Greedy Learning of Graphical Models with Small Girth

Avik Ray, Sujay Sanghavi and Sanjay Shakkottai

Abstract— **This paper develops two new greedy algorithms for learning the Markov graph of discrete probability distributions, from samples thereof. For finding the neighborhood of a node (i.e. variable), the simple, naive greedy algorithm iteratively adds the new node that gives the biggest improvement in prediction performance over the existing set. While fast to implement, this can yield incorrect graphs when there are many short cycles, as now the single node that gives the best prediction can be outside the neighborhood.**

Our new algorithms get around this in two different ways. The *forward-backward greedy* algorithm includes a deletion step, which goes back and prunes incorrect nodes that may have initially been added. The *recursive greedy* algorithm uses forward steps in a two-level process, running greedy iterations in an inner loop, but only including the final node. We show, both analytically and empirically, that these algorithms can learn graphs with small girth which other algorithms - both greedy, and those based on convex optimization - cannot.

I. Introduction

Graphical models have been widely used to tractably capture dependence relations amongst a collection of random variables in a variety of domains, ranging from statistical physics, social networks to biological applications [1]–[6]. A key challenge in these settings is in learning the precise dependence structure among the random variables – a problem that is known to be NP hard in the number of variables [7]. However, with restrictions placed on the class of graphical models considered, it is known that polynomial time algorithms exist. One of the first results in this spirit is that by Chow and Liu [8], where efficient algorithms for learning tree-structured graphical models were developed. Since then, there have been several algorithms developed for learning restricted classes of graphical models (see Section I-B for more details).

A. Main Contributions

In this paper we propose two new greedy algorithms to find the Markov graph for any discrete graphical model. While greedy algorithms (that learn the structure by sequentially adding nodes and edges to the graph) tend to have low computational complexity, they are known to fail (i.e., do not determine the correct graph structure) in loopy graphs with low girth [12], even when they have access to exact statistics. This is because a non-neighbor can be the best node at a particular iteration; once added, it will always remain. Convex optimization based algorithm like in [9] by Ravikumar et al. (henceforth we call this the RWL algorithm) also cannot provide theoretical guarantees of learning under these situations. These methods require strong incoherence conditions to guarantee success. But such conditions are not always satisfied in graphs with small girth. They are seen to fail empirically too. For example if we run the existing algorithms for an Ising model on a diamond network (Figure 2) with $D = 8$ the performance plot in Figure 1 shows that greedy and RWL algorithms fail to learn the correct graph even with large number of samples.

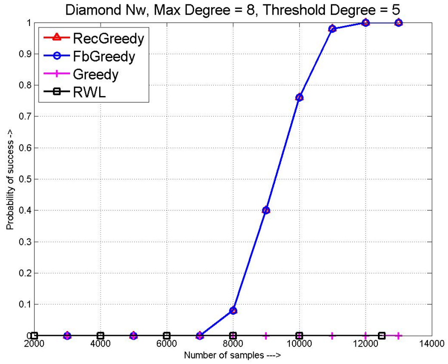

Fig. 1. Performance of different algorithms in an Ising model on diamond network with 10 nodes. Both the $Greedy(\epsilon)$ and RWL algorithms estimate an incorrect edge between nodes 0 and 9 therefore never recovers the true graph G, while our new $RecGreedy(\epsilon)$, $FbGreedy(\epsilon, \alpha)$ algorithms succeed.

In this paper, we present two algorithms that overcome this shortfall of greedy and convex optimization based algorithms. The recursive greedy algorithm is based on the observation that the *last* node added by the simple, naive greedy algorithm is always a neighbor; thus, we can use the naive greedy algorithm as an inner loop that, after every execution, yields just one more neighbor (instead of the entire set). The forward-backward greedy algorithm takes a different tack, interleaving node addition (forward steps) with node removal (backward steps). In particular, in every iteration, the algorithm looks for nodes in the existing set that have a very small marginal effect; these are removed. Note that these nodes may have had a big effect in a previous iteration when they were added, but the inclusion of subsequent nodes shows them to not have enough of a direct effect.

Thus our main contributions are as follows:

- Two greedy algorithms: *(i)* A recursive greedy algorithm, and *(ii)* a forward-backward greedy algorithm, that correctly learn the graphical model structure for

A. Ray, S. Sanghavi and S. Shakkottai are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, USA, Emails: avik@utexas.edu, sanghavi@mail.utexas.edu, shakkott@austin.utexas.edu. We would like to acknowledge NSF grants 0954059 and 1017525, and ARO grant W911NF-11-1-0265.

any non-degenerate, bounded degree graph (including those with small cycles) (Theorem 1);

- Sample complexity and computational (number of iterations) complexity for the recursive and forward-backward greedy algorithms (with high probability) under non-degeneracy and correlation decay assumptions (Theorems 1 and 2);
- Numerical results that indicate tractable computational complexity for loopy graphs (diamond graph, grid).

B. Related Work

Several approaches have been taken so far to learn the graph structure of MRF in presence of cycles. These can be broadly divided into three classes – search based, convex-optimization based, and greedy methods. Search based algorithms like local independence test by Bresler et al. in [10] and the conditional variation distance thresholding (CVDT) by Anandkumar et al. in [13] try to find the smallest set of nodes through exhaustive search, conditioned on which either a given node is independent of rest of the nodes in the graph, or a pair of nodes are independent of each other. These algorithms although have a fairly good sample complexity, but due to exhaustive search they have a high computation complexity.

In case of Ising models a convex optimization based learning algorithm was proposed in [9] by Ravikumar et al. This was further generalized for any pairwise graphical model in [11]. These algorithms have a very good sample complexity of $\Omega(\Delta^3 \log p)$, where Δ is the maximum degree of a node and p is the total number of nodes. However these algorithms require a strong incoherence assumption to guarantee its success.

