On Learning Discrete Graphical Models Using Greedy Methods

Ali Jalali
University of Texas at Austin
alij@mail.utexas.edu

Christopher C. Johnson
University of Texas at Asutin
cjohnson@cs.utexas.edu

Pradeep Ravikumar
University of Texas at Asutin
pradeepr@cs.utexas.edu

Abstract

In this paper, we address the problem of learning the structure of a pairwise graphical model from samples in a high-dimensional setting. Our first main result studies the sparsistency, or consistency in sparsity pattern recovery, properties of a forward-backward greedy algorithm as applied to general statistical models. As a special case, we then apply this algorithm to learn the structure of a discrete graphical model via neighborhood estimation. As a corollary of our general result, we derive sufficient conditions on the number of samples n, the maximum node-degree d and the problem size p, as well as other conditions on the model parameters, so that the algorithm recovers all the edges with high probability. Our result guarantees graph selection for samples scaling as $n = \Omega(d^2 \log(p))$, in contrast to existing convex-optimization based algorithms that require a sample complexity of $\Omega(d^3 \log(p))$. Further, the greedy algorithm only requires a restricted strong convexity condition which is typically milder than irrepresentability assumptions. We corroborate these results using numerical simulations at the end.

1 Introduction

Undirected graphical models, also known as Markov random fields, are used in a variety of domains, including statistical physics, natural language processing and image analysis among others. In this paper we are concerned with the task of estimating the graph structure G of a Markov random field (MRF) over a discrete random vector $X = (X_1, X_2, \ldots, X_p)$, given n independent and identically distributed samples $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$. This underlying graph structure encodes conditional independence assumptions among subsets of the variables, and thus plays an important role in a broad range of applications of MRFs.

Existing approaches: Neighborhood Estimation, Greedy Local Search. Methods for estimating such graph structure include those based on constraint and hypothesis testing [22], and those that estimate restricted classes of graph structures such as trees [8], polytrees [11], and hypertrees [23]. A recent class of successful approaches for graphical model structure learning are based on estimating the local neighborhood of each node. One subclass of these for the special case of bounded degree graphs involve the use of exhaustive search so that their computational complexity grows at least as quickly as $\mathcal{O}(p^d)$, where d is the maximum neighborhood size in the graphical model [1, 4, 9]. Another subclass use convex programs to learn the neighborhood structure: for instance [20, 17, 16] estimate the neighborhood set for each vertex $r \in V$ by optimizing its ℓ_1-regularized conditional likelihood; [15, 10] use ℓ_1/ℓ_2-regularized conditional likelihood. Even these methods, however need to solve regularized convex programs with typically polynomial computational cost of $O(p^4)$ or $O(p^6)$, are still expensive for large problems. Another popular class of approaches are based on using a score metric and searching for the best scoring structure from a candidate set of graph structures. Ex-

act search is typically NP-hard [7]; indeed for general discrete MRFs, not only is the search space intractably large, but calculation of typical score metrics itself is computationally intractable since they involve computing the partition function associated with the Markov random field [26]. Such methods thus have to use approximations and search heuristics for tractable computation. *Question:* Can one use local procedures that are as inexpensive as the heuristic greedy approaches, and yet come with the strong statistical guarantees of the regularized convex program based approaches?

High-dimensional Estimation; Greedy Methods. There has been an increasing focus in recent years on high-dimensional statistical models where the number of parameters p is comparable to or even larger than the number of observations n. It is now well understood that consistent estimation is possible even under such high-dimensional scaling if some low-dimensional structure is imposed on the model space. Of relevance to graphical model structure learning is the structure of sparsity, where a sparse set of non-zero parameters entail a sparse set of edges. A surge of recent work [5, 12] has shown that ℓ_1-regularization for learning such sparse models can lead to practical algorithms with strong theoretical guarantees. A line of recent work (cf. paragraph above) has thus leveraged this sparsity inducing nature of ℓ_1-regularization, to propose and analyze convex programs based on regularized log-likelihood functions. A related line of recent work on learning sparse models has focused on "stagewise" greedy algorithms. These perform simple forward steps (adding parameters greedily), and possibly also backward steps (removing parameters greedily), and yet provide strong statistical guarantees for the estimate after a finite number of greedy steps. The forward greedy variant which performs just the forward step has appeared in various guises in multiple communities: in machine learning as boosting [13], in function approximation [24], and in signal processing as basis pursuit [6]. In the context of statistical model estimation, Zhang [28] analyzed the forward greedy algorithm for the case of sparse linear regression; and showed that the forward greedy algorithm is sparsistent (consistent for model selection recovery) under the same "irrepresentable" condition as that required for "sparsistency" of the Lasso. Zhang [27] analyzes a more general greedy algorithm for sparse linear regression that performs forward and backward steps, and showed that it is sparsistent under a weaker restricted eigenvalue condition. Here we ask the question: *Can we provide an analysis of a general forward backward algorithm for parameter estimation in general statistical models?* Specifically, we need to extend the sparsistency analysis of [28] to general non-linear models, which requires a subtler analysis due to the circular requirement of requiring to control the third order terms in the Taylor series expansion of the log-likelihood, that in turn requires the estimate to be well-behaved. Such extensions in the case of ℓ_1-regularization occur for instance in [20, 25, 3].

