4.1 An Asymptotic Bound

We first consider the magnitude of the conditional expectation of  $\boldsymbol{\delta}(\boldsymbol{\theta})$  with respect to  $\tilde{\mathbf{x}}_\tau$ ,  $\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\boldsymbol{\delta}(\boldsymbol{\theta}) \mid \tilde{\mathbf{x}}_0]\|_2$ . To this end, we define  $\mathbf{U}$  a  $p \times p$  computable matrix that is related to  $\boldsymbol{\theta}$  and the type of MRF in question.  $U_{ij}$ , the component in the  $i^{\text{th}}$  row and the  $j^{\text{th}}$  column of  $\mathbf{U}$ , is defined as follows:

$$U_{ij} = \frac{|\exp(-\xi_{ij}) - 1|b^*}{(1 + b^* \exp(-\xi_{ij}))(1 + b^*)}, \quad (11)$$

where

$$b^* = \max \left\{ r, \min \left\{ s, \exp \left( \frac{\xi_{ij}}{2} \right) \right\} \right\},$$

$$s = \exp \left( -\xi_{ii} - \sum_{k \neq i, k \neq j} \xi_{ik} \max \{ -\text{sgn}(\xi_{ik}), 0 \} \right)$$

$$r = \exp \left( -\theta_{ii} - \sum_{k \neq i, k \neq j} \xi_{i,k} \max \{ \text{sgn}(\xi_{i,k}), 0 \} \right),$$

and  $\text{sgn}(\xi_{ik})$  is the sign function evaluated on  $\xi_{ij} = \theta_{\min\{i,j\}, \max\{i,j\}}$ .

We then define  $\mathbf{B}_i$  as a  $p \times p$  identity matrix except that its  $i^{\text{th}}$  row is replaced by the  $i^{\text{th}}$  row of  $\mathbf{U}$ , with  $i \in \{1, 2, \dots, p\}$ . We further define

$$\mathbf{B} = \mathbf{B}_p \mathbf{B}_{p-1} \mathbf{B}_{p-2} \cdots \mathbf{B}_i \cdots \mathbf{B}_1$$

and the grand sum  $\mathcal{G}(\mathbf{B}) = \sum_{i=1}^p \sum_{j=1}^p B_{ij}$ , where  $B_{ij}$  is the entry in the  $i^{\text{th}}$  row and the  $j^{\text{th}}$  column of  $\mathbf{B}$ . With the definitions above,  $\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\boldsymbol{\delta}(\boldsymbol{\theta}) \mid \tilde{\mathbf{x}}_0]\|_2$  can be upper bounded by Theorem 3.

**Theorem 3.** Let  $\tilde{\mathbf{x}}_\tau$  be the sample generated after running Gibbs sampling for  $\tau$  steps (Gibbs- $\tau$ ) under the parameterization  $\boldsymbol{\theta}$  initialized by  $\tilde{\mathbf{x}}_0 \in \{0, 1\}^p$ ; then with  $m$  denoting the size of sufficient statistics, the following inequality holds:

$$\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\boldsymbol{\delta}(\boldsymbol{\theta}) \mid \tilde{\mathbf{x}}_0]\|_2 \leq 2\sqrt{m}\mathcal{G}(\mathbf{B}^\tau), \quad (12)$$

where  $\mathbf{B}^\tau$  represents the  $\tau^{\text{th}}$  power of  $\mathbf{B}$ .

In Theorem 3, the bound provided is not only observable in each iteration, but also efficient to compute, offering a convenient method to inspect the quality of the gradient approximation. When the spectral norm of  $\mathbf{U}$  is less than 1, the left hand side of (12) will converge to 0. Thus, by increasing  $\tau$ , we can decrease  $\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\boldsymbol{\delta}(\boldsymbol{\theta}) \mid \tilde{\mathbf{x}}_0]\|_2$  to an arbitrarily small value.

Theorem 3 is derived by bounding the influence of a variable on another variable in  $\mathbf{X}$  (i.e., the Dobrushin influence defined in 2) with  $\mathbf{U}$ . Furthermore,  $\mathbf{U}$  defined in (11) is a sharp bound of the Dobrushin influence whenever  $b^* \neq \exp\left(\frac{\xi_{ij}}{2}\right)$ , explaining why (12) using the definition of  $\mathbf{U}$  is tight enough for practical applications.

#### 4.2 A Non-Asymptotic Bound

In order to provide a non-asymptotic guarantee for the quality of the gradient approximation, we need to concentrate  $\|\boldsymbol{\delta}(\boldsymbol{\theta})\|_2$  around  $\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\boldsymbol{\delta}(\boldsymbol{\theta}) \mid \tilde{\mathbf{x}}_0]\|_2$ . Let  $q$  defined in Section 2.2 be given. Then,  $q$  trials of Gibbs sampling are run, resulting in  $q$  samples,  $\{\tilde{\mathbf{x}}_\tau^{(1)}, \tilde{\mathbf{x}}_\tau^{(2)}, \dots, \tilde{\mathbf{x}}_\tau^{(q)}\}$ . That is to say, for each sufficient statistic,  $\psi_j(\boldsymbol{\theta})$ , with  $j \in \{1, 2, \dots, m\}$ , we have  $q$  samples,  $\{\psi_j^{(1)}(\boldsymbol{\theta}), \psi_j^{(2)}(\boldsymbol{\theta}), \dots, \psi_j^{(q)}(\boldsymbol{\theta})\}$ . Defining the sample variance of the corresponding sufficient statistics as  $V_{\psi_j}$ , we have Theorem 4 to provide a non-asymptotic bound for  $\|\boldsymbol{\delta}(\boldsymbol{\theta})\|_2$ :