Recently a greedy learning algorithm was proposed in [12] which tries to find the minimum value of the conditional entropy of a particular node in order to estimate its neighborhood. We call this algorithm as $Greedy(\epsilon)$. It was shown that for graphs with correlation decay and large girth this exactly recovers the graph G. But it fails for graphs with small cycles. A forward-backward greedy algorithm based on convex optimization was also presented recently by Jalali et al. in [18], which works for any pairwise graphical model.

This paper is organized as follows. First we review the definition of a graphical model in section II and then the graphical model learning problem in section III. The two greedy algorithms are described in section IV. Next we give sufficient conditions for the success of the greedy algorithms in section V. In section VI we present the main theorems showing the performance of the recursive greedy and forward-backward greedy algorithms. We compare the performance of our algorithm with other well known algorithms in section VII. Finally in section VIII we present some simulation results.

II. Brief Review: Graphical Models

In this section we briefly review the general graphical model structure and the Ising model. Let $X = (X_1, X_2, \ldots, X_p)$ be a random vector over a discrete set $\mathcal{X}^p$, where $\mathcal{X} = \{1, 2, \ldots, m\}$. $X_S = (X_i : i \in S)$ denote the random vector over the subset $S \subseteq \{1, 2, \ldots, p\}$. Let $G = (V, E)$ denote a graph having p nodes. Let Δ be the maximum degree of the graph G and Δ_i be the degree of the i^{th} node. An undirected graphical model or Markov random field is a tuple $M = (G, X)$ such that each node in G correspond to a particular random variable in X. Moreover G captures the Markov dependence between the variables X_i such that absence of an edge (i, j) implies the conditional independence of variables X_i and X_j given all the other variables.

For any node $r \in V$, let $\mathcal{N}_r$ denote the set of neighbors of r in G. Then the distribution $\mathbb{P}(X)$ has the special Markov property that for any node r, X_r is conditionally independent of $X_{V\setminus\{r\}\bigcup\mathcal{N}_r}$ given $X_{\mathcal{N}_r} = \{X_i : i \in \mathcal{N}_r\}$, the neighborhood of r, i.e.

$$\mathbb{P}(X_r | X_{V\setminus r}) = \mathbb{P}(X_r | X_{\mathcal{N}_r}) \tag{1}$$

Ising Model: An Ising model is a pairwise graphical model where X_i take values in the set $\mathcal{X} = \{-1, 1\}$. For this paper we also consider the node potentials as zero (the zero field Ising model). Hence the distribution take the following simplified form.

$$\mathbb{P}_\Theta(X = x) = \frac{1}{Z} \exp\left\{ \sum_{(i,j)\in E} \theta_{ij} x_i x_j \right\} \tag{2}$$

where $x_i, x_j \in \{-1, 1\}$ and Z is the normalizing constant.

III. Graphical Model Selection

In this section we describe the general graphical model selection problem. The graphical model selection problem is as follows. Given n independent samples $\mathcal{S}_n = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$ from the distribution $\mathbb{P}(X)$, where each $x^{(i)}$ is a p dimensional vector $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_p^{(i)}) \in \{1, \ldots, m\}^p$, the problem is to estimate the Markov graph G corresponding to the distribution $\mathbb{P}(X)$ by recovering the correct edge set E. This problem is very hard in general and has been solved only under special assumptions on the graphical model structure. In some cases a learning algorithm is able to find the correct neighborhood of each node $v \in V$ with a high probability and hence recover the true topology of the graph. We describe some notations. For a subset $S \subseteq V$, we define $P(x_S) = \mathbb{P}(X_S = x_S)$, $x_S \in \mathcal{X}^{|S|}$. The empirical distribution $\widehat{P}(X)$ is the distribution of X computed from the samples. Let $i \in V - S$, the entropy of the random variable X_i conditioned on X_S is written as $H(X_i|X_S)$. The empirical entropy calculated corresponding to the empirical distribution $\widehat{P}$ is denoted by $\widehat{H}$. If P and Q are two probability measures over a finite set $\mathcal{Y}$, then the total variational distance between them is given by, $||P - Q||_{TV} = \frac{1}{2}\sum_{y\in\mathcal{Y}} |P(y) - Q(y)|$.

IV. Greedy Algorithms

In this section we describe two new greedy algorithms for learning the structure of a MRF.

Main idea: The algorithms can be divided into two steps. The first step common to both algorithms is a node pruning step called super-neighborhood selection (described in detail in section V). This step generates a collection of proper super-neighborhoods $\mathcal{S} = \{S_i \subseteq V : \mathcal{N}_i \subseteq S_i \text{ and } i \notin S_i \; \forall i \in V\}$ one for each $i \in V$, such that $|S_i|$ is small. This super-neighborhood set $\mathcal{S}$ is the input to the second step of the algorithms which is described next.

A. Recursive Greedy Algorithm

The simple naive greedy algorithm [12] adds nodes to the neighborhood until there is no further reduction in conditional entropy. This will happen when the true neighborhood is a subset of the estimated neighborhood. Our key observation here is that the *last* node to be added by the naive greedy algorithm will always be in the true neighborhood. We leverage this observation by using the naive greedy algorithm as an inner loop; at the end of every run of this inner loop, we pick only the last node and add it to the estimated neighborhood. The next inner loop starts with this added node as an initial condition, and finds the next one. Hence the algorithm discovers a neighbor in each run of the innermost loop and finds all the neighbors of a given node i in exactly Δ_i iterations of the outer loop.

The above idea works as long as every neighbor has a measurable effect on the conditional entropy, even when there are several other variables in the conditioning. The algorithm is $RecGreedy(\epsilon)$, pseudocode detailed below. It needs a non-degeneracy parameter ϵ, which is the threshold for how much effect each neighbor has on the conditional entropy.

