Learning Graphical Models Using Multiplicative Weights

Adam R. Klivans* Raghu Meka†

April 6, 2017

Abstract

We give a simple, multiplicative weight update algorithm for learning undirected graphical models or Markov random fields (MRFs). The approach is new, and for the well-studied case of Ising models or Boltzmann machines, we obtain an algorithm that uses a nearly optimal number of samples and has running time $\tilde{O}(n^2)$, subsuming and improving on all prior work.

Our main application is an algorithm for learning t-wise MRFs with sample complexity $n^{O(t)}$ (where we suppress some necessary terms that depend on the weights) and running time $n^{O(t)}$. Our algorithm both reconstructs the underlying graph and generates a hypothesis that is close in statistical distance to the underlying probability distribution. All prior work runs in time $n^{\Omega(d)}$ for graphs of bounded degree d and does not generate a hypothesis close in statistical distance even for $t = 3$. We observe that our runtime has the correct dependence on n and t assuming the hardness of learning sparse parities with noise.

Our algorithm– the *Sparsitron*– is easy to implement (has only one parameter) and holds in the on-line setting. Its analysis applies a regret bound from Freund and Schapire's classic Hedge algorithm. It also gives the first solution to the problem of learning sparse Generalized Linear Models (GLMs).

*Department of Computer Science, University of Texas at Austin, klivans@cs.utexas.edu

†Department of Computer Science, UCLA, raghum@cs.ucla.edu

1 Introduction

Undirected graphical models or *Markov random fields* (MRFs) are one of the most well-studied and influential probabilsitic models with applications to a wide range of scientific disciplines [KF09, Lau98, MRS13, HS86, KFL01, Sal09, Cli90, JEMF06]. Here we focus on binary undirected graphical models which are distributions $(Z_1, \ldots, Z_n)$ on $\{1, -1\}^n$ with an associated undirected graph G - known as the *dependency graph* - on n vertices where each Z_i conditioned on the values of $(Z_j : j$ adjacent to i in $G)$ is independent of the remaining variables.

Developing efficient algorithms for inferring the structure of the underlying graph G from random samples from $\mathcal{D}$ is a central problem in machine learning, statistics, physics, and computer science [AKN06, KS01, WRL06, BMS13, NBSS12, TR14] and has attracted considerable attention from researchers in these fields. A famous early example of such an algorithmic result is due to Chow and Liu from 1968 [CL68] who gave an efficient algorithm for learning graphical models where the underlying graph is a tree. Subsequent work considered generalizations of trees [ATHW11] and graphs under various strong assumptions (e.g., correlation decay [BMS13, RSS12]).

The current frontier of MRF learning has focused on the *Ising model* (also known as *Boltzmann machines*) on *bounded-degree graphs*, a special class of graphical models with only *pairwise interactions* and each vertex having degree at most d in the underlying dependency graph. We refer to [Bre15] for an extensive historical overview of the problem. Two important works of note are due to Bresler [Bre15] and [VMLC16] who learn Ising models on bounded degree graphs.

Bresler's algorithm is a combinatorial (greedy) approach that runs in time $\tilde{O}(n^2)$ but requires doubly exponential in d many samples from the distribution (only singly exponential is necessary). [VMLC16] use machinery from convex programming to achieve nearly optimal sample complexity but with running time $\tilde{O}(n^4)$.

1.1 Our Results

The main contribution of this paper is a simple, multiplicative-weight update algorithm for learning MRFs. Using our algorithm we obtain the following new results:

- An efficient online algorithm for learning Ising models on arbitrary graphs with nearly optimal sample complexity and running time $O(n^2)$ per example (precise statements can be found in Section 5). In particular, for bounded degree graphs we achieve a run-time of $O(n^2)$ with nearly optimal sample complexity. This subsumes and improves all prior work including the above results of Bresler [Bre15] and [VMLC16]. Our algorithm works even for unbounded-degree graphs as long as the ℓ_1 norm of the weight vector of each neighborhood is bounded, a condition necessary for efficiency (see discussion following Corollary 5.4).