Our Contributions. In this paper, we address both questions above. In the first part, we analyze the forward backward greedy algorithm [28] for general statistical models. We note that even though we consider the general statistical model case, our analysis is much simpler and accessible than [28], and would be of use even to a reader interested in just the linear model case of Zhang [28]. In the second part, we use this to show that when combined with neighborhood estimation, the forward backward variant applied to local conditional log-likelihoods provides a simple computationally tractable method that adds and deletes edges, but comes with strong *sparsistency* guarantees. We reiterate that the our first result on the sparsistency of the forward backward greedy algorithm for general objectives is of independent interest even outside the context of graphical models. As we show, the greedy method is better than the ℓ_1-regularized counterpart in [20] theoretically, as well as experimentally. The sufficient condition on the parameters imposed by the greedy algorithm is a restricted strong convexity condition [19], which is weaker than the irrepresentable condition required by [20]. Further, the number of samples required for sparsistent graph recovery scales as $O(d^2 \log p)$, where d is the maximum node degree, in contrast to $O(d^3 \log p)$ for the ℓ_1-regularized counterpart. We corroborate this in our simulations, where we find that the greedy algorithm requires fewer observations than [20] for sparsistent graph recovery.

2 Review, Setup and Notation

2.1 Markov Random Fields

Let $X = (X_1, \ldots, X_p)$ be a random vector, each variable X_i taking values in a discrete set $\mathcal{X}$ of cardinality m. Let $G = (V, E)$ denote a graph with p nodes, corresponding to the p variables $\{X_1, \ldots, X_p\}$. A pairwise Markov random field over $X = (X_1, \ldots, X_p)$ is then specified by nodewise and pairwise functions $\theta_r : \mathcal{X} \mapsto \mathbb{R}$ for all $r \in V$, and $\theta_{rt} : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ for all $(r, t) \in E$:

$$\mathbb{P}(x) \propto \exp \Big\{ \sum_{r \in V} \theta_r(x_r) + \sum_{(r,t) \in E} \theta_{rt}(x_r, x_t) \Big\}. \tag{1}$$

In this paper, we largely focus on the case where the variables are binary with $\mathcal{X} = \{-1, +1\}$, where we can rewrite (1) to the Ising model form [14] for some set of parameters $\{\theta_r\}$ and $\{\theta_{rt}\}$ as

$$\mathbb{P}(x) \propto \exp\Big\{ \sum_{r \in V} \theta_r x_r + \sum_{(r,t) \in E} \theta_{rt} x_r x_t \Big\}. \tag{2}$$

2.2 Graphical Model Selection

Let $D := \{x^{(1)}, \ldots, x^{(n)}\}$ denote the set of n samples, where each p-dimensional vector $x^{(i)} \in \{1, \ldots, m\}^p$ is drawn i.i.d. from a distribution $\mathbb{P}_{\theta^*}$ of the form (1), for parameters θ^* and graph $G = (V, E^*)$ over the p variables. Note that the true edge set E^* can also be expressed as a function of the parameters as

$$E^* = \{(r,t) \in V \times V : \theta_{st}^* \neq 0\}. \tag{3}$$

The *graphical model selection* task consists of inferring this edge set E^* from the samples D. The goal is to construct an estimator $\hat{E}_n$ for which $\mathbb{P}[\hat{E}_n = E^*] \to 1$ as $n \to \infty$. Denote by $\mathcal{N}^*(r)$ the set of neighbors of a vertex $r \in V$, so that $\mathcal{N}^*(r) = \{t : (r,t) \in E^*\}$. Then the graphical model selection problem is equivalent to that of estimating the neighborhoods $\hat{\mathcal{N}}_n(r) \subset V$, so that $\mathbb{P}[\hat{\mathcal{N}}_n(r) = \mathcal{N}^*(r); \forall r \in V] \to 1$ as $n \to \infty$.

For any pair of random variables X_r and X_t, the parameter θ_{rt} fully characterizes whether there is an edge between them, and can be estimated via its conditional likelihood. In particular, defining $\Theta_r := (\theta_{r1}, \ldots, \theta_{rp})$, our goal is to use the conditional likelihood of X_r conditioned on $X_{V \setminus r}$ to estimate Θ_r and hence its neighborhood $\mathcal{N}(r)$. This conditional distribution of X_r conditioned on $X_{V \setminus r}$ generated by (2) is given by the logistic model

$$\mathbb{P}\Big(X_r = x_r \Big| X_{V \setminus r} = x_{V \setminus r}\Big) = \frac{\exp(\theta_r x_r + \sum_{t \in V \setminus r} \theta_{rt} x_r x_t)}{1 + \exp(\theta_r + \sum_{r \in V \setminus r} \theta_{rt} x_r)}.$$

Given the n samples D, the corresponding conditional log-likelihood is given by

$$\mathcal{L}(\Theta_r; D) = \frac{1}{n} \sum_{i=1}^{n} \Bigg\{ \log\Bigg(1 + \exp\Big(\theta_r x^{(i)} + \sum_{t \in V \setminus r} \theta_{rt} x_r^{(i)} x_t^{(i)}\Big)\Bigg) - \theta_r x_r^{(i)} - \sum_{t \in V \setminus r} \theta_{rt} x_r^{(i)} x_t^{(i)} \Bigg\}. \tag{4}$$