B. Forward-Backward Greedy Algorithm

Our second algorithm takes a different approach to fix the problem of spurious nodes added by the naive greedy algorithm, by adding a backward step that prunes nodes it detects as being spurious. In particular, after every forward step that adds a node to the estimated neighborhood, the algorithm finds the node in this new estimated neighborhood that has the smallest individual effect on the new conditional entropy. If this is too small, this node is removed from the estimated neighborhood.

The algorithm, $FbGreedy(\epsilon, \alpha)$, is given in pseudocode. It takes two input parameters beside the samples. The first is the same non-degeneracy parameter ϵ as in the $RecGreedy(\epsilon)$ algorithm. The second parameter $\alpha \in (0, 1)$ is utilized by the algorithm to determine the threshold of elimination in the backward step. The algorithm stops when there are no further forward or backward steps.

V. Sufficient Conditions for Learning

In this section we describe the sufficient conditions which guarantees that the $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon, \alpha)$ algorithms recover the correct Markov graph G.

Algorithm 1 $RecGreedy(\epsilon)$

```
Generate super-neighborhood $\mathcal{S}$
for $i = 1$ to $|V|$ do
    $\widehat{N}(i) \leftarrow \phi$
    iterate ← TRUE
    while iterate do
        $\widehat{T}(i) \leftarrow \widehat{N}(i)$
        last ← 0
        complete ← FALSE
        while ! complete do
            $j = \arg\min_{k \in S_i \setminus \widehat{T}(i)} \widehat{H}(X_i | X_{\widehat{T}(i)}, X_k)$
            if $\widehat{H}(X_i|X_{\widehat{T}(i)}, X_j) < \widehat{H}(X_i|X_{\widehat{T}(i)}) - \frac{\epsilon}{2}$ then
                $\widehat{T}(i) \leftarrow \widehat{T}(i) \bigcup \{j\}$
                last ← $j$
            else
                if last $! = 0$ then
                    $\widehat{N}(i) \leftarrow \widehat{N}(i) \bigcup \{\text{last}\}$
                else
                    iterate ← FALSE
                end if
                complete ← TRUE
            end if
        end while
    end while
end for
```

Algorithm 2 $FbGreedy(\epsilon, \alpha)$

```
Generate super-neighborhood $\mathcal{S}$
for $i = 1$ to $|V|$ do
    $\widehat{N}(i) \leftarrow \phi$
    added ← FALSE
    complete ← FALSE
    while ! complete do                                   ▷ Forward Step:
        $j = \arg\min_{k \in S_i \setminus \widehat{N}(i)} \widehat{H}(X_i | X_{\widehat{N}(i)}, X_k)$
        if $\widehat{H}(X_i|X_{\widehat{N}(i)}, X_j) < \widehat{H}(X_i|X_{\widehat{N}(i)}) - \frac{\epsilon}{2}$ then
            $\widehat{N}(i) \leftarrow \widehat{N}(i) \bigcup \{j\}$
            added ← TRUE
        else
            added ← FALSE
        end if                                            ▷ Backward Step:
        $l = \arg\min_{k \in \widehat{N}(i)} \widehat{H}(X_i | X_{\widehat{N}(i) \setminus k})$
        if $\widehat{H}(X_i|X_{\widehat{N}(i) \setminus l}) - \widehat{H}(X_i|X_{\widehat{N}(i)}) < \frac{\alpha\epsilon}{2}$ then
            $\widehat{N}(i) \leftarrow \widehat{N}(i) \setminus \{l\}$
        else
            if ! added then
                complete ← TRUE
            end if
        end if
    end while
end for
```

A. Non-degeneracy

Our non-degeneracy assumption require every neighbor have a significant effect. Other graphical model learning algorithms require similar assumption to ensure correctness [9], [10], [12].

(A1) Non-degeneracy condition: Consider the graphical model $M = (G, X)$, where $G = (V, E)$. Then for all $i \in V$ and $A \subset V$ such that $\mathcal{N}_i \not\subset A$ the following condition holds. Let $j \in \mathcal{N}_i$ and $j \notin A$. Then there exists $\epsilon > 0$ such that

$$H(X_i|X_A, X_j) \quad < \quad H(X_i|X_A) - \epsilon \tag{3}$$

Thus by adding a neighboring node to the conditioning set the conditional entropy strictly decreases by at least ϵ. Also the above condition together with the local Markov property (1) imply that the conditional entropy attains an unique minimum at $H(X_i|X_{\mathcal{N}_i})$.

B. Correlation Decay

Correlation decay broadly means that the influence of a random variable on the distribution of another gradually decreases as the path distance between the corresponding nodes increase in the graph G. In [17] Bento et al. showed that learning graphical models become more difficult in absence of some sort of correlation decay. Many different forms of correlation decay have been assumed in MRF learning algorithms [10], [12], [13]. We assume a weak form of correlation decay similar to the weak spatial mixing assumption in [19]. First we define the following quantity.

Definition 1 *Consider the graphical model $M = (G, X)$. Let $i, j \in V$. Define $\phi_i(j) = \max_{x \neq x'} ||P(X_i|X_j = x) - P(X_i|X_j = x')||_{TV}$. The corresponding function calculated from the empirical distribution $\widehat{P}$ is denoted as $\widehat{\phi}_i(j)$.*

Now the correlation decay assumption is the following.

(A2) Correlation decay: For the graphical model $M = (G, X)$ there exists a monotonic decreasing function $f : \mathbb{Z} \to \mathbb{R}$ such that for any $i, j \in V$

$$\phi_i(j) < f(d(i, j)) \tag{4}$$

where $d(i, j)$ is the graph distance between nodes i and j. It can be shown that the correlation decay assumption in [12] implies (4). Hence this is a weaker assumption. Next we give an example when the decay function $f(.)$ is exponential.