- An algorithm for learning general t-wise Markov random fields with sample complexity roughly $M \approx n^{O(t)}$ and running time $O(M \cdot n^t)$ (we suppress some necessary terms that depend on the weights; precise statements can be found in Section 7); precise statements can be found in Section 7). We reconstruct both the underlying graph and output a function f that generates a distribution arbitrarily close in statistical distance.

As far as we are aware, these are the *first efficient algorithms* for learning higher-order MRFs. All previous work on learning general t-wise MRFs runs in time $n^{\Omega(d)}$ (where d is the underlying degree of the graph) and does not output a function f that can generate an approximation to

the distribution in statistical distance, *even for the special case of* $t = 3$. We give evidence that the $n^{O(t)}$ dependence in our running time is nearly optimal by applying a simple reduction from the problem of learning sparse parities with noise on t variables to learning t-wise MRFs due to Bresler et al. [BGS14]. (learning sparse parities with noise is a notoriously difficult challenge in theoretical computer science). Bresler [Bre15] observed that even for the simplest possible Ising model where the graph has a single edge, beating $O(n^2)$ run-time corresponds to fast algorithms for the well-studied *light bulb* problem [Val88], for which the best known algorithm runs in time $O(n^{1.62})$ [Val15].

Moreover, our algorithm is easy to implement, has only one tunable parameter, and works in an on-line fashion. The algorithm– the *Sparsitron*– solves the problem of learning a sparse Generalized Linear Model. That is, given examples $(X, Y) \in [-1, 1]^n \times [0, 1]$ drawn from a distribution $\mathcal{D}$ with the property that $\mathbb{E}[Y|X = x] = \sigma(w \cdot x)$ for some monotonic, Lipschitz σ and unknown w with $\|w\|_1 \leq \lambda$, the *Sparsitron* efficiently outputs a w' such that $\sigma(w' \cdot x)$ is close to $\sigma(w \cdot x)$ in ℓ_2 and has sample complexity $O(\lambda^2 \log n)$.

1.2 Our Approach

For a graph $G = (V, E)$ on n vertices, let $C_t(G)$ denote all cliques of size at most t in G. We use the Hammersley-Clifford characterization of Markov random fields and define a binary t-wise Markov random field on G to be a distribution $\mathcal{D}$ on $\{1, -1\}^n$ where

$$\Pr_{Z \sim \mathcal{D}}[Z = z] \propto \exp\left(\sum_{I \in C_t(G)} \psi_I(z)\right),$$

and each $\psi_I : \mathbb{R}^n \to \mathbb{R}$ is a function that depends only on the variables in I.

For ease of exposition, we will continue with the case of $t = 2$, the Ising model, and subsequently describe the extension to larger values of t. Let $\sigma(z)$ denote the *sigmoid* function. That is $\sigma(z) = 1/1 + e^{-z}$. Since $t = 2$, we have

$$\Pr[Z = z] \propto \exp\left(\sum_{i \neq j \in [n]} A_{ij} z_i z_j + \sum_i \theta_i z_i\right)$$

for a weight matrix $A \in \mathbb{R}^{n \times n}$ and $\theta \in \mathbb{R}^n$; here, a weight $A_{ij} \neq 0$ if and only if $\{i, j\}$ is an edge in the underlying dependency graph. For a node Z_i, it is easy to see that the probability $Z_i = -1$ conditioned on any setting of the remaining nodes to some value $x \in \{-1, 1\}^{[n]\setminus\{i\}}$ is equal to $\sigma(w \cdot x + \theta)$ where $w \in \mathbb{R}^{[n]\setminus\{i\}}$, $w_j = -2A_{ij}$, $\theta = -\theta_i$.

As such, if we set $X \equiv (Z_j : j \neq i)$ and $Y = (1 - Z_i)/2$, then the conditional expectation of Y given X is *equal* to a sigmoid with unknown weight vector w and threshold θ_i. We can now rephrase our original *unsupervised* learning task as the following *supervised* learning problem: Given random examples (X, Y) with conditional mean function $\mathbb{E}[Y|X = x] = \sigma(w \cdot x + \theta)$, recover w and θ.