In Section 4, we study a greedy algorithm (Algorithm 2) that finds these node neighborhoods $\hat{\mathcal{N}}_n(r) = \mathrm{Supp}(\hat{\Theta}_r)$ of each random variable X_r separately by a greedy stagewise optimization of the conditional log-likelihood of X_r conditioned on $X_{V \setminus r}$. The algorithm then combines these neighborhoods to obtain a graph estimate $\hat{E}$ using an "OR" rule: $\hat{E}_n = \cup_r \{(r,t) : t \in \hat{\mathcal{N}}_n(r)\}$. Other rules such as the "AND" rule, that add an edge only if it occurs in each of the respective node neighborhoods, could be used to combine the node-neighborhoods to a graph estimate. We show in Theorem 2 that the neighborhood selection by the greedy algorithm succeeds in recovering the exact node-neighborhoods with high probability, so that by a union bound, the graph estimates using either the AND or OR rules would be exact with high probability as well.

Before we describe this greedy algorithm and its analysis in Section 4 however, we first consider the general statistical model case in the next section. We first describe the forward backward greedy algorithm of Zhang [28] as applied to general statistical models, followed by a sparsistency analysis for this general case. We then specialize these general results in Section 4 to the graphical model case. The next section is thus of independent interest even outside the context of graphical models.

3 Greedy Algorithm for General Losses

Consider a random variable Z with distribution $\mathbb{P}$, and let $Z_1^n := \{Z_1, \ldots, Z_n\}$ denote n observations drawn i.i.d. according to $\mathbb{P}$. Suppose we are interested in estimating some parameter $\theta^* \in \mathbb{R}^p$ of the distribution $\mathbb{P}$ that is sparse; denote its number of non-zeroes by $s^* := \|\theta^*\|_0$. Let $\mathcal{L} : \mathbb{R}^p \times \mathcal{Z}^n \mapsto \mathbb{R}$ be some loss function that assigns a cost to any parameter $\theta \in \mathbb{R}^p$, for a given set of observations Z_1^n. For ease of notation, in the sequel, we adopt the shorthand $\mathcal{L}(\theta)$ for $\mathcal{L}(\theta; Z_1^n)$. We assume that θ^* satisfies $\mathbb{E}_Z[\nabla \mathcal{L}(\theta^*)] = 0$.

3

Algorithm 1 Greedy forward-backward algorithm for finding a sparse optimizer of $\mathcal{L}(\cdot)$

Input: Data $D := \{x^{(1)}, \ldots, x^{(n)}\}$, Stopping Threshold $\epsilon_{\mathcal{S}}$, Backward Step Factor $\nu \in (0, 1)$
Output: Sparse optimizer $\widehat{\theta}$

$\widehat{\theta}^{(0)} \longleftarrow \mathbf{0}$ and $\widehat{S}^{(0)} \longleftarrow \phi$ and $k \longleftarrow 1$

while true **do** {*Forward Step*}
 $(j_*, \alpha_*) \longleftarrow \arg \min\limits_{j \in (\widehat{S}^{(k-1)})^c \, ; \, \alpha} \mathcal{L}(\widehat{\theta}^{(k-1)} + \alpha e_j; D)$
 $\widehat{S}^{(k)} \longleftarrow \widehat{S}^{(k-1)} \cup \{j_*\}$
 $\delta_f^{(k)} \longleftarrow \mathcal{L}(\widehat{\theta}^{(k-1)}; D) - \mathcal{L}(\widehat{\theta}^{(k-1)} + \alpha_* e_{j_*}; D)$
 if $\delta_f^{(k)} \leq \epsilon_{\mathcal{S}}$ **then**
  break
 end if

 $\widehat{\theta}^{(k)} \longleftarrow \arg \min\limits_{\theta} \mathcal{L}\big(\theta_{\widehat{S}^{(k)}}; D\big)$
 $k \longleftarrow k + 1$

 while true **do** {*Backward Step*}
  $j^* \longleftarrow \arg \min\limits_{j \in \widehat{S}^{(k-1)}} \mathcal{L}(\widehat{\theta}^{(k-1)} - \widehat{\theta}_j^{(k-1)} e_j; D)$
  if $\mathcal{L}\big(\widehat{\theta}^{(k-1)} - \widehat{\theta}_{j^*}^{(k-1)} e_{j^*}; D\big) - \mathcal{L}\big(\widehat{\theta}^{(k-1)}; D\big) > \nu \delta_f^{(k)}$ **then**
   break
  end if

  $\widehat{S}^{(k-1)} \longleftarrow \widehat{S}^{(k)} - \{j^*\}$
  $\widehat{\theta}^{(k-1)} \longleftarrow \arg \min\limits_{\theta} \mathcal{L}\big(\theta_{\widehat{S}^{(k-1)}}; D\big)$
  $k \longleftarrow k - 1$
 end while

end while

We now consider the forward backward greedy algorithm in Algorithm 1 that rewrites the algorithm in [27] to allow for general loss functions. The algorithm starts with an empty set of active variables $\widehat{S}^{(0)}$ and gradually adds (and removes) vairables to the active set until it meets the stopping criterion. This algorithm has two major steps: the forward step and the backward step. In the forward step, the algorithm finds the *best* next candidate and adds it to the active set as long as it improves the loss function at least by $\epsilon_{\mathcal{S}}$, otherwise the stopping criterion is met and the algorithm terminates. Then, in the backward step, the algorithm checks the *influence* of all variables in the presence of the newly added variable. If one or more of the previously added variables do not contribute at least $\nu \epsilon_{\mathcal{S}}$ to the loss function, then the algorithm removes them from the active set. This procedure ensures that at each round, the loss function is improved by at least $(1 - \nu)\epsilon_{\mathcal{S}}$ and hence it terminates within a finite number of steps.