Example 1 (*Exponential correlation decay*)

It can be shown that in any graphical model $M = (G, X)$ if Dobrushin's condition holds then M exhibits an exponential correlation decay. First we restate the definition of influence coefficient from [14], [15].

Definition 2 *Influence coefficient: For any $i, j \in V$ the influence coefficient of node j on node i is*

$$C_{ij} = \max_{\substack{y,z \in \mathcal{X}^{|V|-1} \\ y_k = z_k \ \forall k \neq j}} ||P(X_i|X_{V\setminus i} = y) - P(X_i|X_{V\setminus i} = z)||_{TV}$$

Note that due to the Markov property of the graph $C_{ij} = 0$ for all $j \notin \mathcal{N}_i$. Dobrushin's condition [16] is the following. **Dobrushin's condition:** Let C_{ij} be the influence coefficient of node j on node i. Then Dobrushin's condition require

$$\gamma = \sup_{i \in V} \left(\sum_{j \in V} C_{ij} \right) < 1 \tag{5}$$

In an Ising model (2) with maximum degree Δ and $\theta_{ij} = \theta$ the Dobrushin's condition corresponds to $\gamma = \Delta \tanh 2\theta < 1$. The following lemma connects this to assumption (A2).

Lemma 1 ([20]) *Suppose Dobrushin's condition holds for a Markov random field. Then,*

$$\phi_i(j) \leq \frac{\gamma^{d(i,j)}}{1 - \gamma}$$

where γ is given by (5).

Hence in this case $f(x) = \frac{\gamma^x}{1-\gamma}$ is an exponentially decaying function.

C. Super-Neighborhood Selection

In this section we describe a method to choose a proper super-neighborhood S_i for each node $i \in V$ in the first step of Algorithm 1, 2 when there is correlation decay. A super-neighborhood S_i is said to be proper if it includes the true neighborhood $\mathcal{N}_i$.

Before we describe the procedure, we motivate the need for a super-neighborhood selection method. First, observe that if we run Algorithm 1, 2 with $S_i = V$ for all $i \in V$ and with exact distribution $P(X)$ known, under the non-degeneracy assumption (A1) the algorithm correctly outputs the true neighborhood $\mathcal{N}_i$ with a proper ϵ. However the problem is that for an arbitrary graphical model (or any graphical model with the super-neighborhood set to be very large), the size of the conditioning set $\widehat{T}(i)$ in $RecGreedy(\epsilon)$ or $\widehat{N}(i)$ in $FbGreedy(\epsilon, \alpha)$ can also become very large. This implies that the number of samples required to get a good estimate of the conditional entropy $H(X_i|X_A)$ is $\Omega(|\mathcal{X}|^{|A|+1})$ will be exponentially large (a good estimate is needed to ensure Algorithm 1, 2 give the correct graph G with a high probability). In order to mitigate this problem we need to appropriately bound the size of the set $\widehat{T}(i)$, $\widehat{N}(i)$. To do this we choose a proper super-neighborhood S_i with $|S_i| = \Theta(poly(\Delta))$. Then the size of the conditioning set never exceeds $\max_{i \in V} |S_i| := \xi$ (a constant when Δ is bounded).

The problem of proper super-neighborhood selection becomes easier under the correlation decay assumption (A2). Let κ be such that,

$$\min_{i \in V, j \in \mathcal{N}_i} \phi_i(j) = \kappa \tag{6}$$

The super-neighborhood is then selected as follows.

$$S_i = \{j \in V | \widehat{\phi}_i(j) \geq \frac{\kappa}{2}\} \quad (7)$$

Remark: Note that there may be other ways to generate a proper super-neighborhood based on domain knowledge/structural properties of the system (e.g. in social networks, weather forecasting). All that is needed for our algorithms is to have a super-neighborhood of small size.

Definition 3 (Super-neighborhood radius) *The super-neighborhood radius R is defined as*

$$R = \min\{x \in \mathbb{Z} | f(x) < \kappa/2\} \quad (8)$$

We assume that R exists and R does not grow with p. i.e., $R = O(1)$.

For example if a MRF satisfies Dobrushin's condition then $R < \log \frac{2}{(1-\gamma)\kappa} / \log \frac{1}{\gamma}$. With the above definition it is clear that for a bounded degree graph G the size of the super-neighborhood set S_i is bounded as $|S_i| \leq \xi < \Delta^R$.

VI. Main Result

In this section we state our main result showing the performance of the $RecGreedy(\epsilon)$, $FbGreedy(\epsilon, \alpha)$ algorithms. First we restate a lemma from [12], [22] that will be used to show the concentration of the empirical entropy $\widehat{H}$ with samples.

Lemma 2 *Let P and Q be two discrete distributions over a finite set $\mathcal{X}$ such that $||P - Q||_{TV} \leq \frac{1}{4}$. Then,*

$$|H(P) - H(Q)| \leq 2||P - Q||_{TV} \log \frac{|\mathcal{X}|}{2||P - Q||_{TV}}$$

We now state our main theorem showing the performance of Algorithms 1, 2.