**Theorem 4.** Let  $\boldsymbol{\theta}$ ,  $q$ , and an arbitrary  $\tilde{\mathbf{x}}_0 \in \{0, 1\}^p$  be given. Let  $m$  represent the dimension of  $\boldsymbol{\theta}$  and  $\|\boldsymbol{\delta}(\boldsymbol{\theta})\|_2$  represent the magnitude of the gradient approximation error by running  $q$  trials of Gibbs- $\tau$  initialized by  $\tilde{\mathbf{x}}_0$ . Compute  $\mathbf{B}$  according to Section 4.1 and choose  $\epsilon_j > 0$ . Then, with probability at least  $1 - 2\sum_{j=1}^m \beta_j$ , where  $\beta_j > 0$ ,  $j \in \{1, 2, \dots, m\}$ ,

$$\|\boldsymbol{\delta}(\boldsymbol{\theta})\|_2 \leq 2\sqrt{m} \left( \mathcal{G}(\mathbf{B}^\tau) + \sqrt{\frac{\sum_{j=1}^m \epsilon_j^2}{4m}} \right), \quad (13)$$

with  $\beta_j$  satisfying

$$\epsilon_j = 2 \left( \sqrt{\frac{V_{\psi_j} \ln 2 / \beta_j}{2q}} + \frac{7 \ln 2 / \beta_j}{3(q-1)} \right). \quad (14)$$

Notice that the bound in Theorem 4 is easily *checkable*, i.e., given  $\tau$ ,  $q$ ,  $V_{\psi_j}$ 's, and  $\boldsymbol{\theta}$ , we can determine a bound for  $\|\boldsymbol{\delta}(\boldsymbol{\theta})\|_2$  that holds with high probability. Furthermore, Theorem 4 provides the sample complexity needed for gradient estimation. Specifically, given small enough  $\beta_j$ 's, if we let

$$\mathcal{G}(\mathbf{B}^\tau) = \sqrt{\sum_{j=1}^m \epsilon_j^2 / 4m},$$

we can show that

$$2\sqrt{m} \left( \mathcal{G}(\mathbf{B}^\tau) + \sqrt{\sum_{j=1}^m \epsilon_j^2 / 4m} \right) = O\left(\frac{1}{q}\right).$$

That is to say, by assuming that  $\mathcal{G}(\mathbf{B}^\tau)$  and  $\sqrt{\sum_{j=1}^m \epsilon_j^2 / 4m}$  share the same scale, the upper bound of the gradient approximation error converges to 0 as  $q$  increases. Moreover, we include sample variance,  $V_{\psi_j}$ 's, in (13). This is because the information provided by sample variance leads to an improved data dependent bound.

## 5 PROOF SKETCH OF MAIN RESULTS

As mentioned in Section sec:non-asy-bound, the non-asymptotic result in Theorem 4 is derived from the asymptotic bound in Theorem 3 by concentration inequalities, we therefore only highlight the proof of Theorem 3 in this section, and defer other technical results to Supplements. Specifically, the proof of Theorem 3 is divided into two parts: bounding  $\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\delta(\theta) | \tilde{\mathbf{x}}_0]\|_2$  by the total variation distance (Section 5.1) and bounding the total variation distance (Section 5.2).

### 5.1 Bounding $\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\delta(\theta) | \tilde{\mathbf{x}}_0]\|_2$ by the Total Variation Distance

To quantify  $\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\delta(\theta) | \tilde{\mathbf{x}}_0]\|_2$ , we first introduce the concept of total variation distance [Levin et al., 2009] that measures the distance between two distributions over  $\{0, 1\}^p$ .

**Definition 1.** Let  $u(\mathbf{x})$ , and  $v(\mathbf{x})$  be two probability distributions of  $\mathbf{x} \in \{0, 1\}^p$ . Then the total variation distance between  $u(\mathbf{x})$  and  $v(\mathbf{x})$  is given as:

$$\|u(\mathbf{x}) - v(\mathbf{x})\|_{\text{TV}} = \frac{1}{2} \sum_{\mathbf{x} \in \{0, 1\}^p} |u(\mathbf{x}) - v(\mathbf{x})|.$$

With the definition above,  $\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\delta(\theta) | \tilde{\mathbf{x}}_0]\|_2$  can be upper bounded by the total variation distance between two distributions ( $P_\tau(\mathbf{x} | \tilde{\mathbf{x}}_0)$  and  $P_\theta(\mathbf{x})$ ) using the following lemma:

**Lemma 1.** Let  $\tilde{\mathbf{x}}_\tau$  be the sample generated after running Gibbs sampling for  $\tau$  steps (Gibbs- $\tau$ ) under the parameterization  $\theta$  initialized by  $\tilde{\mathbf{x}}_0 \in \{0, 1\}^p$ , then the following is true:

$$\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\delta(\theta) | \tilde{\mathbf{x}}_0]\|_2 \leq 2\sqrt{m} \|P_\tau(\mathbf{x} | \tilde{\mathbf{x}}_0) - P_\theta(\mathbf{x})\|_{\text{TV}}.$$

With Lemma 1, bounding  $\|\mathbb{E}_{\tilde{\mathbf{x}}_\tau}[\delta(\theta) | \tilde{\mathbf{x}}_0]\|_2$  can be achieved by bounding the total variation distance  $\|P_\tau(\mathbf{x} | \tilde{\mathbf{x}}_0) - P_\theta(\mathbf{x})\|_{\text{TV}}$ . Recent advances in the quality control of Gibbs samplers offer us emphverifiable upper bounds for  $\|P_\tau(\mathbf{x} | \tilde{\mathbf{x}}_0) - P_\theta(\mathbf{x})\|_{\text{TV}}$  on the learning of a variety of MRFs [Mitliagkas and Mackey,

2017]. However, they can not be applied to BPMNs because of the positive constraint on parameters. We describe these next.

### 5.2 Bounding $\|P_\tau(\mathbf{x} | \tilde{\mathbf{x}}_0) - P_\theta(\mathbf{x})\|_{\text{TV}}$

Now we generalize the analysis in Mitliagkas and Mackey [2017] to BPMNs without constraints on the sign of parameters by introducing the definition of the Dobrushin influence matrix and a technical lemma.

**Definition 2** (Dobrushin influence matrix). The Dobrushin influence matrix of  $P_\theta(\mathbf{x})$  is a  $p \times p$  matrix  $\mathbf{C}$  with its component in the  $i^{\text{th}}$  row and the  $j^{\text{th}}$  column,  $C_{ij}$ , representing the influence of  $X_j$  on  $X_i$  given as:

$$C_{ij} = \max_{(\mathbf{X}, \mathbf{Y}) \in N_j} \|P_\theta(X_i | \mathbf{X}_{-i}) - P_\theta(Y_i | \mathbf{Y}_{-i})\|_{\text{TV}},$$

where  $(\mathbf{X}, \mathbf{Y}) \in N_j$  represents  $X_l = Y_l$  for all  $l \neq j$ .

**Lemma 2.** Let  $P_\theta(\mathbf{x})$  represent a binary pairwise Markov network defined in (10) that is parameterized by  $\theta$ . An upper bound of the total influence matrix is given by  $\mathbf{U}$  defined in Section 4.1.

It should be noticed that, similar to the Theorem 12 in Mitliagkas and Mackey [2017], Lemma 2 provides an exact calculation except when  $b^* = \exp\left(\frac{\xi_{i,j}}{2}\right)$ .

Therefore, we can consider the  $\mathbf{U}$  defined in Section 4.1 as an upper bound for Dobrushin influence matrix in BPMN and thus apply  $\mathbf{U}$  to Theorem 9 in Mitliagkas and Mackey [2017]. Then, we have

$$\|P_\tau(\mathbf{x} | \tilde{\mathbf{x}}_0) - P_\theta(\mathbf{x})\|_{\text{TV}} \leq \mathcal{G}(\mathbf{B}^\tau),$$

where  $\mathbf{B}^\tau$  represents the  $\tau^{\text{th}}$  power of  $\mathbf{B}$ . Theorem 3 follows this combined with Lemma 1

## 6 STRUCTURE LEARNING

With the two bounds introduced in Section 4, we can easily examine and control the quality of gradient approximation in each iteration by choosing  $\tau$ . In detail, we introduce a criterion for the selection of  $\tau$  in each iteration. Satisfying the proposed criterion, the objective function is guaranteed to decrease asymptotically. That is to say, the difference between  $g(\theta^{(k+1)})$  and  $g(\hat{\theta})$  is asymptotically *tightened*, compared with the difference between  $g(\theta^{(k)})$  and  $g(\hat{\theta})$ . Therefore, we refer to the proposed criterion as TAY-CRITERION. Furthermore, using TAY-CRITERION we provide an improved SPG method denoted by TAY for short.

Specifically, starting from  $\tau = 1$ , TAY stops increasing  $\tau$  when the following is satisfied:

$$2\sqrt{m}\mathcal{G}(\mathbf{B}^\tau) < \frac{1}{2}\|\mathbf{G}_\alpha(\theta^{(k)})\|_2. \quad (\text{TAY-CRITERION})$$

We can also derive a non-asymptotic counterpart of TAY-CRITERION by combining the results of Theorem 1 and Theorem 4:

$$0 < 2\sqrt{m} \left( \mathcal{G}(\mathbf{B}^\tau) + \sqrt{\frac{\sum_{j=1}^m \epsilon_j^2}{4m}} \right) \leq \frac{1}{2} \|\mathbf{G}_\alpha(\boldsymbol{\theta}^{(k)})\|_2,$$

$$\epsilon_j = 2 \left( \sqrt{\frac{2V_{\psi_j} \ln 2 / \beta_j}{4q}} + \frac{7 \ln 2 / \beta_j}{3(q-1)} \right), \quad (15)$$

where the  $V_{\psi_j}$ 's and  $\beta_j$ 's are defined in Theorem 4. (15) provides the required sample complexity,  $q$ , for TAY in each iteration. However, the selection of  $q$  according to (15) is conservative, because it includes the worst-case scenario where the gradient approximation errors in any two iterations cannot offset each other.