We state the assumptions on the loss function such that sparsistency is guaranteed. Let us first recall the definition of restricted strong convexity from Negahban et al. [18]. Specifically, for a given set $\mathbb{S}$, the loss function is said to satisfy restricted strong convexity (RSC) with parameter κ_l with respect to the set $\mathbb{S}$ if

$$\mathcal{L}(\theta + \Delta; Z_1^n) - \mathcal{L}(\theta; Z_1^n) - \langle \nabla \mathcal{L}(\theta; Z_1^n), \Delta \rangle \; \geq \; \frac{\kappa_l}{2} \|\Delta\|_2^2 \qquad \text{for all } \Delta \in \mathbb{S}. \qquad (5)$$

We can now define sparsity restricted strong convexity as follows. Specifically, we say that the loss function $\mathcal{L}$ satisfies $RSC(k)$ with parameter κ_l if it satisfies RSC with parameter κ_l for the set $\{\Delta \in \mathbb{R}^p : \|\Delta\|_0 \leq k\}$.

In contrast, we say that the loss function satisfies restricted strong smoothness (RSS) with parameter κ_u with respect to a set $\mathbb{S}$ if

$$\mathcal{L}(\theta + \Delta; Z_1^n) - \mathcal{L}(\theta; Z_1^n) - \langle \nabla \mathcal{L}(\theta; Z_1^n), \Delta \rangle \; \leq \; \frac{\kappa_u}{2} \|\Delta\|_2^2 \qquad \text{for all } \Delta \in \mathbb{S}.$$

We can define $RSS(k)$ similarly. The loss function $\mathcal{L}$ satisfies $RSS(k)$ with parameter κ_u if it satisfies RSS with parameter κ_u for the set $\{\Delta \in \mathbb{R}^p : \|\Delta\|_0 \leq k\}$. Given any constants κ_l and κ_u, and a sample based loss function $\mathcal{L}$, we can typically use concentration based arguments to obtain bounds on the sample size required so that the RSS and RSC conditions hold with high probability.

Another property of the loss function that we require is an upper bound λ_n on the ℓ_∞ norm of the gradient of the loss at the true parameter θ^*, i.e., $\lambda_n \geq \|\nabla\mathcal{L}(\theta^*)\|_\infty$. This captures the "noise level" of the samples with respect to the loss. Here too, we can typically use concentration arguments to show for instance that $\lambda_n \leq c_n(\log(p)/n)^{1/2}$, for some constant $c_n > 0$ with high probability.

Theorem 1 (Sparsistency). *Suppose the loss function $\mathcal{L}(\cdot)$ satisfies $RSC\,(\eta\,s^*)$ and $RSS\,(\eta\,s^*)$ with parameters κ_l and κ_u for some $\eta \geq 2 + 4\rho^2(\sqrt{(\rho^2 - \rho)/s^*} + \sqrt{2})^2$ with $\rho = \kappa_u/\kappa_l$. Moreover, suppose that the true parameters θ^* satisfy $\min_{j \in S^*} |\theta_j^*| > \sqrt{32\rho\epsilon_{\mathcal{S}}/\kappa_l}$. Then if we run Algorithm 1 with stopping threshold $\epsilon_{\mathcal{S}} \geq (8\rho\eta/\kappa_l)\,s^*\lambda_n^2$, the output $\widehat{\theta}$ with support $\widehat{S}$ satisfies:*

(a) **Error Bound:** $\|\widehat{\theta} - \theta^*\|_2 \leq \frac{2}{\kappa_l}\sqrt{s^*}\,(\lambda_n\sqrt{\eta} + \sqrt{\epsilon_{\mathcal{S}}}\sqrt{2\kappa_u})$.

(b) **No False Exclusions:** $S^* - \widehat{S} = \emptyset$.

(c) **No False Inclusions:** $\widehat{S} - S^* = \emptyset$.

Proof. The proof theorem hinges on three main lemmas: Lemmas 1 and 2 which are simple consequences of the forward and backward steps failing when the greedy algorithm stops, and Lemma 3 which uses these two lemmas and extends techniques from [21] and [19] to obtain an ℓ_2 error bound on the error. Provided these lemmas hold, we then show below that the greedy algorithm is sparsistent. However, these lemmas require *apriori* that the RSC and RSS conditions hold for sparsity size $|S^* \cup \widehat{S}|$. Thus, we use the result in Lemma 4 that if $RSC(\eta s^*)$ holds, then the solution when the algorithm terminates satisfies $|\widehat{S}| \leq (\eta - 1)s^*$, and hence $|\widehat{S} \cup S^*| \leq \eta s^*$. Thus, we can then apply Lemmas 1, 2 and Lemma 3 to complete the proof as detailed below.