Theorem 1 *Consider a MRF over a graph G with maximum degree Δ, having a distribution $P(X)$.*
*1) **Correctness (non-random):** Suppose (A1) holds and the $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon, \alpha)$ algorithms have access to the true conditional entropies therein, then they correctly estimate the graph G.*
*2) **Sample complexity:** Suppose (A1) holds, proper super-neighborhoods S_i are given and super-neighborhood size $|S_i| < \xi$, for all $i \in V$. Let $0 < \delta < 1$.*

- *When the number of samples $n = \Omega(|\mathcal{X}|^{2\xi} \xi \log \frac{p}{\delta})$ the $RecGreedy(\epsilon)$ correctly estimates G with probability greater than $1 - \delta$.*
- *When the number of samples $n = \Omega(|\mathcal{X}|^{2\xi} \frac{\xi}{\alpha^4} \log \frac{p}{\delta})$ the $FbGreedy(\epsilon, \alpha)$ correctly estimates G with probability greater than $1 - \delta$, for $0 < \alpha < 1$.*

Proof: The proof of correctness with true conditional entropies known is straightforward under non-degenerate assumption (A1). The proof in presence of samples is based on Lemma 3 similar to Lemma 2 in [12] showing the concentration of empirical conditional entropy, which is critical for the success of Algorithms 1, 2. We show that when proper super-neighborhoods S_i are given and $|S_i| \leq \xi$ for all $i \in V$ with $\Omega(|\mathcal{X}|^{2\xi} \frac{\xi}{\alpha^4} \log \frac{p}{\delta})$ samples the empirical distributions and hence the empirical conditional entropies also concentrate around their true values with a high probability and Algorithm 1, 2 correctly recovers the Markov graph G. ■

Lemma 3 *Consider a graphical model $M = (G, X)$ with distribution $P(X)$. Let $0 < \delta_3 < 1$. If the number of samples*

$$n > \frac{2^{15}|\mathcal{X}|^{2(s+2)}}{\epsilon^4 \alpha^4} \left[(s+1)\log 2p|\mathcal{X}| + \log \frac{1}{\delta_3}\right]$$

then with probability at least $1 - \delta_3$

$$|\widehat{H}(X_i|X_S) - H(X_i|X_S)| < \frac{\alpha\epsilon}{8}$$

for any $S \subset V$ such that $|S| \leq s$.

Lemma 3 follows from Lemma 2 and Azuma's inequality. When there is correlation decay (A2) the super-neighborhood selection procedure (7) is also successful with a high probability with only $\Omega(\log p)$ samples, where by success we mean S_i will be proper and $|S_i| < \Delta^R$. This is shown by the following lemmas. We define the minimum marginal probability P_{min} as $P_{min} = \min_{i \in V, x_i \in \mathcal{X}} P(X_i = x_i)$.

Lemma 4 *Consider a graphical model $M = (G, X)$ with distribution $P(X)$, $X \in \mathcal{X}^p$. Let $0 < \delta_1 < 1$. Then if the number of i.i.d. samples*

$$n > \frac{32|\mathcal{X}|^4}{\kappa^2 P_{min}^2} \left[2\log|\mathcal{X}|p + \log \frac{2}{\delta_1}\right]$$

we have with probability at least $1 - \delta_1$

$$|P(X_i|X_j) - \widehat{P}(X_i|X_j)| < \frac{\kappa}{4|\mathcal{X}|} \quad (9)$$

for all $i, j \in V$, where κ is given by (6).

Lemma 5 *Let a graphical model $M = (G, X)$ satisfy assumption (A2). Let $0 < \delta_1 < 1$. Then with probability greater than $1 - \delta_1$, $\mathcal{N}_i \subseteq S_i$ for all $i \in V$ when the number of i.i.d. samples $n = \Omega(\log \frac{p}{\delta_1})$.*

Lemma 6 *Consider a graphical model $M = (G, X)$ with maximum degree Δ satisfying assumption (A2) with decay function $f(.)$. Let $0 < \delta_2 < 1$. Then with probability greater than $1 - \delta_2$ we have $|S_i| < \Delta^R$ when the number of samples $n = \Omega(\log \frac{p}{\delta_2})$, where R is given by (8).*

Lemmas 4, 5, 6 also follow from Azuma's concentration inequality and the proofs are omitted due to space constraint. Although the sample complexities of $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon, \alpha)$ algorithms are slightly more than other non-greedy algorithms [9], [10], [13], the main appeal of these greedy algorithms lie in their low computation complexity.

The following theorem characterizes the computation complexity of Algorithms 1, 2. In order to do so the first step is to bound the number of greedy steps. We have the following corollaries following from Theorem 1.

Corollary 1 *Let* $T = \min\{\frac{2\log|\mathcal{X}|}{\epsilon}, \xi\}$*. Then the number of greedy steps in each recursion of the* $RecGreedy(\epsilon)$ *is less than* T*.*

Corollary 2 *The number of steps in the* $FbGreedy(\epsilon, \alpha)$ *is bounded by* $\frac{4\log|\mathcal{X}|}{\epsilon(1-\alpha)}$*.*

When calculating the run-time, each arithmetic operation and comparison is counted as an unit-time operation. For example to execute line 10 in Algorithm 1, each comparison takes an unit-time and each entropy calculation takes $O(n)$ time (since there are n samples using which the empirical conditional entropy is calculated). Since there are at most $|S_i| \leq \xi$ comparisons the total time required to execute this line is $O(n\xi)$.

Theorem 2 (Run-time) *Consider a graphical model* $M = (G, X)$*, with maximum degree* Δ*, satisfying assumptions (A1) and* $|S_i| < \xi$*, for all* $i \in V$*. Then the expected run-time of the second step of* $RecGreedy(\epsilon)$ *is* $O(\delta p \xi^3 n + (1-\delta)\frac{p}{\epsilon}\Delta\xi n)$ *and that of the* $FbGreedy(\epsilon, \alpha)$ *algorithm is* $O(\frac{p}{(1-\alpha)\epsilon}\xi n)$*.*

The proofs of Corollary 1, 2 and Theorem 2 are omitted for brevity.