In Section 6.1 and 6.2, we theoretically analyze the performance guarantees of TAY-CRITERION and the convergence of TAY, respectively.

## 6.1 Guarantees of TAY-CRITERION

The theorem below provides the performance guarantee for TAY-CRITERION in each iteration.

**Theorem 5.** Let  $\boldsymbol{\theta}^{(k)}$  and  $\tilde{\mathbf{x}}_0$  be given. Let  $q$  and  $\mathbf{B}$  defined in Theorem 4 be given. For  $\boldsymbol{\theta}^{(k+1)}$  generated in Algorithm 3 using TAY-CRITERION, the following is true:

$$\lim_{q \rightarrow \infty} \mathbb{P} \left( g(\boldsymbol{\theta}^{(k+1)}) < g(\boldsymbol{\theta}^{(k)}) \mid 2\sqrt{m} \mathcal{G}(\mathbf{B}^\tau) < \frac{1}{2} \|\mathbf{G}_\alpha(\boldsymbol{\theta}^{(k)})\|_2 \right) = 1.$$

Theorem 5 makes a statement that the objective function value decreases with large  $q$ . Specifically, TAY-CRITERION assumes that the upper bound of the conditional expectation of  $\|\boldsymbol{\delta}(\boldsymbol{\theta})\|_2$  is small enough to satisfy the sufficient condition proven in Theorem 1. When the number of samples  $q$  is large enough,  $\|\boldsymbol{\delta}(\boldsymbol{\theta})\|_2$  itself is very likely to meet the condition and hence the objective function is also likely to decrease with TAY-CRITERION satisfied.

## 6.2 Convergence of TAY

Finally, based on Theorem 2 and Theorem 5, we derive the following theorem on the convergence of TAY.

**Theorem 6.** Let  $\mathcal{K} = (\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots, \boldsymbol{\theta}^{(\kappa)})$  be the iterates generated by TAY. Then, with  $k \in \{1, 2, \dots, \kappa - 1\}$ , the following is true:

$$\lim_{q \rightarrow \infty} \mathbb{P} \left[ g(\boldsymbol{\theta}^{(\kappa)}) - g(\hat{\boldsymbol{\theta}}) \leq \frac{L}{2\kappa} \left( \|\boldsymbol{\theta}^{(0)} - \hat{\boldsymbol{\theta}}\|_2 + \frac{2}{L} \sum_{k=1}^{\kappa} \|\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})\|_2 \right)^2 \right] = 1,$$

where  $\hat{\boldsymbol{\theta}}$  is defined in (1).

## 6.3 Generalizations

As we demonstrate in Section 4 and Section 5, the derivation of our main results relies on bounding the Dobrushin influence with  $\mathbf{U}$  and we show a procedure to construct  $\mathbf{U}$  in the context of BPMNs. Moreover, Mitliagkas and Mackey [2017] and Liu and Domke [2014] provide upper bounds  $\mathbf{U}$ 's for other types of discrete pairwise MRFs. Therefore, combined with their results, our framework can also be applied to other discrete pairwise Markov networks. Dealing with pairwise MRFs is without any loss of generality, since any discrete MRF can be transformed into a pairwise one [Wainwright et al., 2008, Ravikumar et al., 2010].

## 7 EXPERIMENTS

We demonstrate that the structure learning of discrete MRFs benefits substantially from the application of TAY with synthetic data and that the bound provided on the gradient estimation error by Theorem 3 is tighter than existing bounds. To illustrate that TAY is readily available for practical problems, we also run TAY using a real world dataset. Because of the limit of space, we only report the experiments under one set of representative experiment configurations. Exhaustive results using different experiment configurations are presented in the Supplements.

### 7.1 Structure Learning

In order to demonstrate the utility of TAY for effectively learning the structures of BPMNs, we simulate two BPMNs (one with 10 nodes and the other one with 20 nodes):

- We set the number of features to  $p = 10$  ( $p = 20$ ). Components of  $\boldsymbol{\theta}$  in the ground truth model are randomly chosen to be nonzero with an edge generation probability of 0.3. The non-zero components of the real parameter have a uniform distribution on  $[-2, -1] \cup [1, 2]$
- 1000 (2000 for 20 nodes) samples are generated by Gibbs sampling with 1000 burn-in steps.