Learning a conditional mean function of the form $u(w \cdot x)$ with fixed, known u is *precisely* the problem of learning a *Generalized Linear Model* or GLM and has been studied extensively in machine learning. The first provably efficient algorithm for learning GLMs where u is both monotone and Lipschitz was given by Kalai and Sastry [KS09], who called their algorithm the "Isotron." Their result was simplified and extended by Kakade, Kalai, Kanade, and Shamir [KKKS11] who introduced the "GLMtron."

Notice that $\sigma(z)$ is both monotone and 1-Lipschitz. Therefore, directly applying the GLMtron in our setting will result in a w' and θ' such that

$$\mathbb{E}[(\sigma(w' \cdot x + \theta') - \sigma(w \cdot x + \theta))^2] \leq \varepsilon. \tag{1.1}$$

Unfortunately, the sample complexity of the GLMtron depends on $\|w\|_2$, which results in sub-optimal bounds on sample complexity. We desire sample complexity dependent on $\|w\|_1$, essentially the *sparsity* of w. In addition, we need an *exact recovery* algorithm. That is, we need to ensure that w' itself is close to w in ℓ_1 and not just that the ℓ_2-*error* as in Equation 1.1 is small. We address these two challenges next.

Our algorithm, the *Sparsitron*, uses a multiplicative-weight update rule for learning w, as opposed to the GLMtron or Isotron, both of which use additive update rules. This enables us to achieve essentially optimal sample complexity. The Sparsitron is simple to describe (see Algorithm 2) and depends on only one parameter λ, the upper bound on the ℓ_1-norm. Its analysis only uses a regret bound from the classic Hedge algorithm due to Freund and Schapire [FS97].

Although the Sparsitron finds a w' such that $\mathbb{E}_X[(\sigma(w' \cdot X + \theta') - \sigma(w \cdot X + \theta))^2]$ is small, we still must prove that w' is actually close to w. Achieving such strong recovery guarantees for arbitrary distributions is typically a much harder problem (and can be provably hard in some cases for related problems [FGKP09, GR09]). In our case, we exploit the nature of MRFs by a clean property of such distributions: Call a distribution $\mathcal{D}$ on $\{1, -1\}^n$ δ-*unbiased* if each variable Z_i is 1 or -1 with probability at least δ conditioned on any setting of the other variables. Under typical necessary conditions of an MRF, the resulting distribution Z is δ-unbiased for a non-negligible δ. We show that for such δ-unbiased distributions achieving reasonably small ℓ_2-error as in Equation 1.1 implies that the recovered coefficient w' is in fact close to w.

To obtain our results for learning t-wise Markov random fields, we generalize the above approach to handle functions of the form $\sigma(p(x))$ where p is a degree t multilinear polynomial. Sparsitron can be straightforwardly extended to handle low-degree polynomials by *linearizing* such polynomials (i.e., working in the (n^t)-dimensional space of coefficients). We then have to show that achieving small ℓ_2-error - $\mathbb{E}_X[(\sigma(p(X)) - \sigma(q(X)))^2] \ll 1$ - implies that the polynomials p, q are close. This presents several additional technical challenges; still, in a self-contained proof, we show this holds whenever the underlying distribution is δ-unbiased as is the case for MRFs.

1.3 Best-Experts Interpretation of Our Algorithm

Our algorithm can be viewed as a surprisingly simple weighted voting scheme (a.k.a. "Best-Experts" strategy) to uncover the underlying graph structure $G = (\{v_1, \ldots, v_n\}, E)$ of a Markov random field. Consider an Ising model where for a fixed vertex v_i, we want to determine v_i's neighborhood and edge weights. Let $Z = (Z_1, \ldots, Z_n)$ denote random draws from the Ising model.