Remark: Suppose that $\frac{4\log|\mathcal{X}|}{\epsilon(1-\alpha)} < \xi$. Then if we take $\alpha < \frac{\Delta - 1}{\Delta}$ the $FbGreedy(\epsilon, \alpha)$ has a better run time guarantee than the $RecGreedy(\epsilon)$ algorithm for small δ. But when $\Delta\xi < \frac{2\log|\mathcal{X}|}{\epsilon(1-\alpha)}$ then the $RecGreedy(\epsilon)$ algorithm has a better guarantee. Note that the super-neighborhood selection step (7) has an additional complexity of $O(p^2)$.

VII. Performance Comparison

In this section we compare the performance of the $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon, \alpha)$ algorithms with other graphical model learning algorithms.

A. Comparison with $Greedy(\epsilon)$ algorithm:

The $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon, \alpha)$ algorithms are strictly better than the $Greedy(\epsilon)$ algorithm in [12]. This is because Algorithms 1 and 2 always finds the correct graph G when the $Greedy(\epsilon)$ finds the correct graph, but they are applicable to a wider class of graphical models since they do not require the assumption of large girth to guarantee its success. Further the correlation decay assumption (A2) in this paper is weaker than the assumption in [12]. Note that the $RecGreedy(\epsilon)$ algorithm uses the $Greedy(\epsilon)$ algorithm in each recursion step and the $FbGreedy(\epsilon, \alpha)$ algorithm uses the $Greedy(\epsilon)$ algorithm in its forward step. Hence when $Greedy(\epsilon)$ finds the true neighborhood $\mathcal{N}_i$ of node i, Algorithm 1 will find the correct neighborhood in each of the recursive steps and Algorithm 2 outputs the correct neighborhood directly without having to utilize any of the backward steps. Hence Algorithms 1 and 2 also succeed in finding the true graph G. We now demonstrate a clear example of a graph where $Greedy(\epsilon)$ fails to recover the true graph but the Algorithms 1, 2 is successful. This example is also presented in [12]. Consider an Ising model on the graph in Figure 2. We have the following proposition.

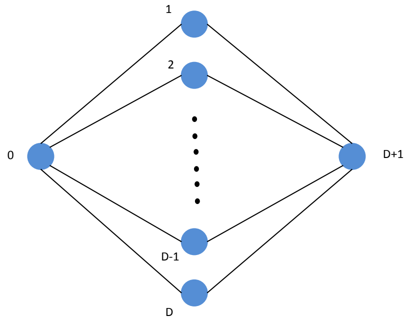

Fig. 2. An example of a diamond network with $D+2$ nodes and maximum degree D where $Greedy(\epsilon)$ fails but $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon, \alpha)$ algorithms correctly recover the true graph.

Proposition 1 *Consider an Ising model with* $V = \{0, 1, \ldots, D, D+1\}$ *and* $E = \{(0, i), (i, D+1)\ \forall i\ : 1 \leq i \leq D\}$ *with a distribution function* $P(x) = \frac{1}{Z}\prod_{(ij)\in E} e^{\theta x_i x_j}$*,* $X_i \in \{1, -1\}$*. Then with* $D > \frac{2\theta}{\log\cosh(2\theta)} + 1$ *we have*

$$H(X_0|X_{D+1}) < H(X_0|X_1)$$

The proof follows from simple calculation. Hence for the Ising model considered above (Figure 2) with $D > \frac{2\theta}{\log\cosh(2\theta)} + 1$ the $Greedy(\epsilon)$ incorrectly includes node $D+1$ in the neighborhood set in the first step. However with an appropriate ϵ the MRF satisfies assumption $(A1)$. Hence by taking $S_i = V$, Theorem 1 ensures that the $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon, \alpha)$ algorithms correctly estimate the graph G.

B. Comparison with search based algorithms:

Search based graphical model learning algorithms like the Local Independence Test (LIT) by Bresler et al. [10] and the Conditional Variation Distance Thresholding (CVDT) by Anandkumar et al. [13] generally have good sample complexity but suffer due to their high computation complexity. As we will see the $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon, \alpha)$ algorithms have slightly more sample complexity but significantly lower computational complexity than the search based algorithms. Moreover to run the search based algorithms one needs to know the maximum degree Δ for LIT and the maximum size of the separator η for the CVDT algorithm. However the greedy algorithms can be run without knowing the maximum degree of the graph.

For bounded degree graphs the LIT algorithm has a sample complexity of $\Omega(|\mathcal{X}|^{4\Delta}\Delta\log\frac{2p}{\delta})$. Without any assumption on the maximum size of the separator, for bounded degree graphs the CVDT algorithm also has a similar sample complexity of $\Omega(|\mathcal{X}|^{2\Delta}(\Delta+2)\log\frac{p}{\delta})$. Note that the quantity P_{min} in the sample complexity expression for CVDT algorithm (Theorem 2 in [13]) is the minimum probability of $P(X_S = x_S)$ where $|S| \leq \eta + 1$. This scales with Δ as $P_{min} \leq \frac{1}{|\mathcal{X}|^{\eta+1}}$. For general degree bounded graphs we have $\eta = \Delta$. The sample complexity for $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon,\alpha)$ algorithms is slightly higher at $\Omega(|\mathcal{X}|^{2\xi}\xi\log\frac{p}{\delta})$ and $\Omega(|\mathcal{X}|^{2\xi}\frac{\xi}{\alpha^4}\log\frac{p}{\delta})$ respectively (since $\xi > \Delta$). However the computation complexity of the LIT algorithm is $O(p^{2\Delta+1}\log p)$ and that of the CVDT algorithm is $O(|\mathcal{X}|^{\Delta}p^{\Delta+2}n)$, which is much larger that $O(\frac{p}{\epsilon}\Delta\xi n)$ for $RecGreedy(\epsilon)$ algorithm and $O(\frac{p}{(1-\alpha)\epsilon}\xi n)$ for the $FbGreedy(\epsilon,\alpha)$ algorithm (since $\xi = \Theta(poly(\Delta))$ and $\xi < \Delta^R$ when (A2) holds). Note however that using the correlation decay property and super-neighborhood selection, the computation complexity of search based algorithms can be decreased. We have the following proposition.