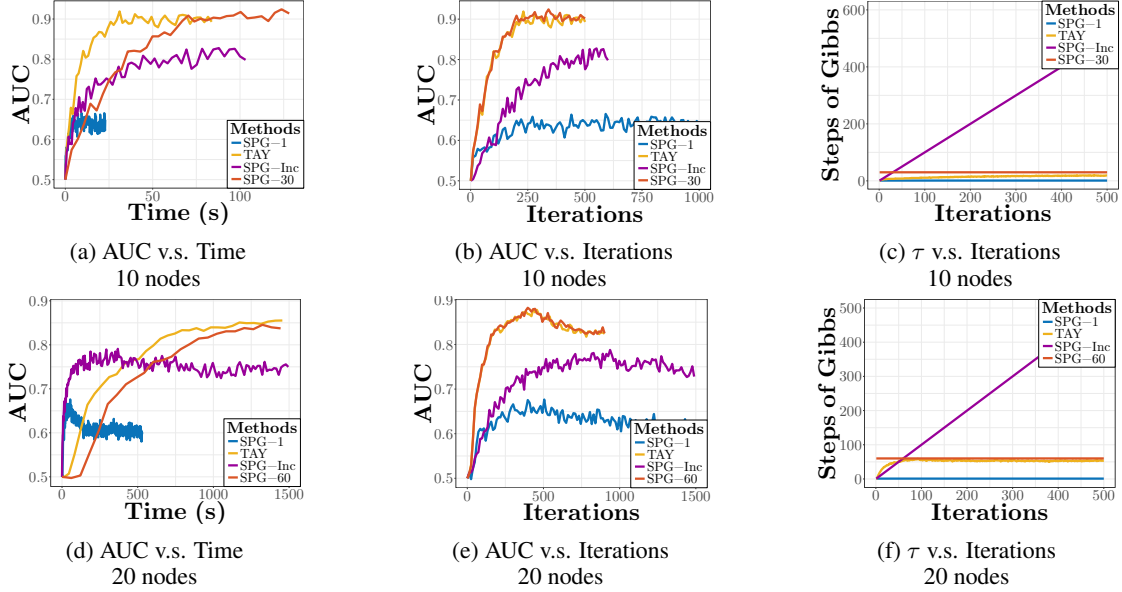


Figure 1: Area under curve (AUC) and the steps of Gibbs sampling ( $\tau$ ) in each iteration for structure learning of a 20-node network.

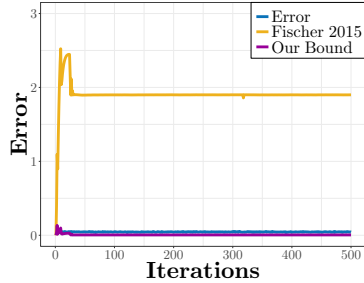


Figure 2: The gradient approximation error, the existing bound and the bound (12) in the structure learning of a 10-node network.

- The results are averaged over 10 trials.

The sizes of the BPMNs generated in this paper are comparable to those in [Honorio, 2012a, Atchade et al., 2014, Miasojedow and Rejchel, 2016].

Then, using the generated samples, we consider SPG and TAY. According to the analysis in Section 4, the quality of the gradient approximation is closely related to the number of Gibbs sampling steps  $\tau$ . However, for SPG, there are no convincing schemes for selecting  $\tau$ . Therefore, we consider a large enough  $\tau = 30$  ( $\tau = 60$  for 20 nodes) to make sure that the gradient approximation error is small enough. Furthermore, we also evaluate the performance of the algorithm using an increasing  $\tau$  ( $\tau = k$  in the  $k^{\text{th}}$  iteration), suggested by Atchade et al. [2014] (SPG-Inc).

To strike a fair comparison, we use the same step length  $\alpha = 0.4$  and regularization parameter  $\lambda = 0.025$  ( $\lambda = 0.017$  for 20 nodes) for different methods. We do not tune the step length individually for each method, since Atchade et al. [2014] has shown that various learning rate selection schemes have minimal impact on the performance in the context of SPG. The number of chains used in Gibbs sampling,  $q$ , is not typically a tunable parameter either, since it indicates the allocation of the computational resources. For each method, it can be easily noticed that the larger the number of samples is the slower but more accurate the method will be. Furthermore, if the  $q$ 's are different for different methods, it would be difficult to distinguish the effect of  $\tau$  from that of  $q$ . Therefore, we set it to 2000 for 10-node networks and 5000 for 20-node networks. Performances of different methods are compared using the area under curve (AUC) of receiver operating characteristic (ROC) for structure learning in Figure 1. The Gibbs sampling steps in each method are also compared in Figure 1.

Notice that we plot AUCs against both time (Figure 1a and Figure 1d) and iterations (Figure 1b and Figure 1e). The two kinds of plots provide different information about the performances of different methods: the former ones focus on overall complexity and the latter illustrate iteration complexity. We run each method until it converges. Using much less time, TAY achieves a similar AUC to SPG with  $\tau = 30$  and  $\tau = 60$ . Moreover, SPG with  $\tau = 1$  reaches the lowest AUC, since the quality of the gradient approximation cannot be guaranteed with such a small  $\tau$ . Therefore, the experimental results