(a) The result follows directly from Lemma 3, and noting that $|\widehat{S} \cup S^*| \leq \eta s^*$. In this Lemma, we show that the upper bound holds by drawing from fixed point techniques in [21] and [19], and by using a simple consequence of the forward step failing when the greedy algorithm stops.

(b) We follow the chaining argument in [27]. For any $\tau \in \mathbb{R}$, we have
$$\tau|\{j \in S^* - \widehat{S} : |\theta_j^*|^2 > \tau\}| \leq \|\theta_{S^*-\widehat{S}}^*\|_2^2 \leq \|\theta^* - \widehat{\theta}\|_2^2$$
$$\leq \frac{8\eta s^*\lambda_n^2}{\kappa_l^2} + \frac{16\kappa_u\epsilon_{\mathcal{S}}}{\kappa_l^2}|S^* - \widehat{S}|,$$
where the last inequality follows from part (a) and the inequality $(a + b)^2 \leq 2a^2 + 2b^2$. Now, setting $\tau = \frac{32\kappa_u\epsilon_{\mathcal{S}}}{\kappa_l^2}$, and dividing both sides by $\tau/2$ we get
$$2|\{j \in S^* - \widehat{S} : |\theta_j^*|^2 > \tau\}| \leq \frac{\eta s^*\lambda_n^2}{2\kappa_u\epsilon_{\mathcal{S}}} + |S^* - \widehat{S}|.$$
Substituting $|\{j \in S^* - \widehat{S} : |\theta_j^*|^2 > \tau\}| = |S^* - \widehat{S}| - |\{j \in S^* - \widehat{S} : |\theta_j^*|^2 \leq \tau\}|$, we get
$$|S^* - \widehat{S}| \leq |\{j \in S^* - \widehat{S} : |\theta_j^*|^2 \leq \tau\}| + \frac{\eta s^*\lambda_n^2}{2\kappa_u\epsilon_{\mathcal{S}}} \leq |\{j \in S^* - \widehat{S} : |\theta_j^*|^2 \leq \tau\}| + 1/2,$$
due to the setting of the stopping threshold $\epsilon_{\mathcal{S}}$. This in turn entails that
$$|S^* - \widehat{S}| \leq |\{j \in S^* - \widehat{S} : |\theta_j^*|^2 \leq \tau\}| = 0,$$
by our assumption on the size of the minimum entry of θ^*.

(c) From Lemma 2, which provides a simple consequence of the backward step failing when the greedy algorithm stops, for $\widehat{\Delta} = \widehat{\theta} - \theta^*$, we have $\epsilon_{\mathcal{S}}/\kappa_u|\widehat{S} - S^*| \leq \|\widehat{\Delta}_{\widehat{S}-S^*}\|_2^2 \leq \|\widehat{\Delta}\|_2^2$, so that using Lemma 3 and that $|S^* - \widehat{S}| = 0$, we obtain that $|\widehat{S} - S^*| \leq \frac{4\eta s^*\lambda_n^2\kappa_u}{\epsilon_{\mathcal{S}}\kappa_l^2} \leq 1/2$, due to the setting of the stopping threshold $\epsilon_{\mathcal{S}}$. $\qquad\square$

Algorithm 2 Greedy forward-backward algorithm for pairwise discrete graphical model learning
---

Input: Data $D := \{x^{(1)}, \ldots, x^{(n)}\}$, Stopping Threshold $\epsilon_{\mathcal{S}}$, Backward Step Factor $\nu \in (0, 1)$
Output: Estimated Edges $\widehat{E}$

for $r \in V$ **do**
    Run Algorithm 1 with the loss $\mathcal{L}(\cdot)$ set as in (4), to obtain $\widehat{\Theta}_r$ with support $\widehat{\mathcal{N}_r}$
end for

Output $\widehat{E} = \bigcup_r \left\{ (r, t) : t \in \widehat{\mathcal{N}_r} \right\}$

3.1 Lemmas for Theorem 1

We list the simple lemmas that characterize the solution obtained when the algorithm terminates, and on which the proof of Theorem 1 hinges.

Lemma 1 (Stopping Forward Step). *When the algorithm 1 stops with parameter $\widehat{\theta}$ supported on $\widehat{S}$, we have*

$$\left| \mathcal{L}\left(\widehat{\theta}\right) - \mathcal{L}\left(\theta^*\right) \right| < \sqrt{2 \left| S^* - \widehat{S} \right| \kappa_u \, \epsilon_{\mathcal{S}}} \, \left\| \widehat{\theta} - \theta^* \right\|_2 .$$

Lemma 2 (Stopping Backward Step). *When the algorithm 1 stops with parameter $\widehat{\theta}$ supported on $\widehat{S}$, we have*

$$\left\| \widehat{\Delta}_{\widehat{S} - S^*} \right\|_2^2 \geq \frac{\epsilon_{\mathcal{S}}}{\kappa_u} \left| \widehat{S} - S^* \right| .$$

Lemma 3 (Stopping Error Bound). *When the algorithm 1 stops with parameter $\widehat{\theta}$ supported on $\widehat{S}$, we have*

$$\left\| \widehat{\theta} - \theta^* \right\|_2 \leq \frac{2}{\kappa_l} \left(\lambda_n \sqrt{\left| S^* \cup \widehat{S} \right|} + \sqrt{2 \left| S^* - \widehat{S} \right| \kappa_u \epsilon_{\mathcal{S}}} \right) .$$