Proposition 2 *Consider a graphical model $M = (G, X)$, where $G = (V, E)$ have maximum degree Δ, satisfying correlation decay (A2). Then by super-neighborhood selection the CVDT algorithm has an expected run-time of $O(p\Delta^{(\Delta+1)R}|\mathcal{X}|^{\Delta}n)$, when the super-neighborhood is chosen as* (7).

However with correlation decay (A2) the run-time of Algorithms 1, 2 are $O(\frac{p}{\epsilon}\Delta^{R+1}n)$ and $O(\frac{p}{(1-\alpha)\epsilon}\Delta^{R}n)$ respectively still smaller than the CVDT algorithm.

C. Comparison with convex optimization based algorithms:

In [9] Ravikumar et al. presented a convex optimization based learning algorithm for Ising models, which we have referred as the RWL algorithm. It was later extended for any pairwise graphical model by Jalali et al. in [11]. Although these algorithms have a low sample complexity of $\Omega(\Delta^3\log p)$, these algorithms have a computation complexity of $O(p^4)$ higher than the $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon,\alpha)$ algorithms. Moreover the greedy algorithms we propose are applicable for a wider class of graphical models. Finally these optimization based algorithms require a strong incoherence property to guarantee its success; conditions which may not hold even for a large class of Ising models as shown in [17]. In our simulation section we will see that even for Ising model on the diamond network (Figure 2) the RWL algorithm fails to recover the correct graph even with large number of samples whenever there is a strong correlation between non-neighbors, our algorithm successfully recovers the correct graph in such scenarios. In [18] Jalali et al. presented a forward-backward algorithm based on convex optimization for learning *pairwise graphical models* (as opposed to general graphical models in this paper). It has even lower sample complexity of $\Omega(\Delta^2\log p)$ and works under slightly milder assumptions than the RWL algorithm.

VIII. Simulation Results

In this section we present some simulation results characterizing the performance of $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon,\alpha)$ algorithms. We compare the performance with the $Greedy(\epsilon)$ algorithm [12] and the logistic regression based RWL algorithm [9] in an Ising model. We consider two graphs, a 4×4 square grid (Figure 3) and the diamond network (Figure 2). In each case we consider an Ising model on the graphs. For the 4×4 grid we take the edge weights $\theta \in \{.25, -.25\}$, generated randomly. For the diamond network we take all equal edge weights $\theta = .25$. Independent and identically distributed samples are generated from the models using Gibbs sampling and the algorithms are run with increasing number of samples. The parameter ϵ for the greedy algorithms and the ℓ_1 regularization parameter λ for the RWL algorithm are chosen through cross validation. For the $FbGreedy(\epsilon,\alpha)$ algorithm α was taken as .9.

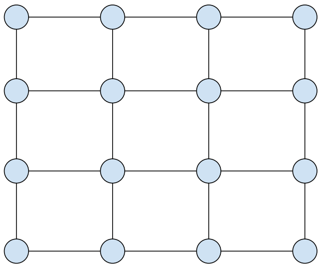

Fig. 3. A 4x4 grid with $\Delta = 4$ and $p = 16$ used for the simulation of the $RecGreedy(\epsilon)$, $FbGreedy(\epsilon,\alpha)$ algorithms.

First we show that for the diamond network (Figure 2) whenever $D > D_{th} = \frac{2\theta}{\log\cosh(2\theta)+1}$ the RWL algorithm fails to recover the correct graph. We run the RWL algorithm in diamond network with increasing maximum degree D keeping θ fixed. We take $\theta = .25$ for which $D_{th} = \frac{2\times.25}{\log\cosh(2\times.25)} + 1 = 5.16$. The performance is shown in Figure 4. We clearly see that the failure of the RWL algorithm in diamond network corresponds exactly to the case when $D > D_{th}$. The RWL algorithm fails since it predicts a false edge between nodes 0 and $D+1$. This is surprising since this is also the condition in Proposition 1 which describes the case when $Greedy(\epsilon)$ algorithm fails for the diamond network due to the same reason of estimating a false edge. In some sense $D = D_{th}$ marks the transition between weak and strong correlation between non-neighbors in the diamond network, and both $Greedy(\epsilon)$ and RWL algorithms fail whenever there is a strong correlation. However see next that our greedy Algorithms 1, 2 succeed even when $D > D_{th}$.

Figure 1 shows the performance of the various algorithms in the case of the diamond network with $D = 8 > D_{th} = 5.16$. The $Greedy(\epsilon)$ and RWL algorithms are unable to recover the graph but the $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon,\alpha)$ recover the true graph G, they also show the same error performance. However Figure 5 shows that $FbGreedy(\epsilon,\alpha)$

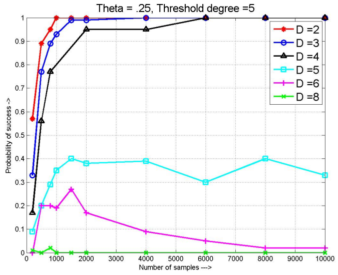

Fig. 4. Performance of the RWL algorithm in diamond network of Figure 2 for varying maximum degree with $\theta = .25$ and $D_{th} = 5$. RWL fails whenever $D > D_{th}$.

has a better runtime than the $RecGreedy(\epsilon)$ algorithm for the diamond network.