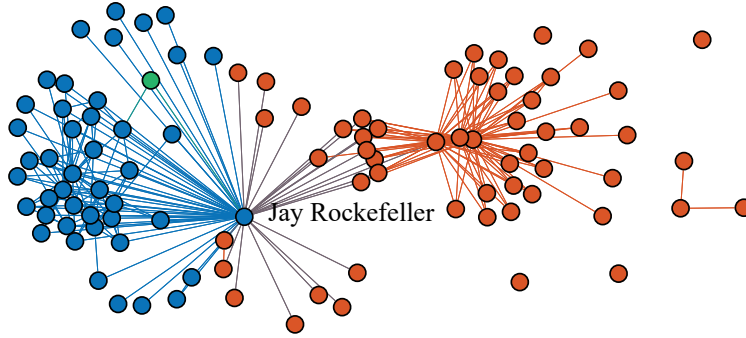


Figure 3: The result of TAY on the senator voting data: Red vertices denote Republicans, blue Democrats, and green Independent. The figure is rendered by Gephi [Bastian et al., 2009].

indicate that TAY adaptively chooses a  $\tau$  achieving reasonable accuracy as well as efficiency for structure learning in each iteration. For a more thorough comparison, we also contrast the performance of TAY and a non-SPG-based method, i.e., the pseudo-likelihood method [Höfling and Tibshirani, 2009, Geng et al., 2017], in the Supplements. As a result, the two methods achieve comparable AUCs.

## 7.2 Tightness of the Proposed Bound

According to the empirical results above, TAY needs a  $\tau$  only on the order of ten, suggesting that the bound in Theorem 3 is tight enough for practical applications. To illustrate this more clearly, we compare (12) with another bound on the expectation of the gradient approximation error derived by Fischer [2015]. Specifically, we calculate the gradient approximation error, the bound (12), and Fischer [2015]’s bound, in each iteration of learning a 10-node network. The results are reported in Figure 2. Notice that the bound in Fischer [2015] gets extraordinarily loose with more iterations. Considering this, we may need run Gibbs chains for thousands of steps if we use this bound. In contrast, bound (12) is close to and even slightly less than the real error. This is reflective of the fact that the proposed bound is on the expectation instead of the error itself. As a result, (12) is much tighter and thus more applicable.

## 7.3 Real World Data

In our final experiment, we run TAY using the Senate voting data from the second session of the 109<sup>th</sup> Congress [USS]. The dataset has 279 samples and 100 variables. Each sample represents the vote cast by each of the 100 senators for a particular bill, where 0 represents nay, and 1 represents yea. Missing data are imputed as 0’s. The task of interest is to learn a

BPMN model that identifies some clusters that represent the dependency between the voting inclination of each senator and the party with which the senator is affiliated.

We use TAY with  $\alpha = 0.4$ . 5000 Markov chains are used for Gibbs sampling. Since our task is exploratory analysis,  $\lambda = 0.1$  is selected in order to deliver an interpretable result. The proposed algorithm is run for 100 iterations. The resultant BPMN with only edges corresponding to the positive parameters is shown in Figure 3, where each node represents the voting record of a senator and the edges represent some positive dependency between the pair of senators connected. The nodes in red represent Republicans and the nodes in blue represents Democrats. The clustering effects of voting consistency within a party are captured, coinciding with conventional wisdom. More interestingly, Jay Rockefeller, as a Democrat, has many connections with Republicans. This is consistent with the fact that his family has been a “traditionally Republican dynasty” [Wikipedia, 2017].

## 8 CONCLUSION

We consider SPG for  $L_1$ -regularized discrete MRF estimation. Furthermore, we conduct a careful analysis of the gradient approximation error of SPG and provide upper bounds to quantify its magnitude. With the aforementioned analysis, we introduce a learning strategy called TAY and show that it can improve the accuracy and efficiency of SPG.

**Acknowledgement:** Sinong Geng, Zhaobin Kuang, and David Page would like to gratefully acknowledge the NIH BD2K Initiative grant U54 AI117924 and the NIGMS grant 2R01 GM097618. Stephen Wright would like to gratefully acknowledge NSF Awards IIS-1447449, 1628384, 1634597, and 1740707; AFOSR Award FA9550-13-1-0138; Subcontracts 3F-30222 and 8F-30039 from Argonne National Laboratory; and DARPA Award N660011824020.