Lemma 4 (Stopping Size). *If $\epsilon_{\mathcal{S}} > \frac{\lambda_n^2}{\kappa_u} \left(\sqrt{\frac{2}{\eta-1}} - \sqrt{\frac{2}{\eta}} \right)^{-2}$ and $RSC\left(\eta s^*\right)$ holds for some $\eta \geq 2 + 4\rho^2 \left(\sqrt{\frac{\rho^2 - \rho}{s^*}} + \sqrt{2} \right)^2$, then the algorithm 1 stops with $k \leq (\eta - 1)s^*$.*

Notice that if $\epsilon_{\mathcal{S}} \geq (8\rho\eta/\kappa_l) \, (\eta^2/(4\rho^2)) \, \lambda_n^2$, then, the assumption of this lemma is satisfied. Hence for large value of $s^* \geq 8\rho^2 > \eta^2/(4\rho^2)$, it suffices to have $\epsilon_{\mathcal{S}} \geq (8\rho\eta/\kappa_l) \, s^* \lambda_n^2$.

4 Greedy Algorithm for Pairwise Graphical Models

Suppose we are given set of n i.i.d. samples $D := \{x^{(1)}, \ldots, x^{(n)}\}$, drawn from a pairwise Ising model as in (2), with parameters θ^*, and graph $G = (V, E^*)$. It will be useful to denote the maximum node-degree in the graph E^* by d. As we will show, our model selection performance depends critically on this parameter d. We propose Algorithm 2 for estimating the underlying graphical model from the n samples D.

Theorem 2 (Pairwise Sparsistency). *Suppose we run Algorithm 2 with stopping threshold $\epsilon_{\mathcal{S}} \geq c_1 \frac{d \log p}{n}$, where d is the maximum node degree in the graphical model, and the true parameters θ^* satisfy $\frac{c_3}{\sqrt{d}} > \min_{j \in S^*} |\theta_j^*| > c_2 \sqrt{\epsilon_{\mathcal{S}}}$, and further that number of samples scales as*

$$n > c_4 \, d^2 \, \log p,$$

for some constants c_1, c_2, c_3, c_4. Then, with probability at least $1 - c' \exp(-c'' n)$, the output $\widehat{\theta}$ supported on $\widehat{S}$ satisfies:

    (a) **No False Exclusions:** $E^* - \widehat{E} = \emptyset$.

(b) **No False Inclusions:** $\widehat{E} - E^* = \emptyset$.

Proof. This theorem is a corollary to our general Theorem 1. We first show that the conditions of Theorem 1 hold under the assumptions in this corollary.

RSC, RSS. We first note that the conditional log-likelihood loss function in (4) corresponds to a logistic likelihood. Moreover, the covariates are all binary, and bounded, and hence also sub-Gaussian. [19, 2] analyze the RSC and RSS properties of generalized linear models, of which logistic models are an instance, and show that the following result holds if the covariates are sub-Gaussian. Let $\partial\mathcal{L}(\Delta;\theta^*) = \mathcal{L}(\theta^* + \Delta) - \mathcal{L}(\theta^*) - \langle\nabla\mathcal{L}(\theta^*),\Delta\rangle$ be the second order Taylor series remainder. Then, Proposition 2 in [19] states that that there exist constants κ_1^l and κ_2^l, independent of n, p such that with probability at least $1 - c_1\exp(-c_2 n)$, for some constants $c_1, c_2 > 0$,

$$\partial\mathcal{L}(\Delta;\theta^*) \;\geq\; \kappa_1^l\|\Delta\|_2\left\{\|\Delta\|_2 - \kappa_2^l\sqrt{\frac{\log(p)}{n}}\|\Delta\|_1\right\} \qquad \text{for all } \Delta : \|\Delta\|_2 \leq 1.$$

Thus, if $\|\Delta\|_0 \leq k := \eta d$, then $\|\Delta\|_1 \leq \sqrt{k}\|\Delta\|_2$, so that

$$\partial\mathcal{L}(\Delta;\theta^*) \;\geq\; \|\Delta\|_2^2\left(\kappa_1^l - \kappa_2^l\sqrt{\frac{k\log p}{n}}\right) \;\geq\; \frac{\kappa_1^l}{2}\|\Delta\|_2^2,$$

if $n > 4(\kappa_2^l/\kappa_1^l)^2\,\eta d\,\log(p)$. In other words, with probability at least $1 - c_1\exp(-c_2 n)$, the loss function $\mathcal{L}$ satisfies $RSC(k)$ with parameter κ_1^l provided $n > 4(\kappa_2^l/\kappa_1^l)^2\,\eta d\,\log(p)$. Similarly, it follows from [19, 2] that there exist constants κ_1^u and κ_2^u such that with probability at least $1 - c_1'\exp(-c_2' n)$,

$$\partial\mathcal{L}(\Delta;\theta^*) \;\leq\; \kappa_1^u\|\Delta\|_2\{\|\Delta\|_2 - \kappa_2^u\|\Delta\|_1\} \qquad \text{for all } \Delta : \|\Delta\|_2 \leq 1,$$

so that by a similar argument, with probability at least $1 - c_1'\exp(-c_2' n)$, the loss function $\mathcal{L}$ satisfies $RSS(k)$ with parameter κ_1^u provided $n > 4(\kappa_2^u/\kappa_1^u)^2\,\eta d\,\log(p)$.