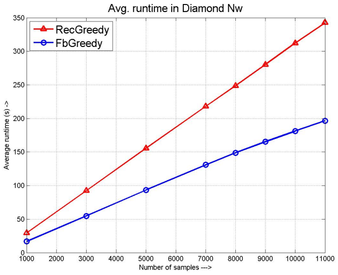

Fig. 5. Figure showing the average runtime performance of $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon, \alpha)$ algorithms for the diamond network with $p = 10$, $\Delta = 8$, for varying sample size.

Figure 6 shows the performance of the different algorithms for a 4×4 grid network. We see that for this network the RWL algorithm shows a better sample complexity than either of $RecGreedy(\epsilon)$ or $FbGreedy(\epsilon, \alpha)$ as predicted by the performance analysis. This network exhibits a weak correlation among non-neighbors, hence the $Greedy(\epsilon)$ is able to correctly recover the graph, which obviously implies that the $RecGreedy(\epsilon)$ and $FbGreedy(\epsilon, \alpha)$ also correctly recovers the graph, and all have the same performance.

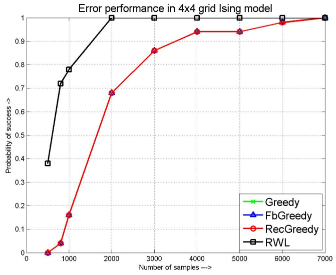

Fig. 6. Performance comparison of $RecGreedy(\epsilon)$, $FbGreedy(\epsilon, \alpha)$, $Greedy(\epsilon)$ and RWL algorithms in a 4×4 grid with $p = 16$, $\Delta = 4$ for varying sample size. The error event is defined as $\mathcal{E} = \{\exists i \in V | \widehat{\mathcal{N}}_i \neq \mathcal{N}_i\}$. All three greedy algorithms have the same error performance for this graph.

References

[1] E. Ising, "Beitrag zur theorie der ferromagnetismus", *Zeitschrift fur Physik* 31, pp. 253–258, 1925.

[2] C. D. Manning and H. Schutze, "Foundations of Statistical Natural Language Processing", MIT Press, Cambridge, MA. MR1722790, 1999.

[3] G. Cross and A. Jain, "Markov random field texture models", *IEEE Trans. PAMI*, 5, pp. 25–39, 1983.

[4] M. Hassner and J. Sklansky, "The use of Markov random fields as models of texture", *Comp. Graphics Image Proc.* 12, pp. 357–370, 1980.

[5] S. Wasserman and P. Pattison "Logit models and logistic regressions for social networks 1. An introduction to Markov graphs and p^*", *Psychometrika* 61, pp. 401-425, 1996.

[6] A. Dobra, C. Hans, B. Jones, J. R. Nevins, G. Yao, and M. West, "Sparse graphical models for exploring gene expression data", *J. Multiv. Anal. 90*, 196–212, 2004.

[7] D. Karger, N. Srebro, "Learning Markov networks: maximum bounded tree-width graphs", *Symposium on Discrete Algorithms*, pp. 392-401, 2001.

[8] C. Chow, C. Liu, "Approximating Discrete Probability Distributions with Dependence Trees", *IEEE Trans. on Information Theory*, vol. 14, pp. 462-467, 1968.

[9] P. Ravikumar, M. Wainwright, J. D. Lafferty, " High-Dimensional Ising Model Selection Using ℓ_1-Regularized Logistic Regression" *The Annals of Statistics*, vol. 38, no. 3, pp. 1287-1319, 2010.

[10] G. Bresler, E. Mossel and A. Sly, " Reconstruction of Markov Random Field from Samples: Some Observations and Algorithms" *Proceedings of the* 11^{th} *international workshop*, APPROX 2008, and *12th international workshop*, RANDOM 2008, pp. 343-356.

[11] A. Jalali, P. Ravikumar, V. Vasuki and S. Sanghavi, "On Learning Discrete Graphical Models using Group-Sparse Regularization", *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[12] P. Netrapalli, S. Banerjee, S. Sanghavi and S. Shakkottai, "Greedy Learning of Markov Network Structure", 48^{th} *Annual Allerton Conference*, 2010.

[13] A. Anandkumar, V. Y. F Tan and A. S. Willsky, "High-Dimensional Structure Learning of Ising Models: Local Separation Criterion", *http://arxiv.org/abs/1107.1736*, July 2011.

[14] A. Anandkumar, J. E. Yukich, and A. Willsky, "Scaling Laws for Random Spatial Graphical Models", *In Proc. of IEEE ISIT*, Austin, USA, June 2010.

[15] S. Winkler, S. Tatikonda, "Criteria for Rapid Mixing of Gibbs Samplers and Uniqueness of Gibbs Measures", *Allerton Conf. on Communication, Control, and Computing*, 2006.

[16] R. L. Dobrushin, "The Problem of Uniqueness of a Gibbsian Random Field and the Problem of Phase Transitions", *Functional Analysis and its Applications*, vol. 2, pp. 302312, 1968.

[17] J. Bento and A. Montanari, "Which graphical models are difficult to learn?", *http://arxiv.org/abs/0910.5761.*, 2009.

[18] A. Jalali, C. Johnson, P. Ravikumar, "On Learning Discrete Graphical Models using Greedy Methods", *In Advances in Neural Information Processing Systems (NIPS) 24*, 2011.

[19] D. Weitz, "Counting independent sets up to the tree threshold", *in Proc. of ACM symp. on Theory of Computing*, pp. 140 – 149, 2006.

[20] A. Montanari, "Lecture Notes: Inference in Graphical Models", *(available online)*, 2011.

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

[22] T. Cover and J. Thomas, "Elements of Information Theory". *John Wiley and Sons, Inc.*, 2006.