## References

- U.S. Senate. <http://www.senate.gov/index.htm>. (Accessed on 10/11/2016).
- Y. F. Atchade, G. Fort, and E. Moulines. On stochastic proximal gradient algorithms. 2014.
- M. Bastian, S. Heymann, M. Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009.
- Y. Bengio and O. Delalleau. Justifying and generalizing contrastive divergence. *Neural computation*, 21(6):1601–1621, 2009.
- A. Fischer. Training restricted boltzmann machines. *KI-Künstliche Intelligenz*, 29(4):441–444, 2015.
- A. Fischer and C. Igel. Bounding the bias of contrastive divergence learning. *Neural computation*, 23(3):664–673, 2011.
- S. Geng, Z. Kuang, and D. Page. An efficient pseudo-likelihood method for sparse binary pairwise Markov network estimation. *arXiv preprint arXiv:1702.08320*, 2017.
- S. Geng, Z. Kuang, P. Peissig, and D. Page. Temporal square root graphical models. In *Proceedings of the Thirty-Fifth International Conference on Machine Learning ( 2018 )*, 2018.
- H. Höfling and R. Tibshirani. Estimation of sparse binary pairwise Markov networks using pseudo-likelihoods. *Journal of Machine Learning Research*, 10(Apr):883–906, 2009.
- J. Honorio. Convergence rates of biased stochastic optimization for learning sparse ising models. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 257–264, New York, NY, USA, July 2012a. Omnipress. ISBN 978-1-4503-1285-1.
- J. Honorio. Lipschitz parametrization of probabilistic graphical models. *arXiv preprint arXiv:1202.3733*, 2012b.
- D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using  $l_1$ -regularization. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pages 817–824. MIT Press, 2006.
- D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov chains and mixing times*. American Mathematical Soc., 2009.
- J. Liu and D. Page. Bayesian estimation of latently-grouped parameters in undirected graphical models. In *Advances in Neural Information Processing Systems*, pages 1232–1240, 2013.
- J. Liu, D. Page, H. Nassif, J. Shavlik, P. Peissig, C. McCarty, A. A. Onitilo, and E. Burnside. Genetic variants improve breast cancer risk prediction on mammograms. In *AMIA Annual Symposium Proceedings*, volume 2013, page 876. American Medical Informatics Association, 2013.
- J. Liu, D. Page, P. Peissig, C. McCarty, A. A. Onitilo, A. Trentham-Dietz, and E. Burnside. New genetic variants improve personalized breast cancer diagnosis. *AMIA Summits on Translational Science Proceedings*, 2014:83, 2014a.
- J. Liu, C. Zhang, E. Burnside, and D. Page. Learning heterogeneous hidden Markov random fields. In *Artificial Intelligence and Statistics*, pages 576–584, 2014b.
- J. Liu, C. Zhang, D. Page, et al. Multiple testing under dependence via graphical models. *The Annals of Applied Statistics*, 10(3):1699–1724, 2016.
- X. Liu and J. Domke. Projecting markov random field parameters for fast mixing. In *Advances in Neural Information Processing Systems*, pages 1377–1385, 2014.
- A. Maurer and M. Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.
- B. Miasojedow and W. Rejchel. Sparse estimation in ising model via penalized Monte Carlo methods. *arXiv preprint arXiv:1612.07497*, 2016.
- I. Mitliagkas and L. Mackey. Improving Gibbs sampler scan quality with DoGS, 2017.
- N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.
- P. Ravikumar, M. J. Wainwright, J. D. Lafferty, et al. High-dimensional ising model selection using  $l_1$ -regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.
- M. Schmidt, N. L. Roux, and F. R. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in neural information processing systems*, pages 1458–1466, 2011.
- L. Vandenberghe. Lecture notes in ee236c-optimization methods for large-scale systems (spring 2016), 2016.
- V. Viallon, O. Banerjee, E. Jouglu, G. Rey, and J. Coste. Empirical comparison study of approximate methods for structure selection in binary graphical models. *Biometrical Journal*, 56(2):307–331, 2014.
- M. J. Wainwright, P. Ravikumar, and J. D. Lafferty. High-dimensional graphical model selection using  $l_1$ -regularized logistic regression. *Advances in neural information processing systems*, 19:1465, 2007.
- M. J. Wainwright, M. I. Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.
- Wikipedia. Jay rockefeller—Wikipedia, the free encyclopedia, 2017. URL [https://en.wikipedia.org/wiki/Jay\\_Rockefeller](https://en.wikipedia.org/wiki/Jay_Rockefeller). (Accessed on 05/06/2017).

## Supplements

### A Proofs

#### A.1 Proof of Theorem 1

We first introduce the following technical lemma.

**Lemma 3.** Let  $g(\boldsymbol{\theta})$ ,  $f(\boldsymbol{\theta})$ , and  $h(\boldsymbol{\theta})$  be defined as in Section 2.1; hence  $f(\boldsymbol{\theta})$  is convex and differentiable, and  $\nabla f(\boldsymbol{\theta})$  is Lipschitz continuous with Lipschitz constant  $L$ . Let  $\alpha \leq 1/L$ . Let  $\mathbf{G}_\alpha(\boldsymbol{\theta})$  and  $\Delta f(\boldsymbol{\theta})$  be defined as in Section (2.2). Then for all  $\boldsymbol{\theta}_1$  and  $\boldsymbol{\theta}_2$ , the following inequality holds:

$$g(\boldsymbol{\theta}_1^\dagger) \leq g(\boldsymbol{\theta}_2) + \mathbf{G}_\alpha^\top(\boldsymbol{\theta}_1)(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) + (\nabla f(\boldsymbol{\theta}_1) - \Delta f(\boldsymbol{\theta}_1))^\top(\boldsymbol{\theta}_1^\dagger - \boldsymbol{\theta}_2) - \frac{\alpha}{2} \|\mathbf{G}_\alpha(\boldsymbol{\theta}_1)\|_2^2, \quad (16)$$

where  $\boldsymbol{\theta}_1^\dagger = \boldsymbol{\theta}_1 - \alpha \mathbf{G}_\alpha(\boldsymbol{\theta}_1)$ .