Noise Level. Next, we obtain a bound on the noiselevel $\lambda_n \geq \|\nabla\mathcal{L}(\theta^*)\|_\infty$ following similar arguments to [20]. Let W denote the gradient $\nabla\mathcal{L}(\theta^*)$ of the loss function (4). Any entry of W has the form $W_t = \frac{1}{n}\sum_{i=1}^n Z_{rt}^{(i)}$, where $Z_{rt}^{(i)} = x_t^{(i)}(x_r^{(i)} - \mathbb{P}(x_r = 1|x_{\backslash s}^{(i)}))$ are zero-mean, i.i.d. and bounded $|Z_{rt}^{(i)}| \leq 1$. Thus, an application of Hoeffding's inequality yields that $\mathbb{P}[|W_t| > \delta] \leq 2\exp(-2n\delta^2)$. Applying a union bound over indices in W, we get $\mathbb{P}[\|W\|_\infty > \delta] \leq 2\exp(-2n\delta^2 + \log(p))$. Thus, if $\lambda_n = (\log(p)/n)^{1/2}$, then $\|W\|_\infty \leq \lambda_n$ with probability at least $1 - \exp(-n\lambda_n^2 + \log(p))$.

We can now verify that under the assumptions in the corollary, the conditions on the stopping size $\epsilon_{\mathcal{S}}$ and the minimum absolute value of the non-zero parameters $\min_{j\in S^*}|\theta_j^*|$ are satisfied. Moreover, from the discussion above, under the sample size scaling in the corollary, the required RSC and RSS conditions hold as well. Thus, Theorem 1 yields that each node neighborhood is recovered with no false exclusions or inclusions with probability at least $1 - c'\exp(-c'' n)$. An application of a union bound over all nodes completes the proof.

$\square$

Remarks. The sufficient condition on the parameters imposed by the greedy algorithm is a restricted strong convexity condition [19], which is weaker than the irrepresentable condition required by [20]. Further, the number of samples required for sparsistent graph recovery scales as $O(d^2\log p)$, where d is the maximum node degree, in contrast to $O(d^3\log p)$ for the ℓ_1 regularized counterpart. We corroborate this in our simulations, where we find that the greedy algorithm requires fewer observations than [20] for sparsistent graph recovery.

We also note that the result can also be extended to the general pairwise graphical model case, where each random variable takes values in the range $\{1,\ldots,m\}$. In that case, the conditional likelihood of each node conditioned on the rest of the nodes takes the form of a multiclass logistic model, and the greedy algorithm would take the form of a "group" forward-backward greedy algorithm, which would add or remove all the parameters corresponding to an edge as a group. Our analysis however naturally extends to such a group greedy setting as well. The analysis for RSC and RSS remains the same and for bounds on λ_n, see equation (12) in [15]. We defer further discussion on this due to the lack of space.

(a) Chain (Line Graph)

(b) 4-Nearest Neighbor (Grid Graph)

(c) Star
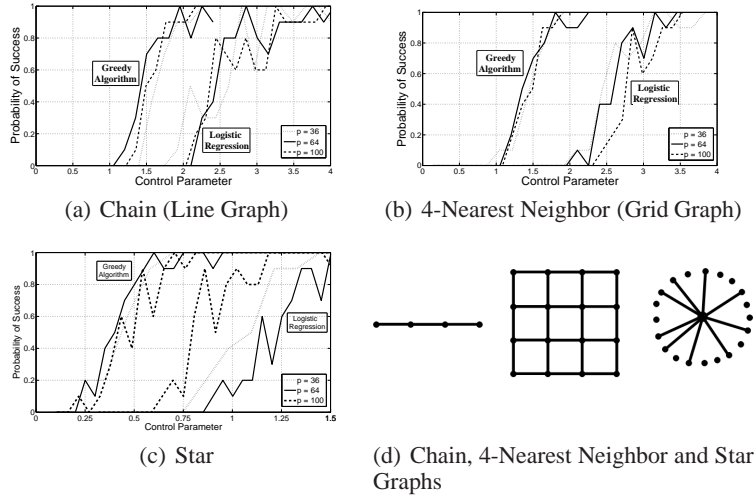
(d) Chain, 4-Nearest Neighbor and Star Graphs

Fig 1: Plots of success probability $\mathbb{P}[\widehat{\mathcal{N}}_{\pm}(r) = \mathcal{N}^*(r), \forall r \in V]$ versus the control parameter $\beta(n, p, d) = n/[20d\log(p)]$ for Ising model on (a) chain $(d = 2)$, (b) 4-nearest neighbor $(d = 4)$ and (c) Star graph $(d = 0.1p)$. The coupling parameters are chosen randomly from $\theta_{st}^* = \pm 0.50$ for both greedy and node-wise ℓ_1-regularized logistic regression methods. As our theorem suggests and these figures show, the greedy algorithm requires less samples to recover the exact structure of the graphical model.