*Proof.* The proof is based on the convergence analysis of the standard proximal gradient method [Vandenberghe, 2016].  $f(\boldsymbol{\theta})$  is a convex differentiable function whose gradient is Lipschitz continuous with Lipschitz constant  $L$ . By the quadratic bound of the Lipschitz property:

$$f(\boldsymbol{\theta}_1^\dagger) \leq f(\boldsymbol{\theta}_1) - \alpha \nabla^\top f(\boldsymbol{\theta}_1) \mathbf{G}_\alpha(\boldsymbol{\theta}_1) + \frac{\alpha^2 L}{2} \|\mathbf{G}_\alpha(\boldsymbol{\theta}_1)\|_2^2.$$

With  $\alpha \leq 1/L$ , and adding  $h(\boldsymbol{\theta}_1^\dagger)$  on both sides of the quadratic bound, we have an upper bound for  $g(\boldsymbol{\theta}_1^\dagger)$ :

$$g(\boldsymbol{\theta}_1^\dagger) \leq f(\boldsymbol{\theta}_1) - \alpha \nabla^\top f(\boldsymbol{\theta}_1) \mathbf{G}_\alpha(\boldsymbol{\theta}_1) + \frac{\alpha}{2} \|\mathbf{G}_\alpha(\boldsymbol{\theta}_1)\|_2^2 + h(\boldsymbol{\theta}_1^\dagger).$$

By convexity of  $f(\boldsymbol{\theta})$  and  $h(\boldsymbol{\theta})$ , we have:

$$\begin{aligned} f(\boldsymbol{\theta}_1) &\leq f(\boldsymbol{\theta}_2) + \nabla^\top f(\boldsymbol{\theta}_1)(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2), \\ h(\boldsymbol{\theta}_1^\dagger) &\leq h(\boldsymbol{\theta}_2) + (\mathbf{G}_\alpha(\boldsymbol{\theta}_1) - \Delta f(\boldsymbol{\theta}_1))^\top(\boldsymbol{\theta}_1^\dagger - \boldsymbol{\theta}_2), \end{aligned}$$

which can be used to further upper bound  $g(\boldsymbol{\theta}_1^\dagger)$ , and results in (16). Note that we have used the fact that  $\mathbf{G}_\alpha(\boldsymbol{\theta}_1) - \Delta f(\boldsymbol{\theta}_1)$  is a subgradient of  $h(\boldsymbol{\theta}_1^\dagger)$  in the last inequality.  $\square$

With Lemma 3, we are now able to prove Theorem 1. In Lemma 3, let  $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \boldsymbol{\theta}^{(k)}$ . Then by (8),  $\boldsymbol{\theta}_1^\dagger = \boldsymbol{\theta}^{(k+1)}$ . The inequality in (16) can then be simplified as:

$$g(\boldsymbol{\theta}^{(k+1)}) - g(\boldsymbol{\theta}^{(k)}) \leq \alpha \boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})^\top \mathbf{G}_\alpha(\boldsymbol{\theta}^{(k)}) - \frac{\alpha}{2} \|\mathbf{G}_\alpha(\boldsymbol{\theta}^{(k)})\|_2^2.$$

By the Cauchy-Schwarz inequality and the sufficient condition that  $\|\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})\|_2 < \frac{1}{2} \|\mathbf{G}_\alpha(\boldsymbol{\theta}^{(k)})\|_2$ , we can further simplify the inequality and conclude  $g(\boldsymbol{\theta}^{(k+1)}) < g(\boldsymbol{\theta}^{(k)})$ .

#### A.2 Proof of Theorem 2

To prove Theorem 2, we first review Proposition 1 in Schmidt et al. [2011]:

**Theorem 7** (Convergence on Average, Schmidt et al. [2011]). Let  $\mathcal{K} = (\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots, \boldsymbol{\theta}^{(\kappa)})$  be the iterates generated by Algorithm 3, then

$$g\left(\frac{1}{\kappa} \sum_{k=1}^{\kappa} \boldsymbol{\theta}^{(k)}\right) - g(\hat{\boldsymbol{\theta}}) \leq \frac{L}{2\kappa} \left( \|\boldsymbol{\theta}^{(0)} - \hat{\boldsymbol{\theta}}\|_2 + \frac{2}{L} \sum_{k=1}^{\kappa} \|\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})\|_2 \right)^2.$$

Furthermore, according to the assumption that  $g(\boldsymbol{\theta}^{(k+1)}) \leq g(\boldsymbol{\theta}^{(k)})$  with  $k \in \{1, 2, \dots, \kappa\}$ , we have:  $g\left(\frac{1}{\kappa} \sum_{k=1}^{\kappa} \boldsymbol{\theta}^{(k)}\right) \geq g(\boldsymbol{\theta}^{(\kappa)})$ . Therefore,

$$g(\boldsymbol{\theta}^{(\kappa)}) - g(\hat{\boldsymbol{\theta}}) \leq \frac{L}{2\kappa} \left( \|\boldsymbol{\theta}^{(0)} - \hat{\boldsymbol{\theta}}\|_2 + \frac{2}{L} \sum_{k=1}^{\kappa} \|\boldsymbol{\delta}(\boldsymbol{\theta}^{(k)})\|_2 \right)^2.$$