5 Experimental Results

We now present experimental results that illustrate the power of Algorithm 2 and support our theoretical guarantees. We simulated structure learning of different graph structures and compared the learning rates of our method to that of node-wise ℓ_1-regularized logistic regression as outlined in [20].

We performed experiments using 3 different graph structures: (a) chain (line graph), (b) 4-nearest neighbor (grid graph) and (c) star graph. For each experiment, we assumed a pairwise binary Ising model in which each $\theta_{rt}^* = \pm 1$ randomly. For each graph type, we generated a set of n i.i.d. samples $\{x^{(1)}, ..., x^{(n)}\}$ using Gibbs sampling. We then attempted to learn the structure of the model using both Algorithm 2 as well as node-wise ℓ_1-regularized logistic regression. We then compared the actual graph structure with the empirically learned graph structures. If the graph structures matched completely then we declared the result a *success* otherwise we declared the result a *failure*. We compared these results over a range of sample sizes (n) and averaged the results for each sample size over a batch of size 10. For all greedy experiments we set the stopping threshold $\epsilon_{\mathcal{S}} = \frac{c\log(np)}{n}$, where c is a tuning constant, as suggested by Theorem 2, and set the backwards step threshold $\nu = 0.5$. For all logistic regression experiments we set the regularization parameter $\lambda_n = c'\sqrt{\log(p)/n}$, where c' was set via cross-validation.

Figure 1 shows the results for the chain $(d = 2)$, grid $(d = 4)$ and star $(d = 0.1p)$ graphs using both Algorithm 2 and node-wise ℓ_1-regularized logistic regression for three different graph sizes $p \in \{36, 64, 100\}$ with mixed (random sign) couplings. For each sample size, we generated a batch of 10 different graphical models and averaged the probability of success (complete structure learned) over the batch. Each curve then represents the probability of success versus the control parameter $\beta(n, p, d) = n/[20d\log(p)]$ which increases with the sample size n. These results support our theoretical claims and demonstrate the efficiency of the greedy method in comparison to node-wise ℓ_1-regularized logistic regression [20].

6 Acknowledgements

References

[1] P. Abbeel, D. Koller, and A. Y. Ng. Learning factor graphs in polynomial time and sample complexity. *Jour. Mach. Learning Res.*, 7:1743–1788, 2006.

[2] A. Agarwal, S. Negahban, and M. Wainwright. Convergence rates of gradient methods for high-dimensional statistical recovery. In *NIPS*, 2010.

[3] F. Bach. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384–414, 2010.

[4] G. Bresler, E. Mossel, and A. Sly. Reconstruction of markov random fields from samples: Some easy observations and algorithms. In *RANDOM 2008*.

[5] E. Candes and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 2006.

[6] S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Computing*, 20(1):33–61, 1998.

[7] D. Chickering. Learning Bayesian networks is NP-complete. *Proceedings of AI and Statistics*, 1995.

[8] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Info. Theory*, 14(3):462–467, 1968.

[9] I. Csiszár and Z. Talata. Consistent estimation of the basic neighborhod structure of Markov random fields. *The Annals of Statistics*, 34(1):123–145, 2006.

[10] C. Dahinden, M. Kalisch, and P. Buhlmann. Decomposition and model selection for large contingency tables. *Biometrical Journal*, 52(2):233–252, 2010.

[11] S. Dasgupta. Learning polytrees. In *Uncertainty on Artificial Intelligence*, pages 134–14, 1999.

[12] D. Donoho and M. Elad. Maximal sparsity representation via ℓ_1 minimization. *Proc. Natl. Acad. Sci.*, 100:2197–2202, March 2003.

[13] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 28:337–374, 2000.

[14] E. Ising. Beitrag zur theorie der ferromagnetismus. *Zeitschrift für Physik*, 31:253–258, 1925.

[15] A. Jalali, P. Ravikumar, V. Vasuki, and S. Sanghavi. On learning discrete graphical models using group-sparse regularization. In *Inter. Conf. on AI and Statistics (AISTATS) 14*, 2011.

[16] S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of markov networks using l1-regularization. In *Neural Information Processing Systems (NIPS) 19*, 2007.

[17] N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3), 2006.

[18] S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Neural Information Processing Systems (NIPS) 22*, 2009.

[19] S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Arxiv*, 2010.

[20] P. Ravikumar, M. J. Wainwright, and J. Lafferty. High-dimensional ising model selection using ℓ_1-regularized logistic regression. *Annals of Statistics*, 38(3):1287–1319.

[21] A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. 2: 494–515, 2008.

[22] P. Spirtes, C. Glymour, and R. Scheines. Causation, prediction and search. *MIT Press*, 2000.

[23] N. Srebro. Maximum likelihood bounded tree-width Markov networks. *Artificial Intelligence*, 143(1): 123–138, 2003.

[24] V. N. Temlyakov. Greedy approximation. *Acta Numerica*, 17:235–409, 2008.

[25] S. van de Geer. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, 36: 614–645, 2008.

[26] D. J. A. Welsh. *Complexity: Knots, Colourings, and Counting*. LMS Lecture Note Series. Cambridge University Press, Cambridge, 1993.

[27] T. Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Neural Information Processing Systems (NIPS) 21*, 2008.

[28] T. Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568, 2009.