- Initially, all vertices $v_j (j \neq i)$ could be neighbors. We create a vector of "candidate" neighbors of length $2n-2$ with entries $(j, +)$ and $(j, -)$ for all $j \neq i$. Intuitively, since we do not know if node v_j will be negatively or positively correlated with v_i, we include two candidate neighbors, $(j, +), (j, -)$ to cover the two cases.

- At the outset, every candidate is equally likely to be a neighbor of v_i and so receives an initial *weight* of $1/(2n-2)$. Now consider a random draw from the Ising model $Z = (Z_1, \ldots, Z_n)$. For each $j \neq i$ we view each Z_j (and its negation -Z_j) as the *vote* of $(j, +)$ for the value Z_i

(respectively of $(j, -)$). The overall *prediction* p of our candidates is equal to a weighted sum of their votes (we always assume the weights are non-negative and normalized appropriately).

- For candidate neighbor v_j let the *penalty* of the prediction p (as motivated by the conditional mean function) be equal to $\ell_j = (\sigma(-2p) - (1 - Z_i)/2)Z_j$. Each candidate v_j's weight is simply multiplied by β^{ℓ_j} (for some suitably chosen *learning rate* β.[1]) It is easy to see that candidates who predict Z_i *correctly* will be penalized *less* than neighbors whose predictions are incorrect.

Remarkably, the weights of this algorithm will converge to the weights of the underlying Ising model, and the rate of this convergence is optimal. Weights of vertices that are not neighbors of v_i will rapidly decay to zero.

For clarity, we present the updates for a single iteration of our Sparsitron algorithm applied to Ising model in Algorithm 1. The iterative nature of the algorithm is reminiscent of algorithms such as belief propagation and stochastic gradient descent that are commonly used in practice. Exploring connections with these algorithms (if any) is an intriguing question.

Algorithm 1 Updates for SPARSITRON applied to learning Ising models

Initialize $W_{ij}^+ = W_{ij}^- = 1/2(n-1)$ and $\hat{A}_{ij} = 0$ for $i \neq j$.
PARAMETERS: *Sparsity bound* λ.

1: **for** each new example $(Z_1, \ldots, Z_n)$ **do**:
2: Compute the current *predictions*: $p_i = \sum_{j \neq i} \hat{A}_{ij} Z_j$ for all i.
3: **for** each $i \neq j$ **do**
4: Compute the penalties: Set $\ell_{ij} = (\sigma(-2p_i) - (1 - Z_i)/2) \cdot Z_j$.
5: Update the weights: Set $W_{ij}^+ = W_{ij}^+ \cdot \beta^{\ell_{ij}}$; $W_{ij}^- = W_{ij}^- \cdot \beta^{-\ell_{ij}}$.
6: **for** each $i \neq j$ **do**
7: Compute edge weights: $\hat{A}_{ij} = \frac{\lambda}{\sum_{\ell \neq i}(W_{i\ell}^+ + W_{i\ell}^-)} \cdot \left(W_{ij}^+ - W_{ij}^-\right)$.

1.4 Organization

We begin by describing the Sparsitron algorithm for learning sparse generalized models and prove its correctness. We then show, given a hypothesis output by the Sparsitron, how to recover the underlying weight vector *exactly* under δ-unbiased distributions. For ease of exposition, we begin by assuming that we are learning an Ising model.

We then describe how to handle the more general case of learning t-wise MRFs. This requires working with multilinear polynomials, and studying their behavior (especially, how small they can be) under δ-unbiased distributions.

2 Preliminaries

We will use the following notations and conventions.

[1] For our analysis, the learning rate can be set using standard techniques, e.g., $\beta = 1 - \sqrt{\log n/T}$ when processing T examples.

- For a vector $x \in \mathbb{R}^n$, $x_{-i} \in \mathbb{R}^{[n]\setminus\{i\}}$ denotes $(x_j : j \neq i)$.
- We write multilinear polynomials $p : \mathbb{R}^n \to \mathbb{R}$ as $p(x) = \sum_I \hat{p}(I) \prod_{i\in I} x_i$; in particular, $\hat{p}(I)$ denotes the coefficient of the monomial $\prod_{i\in I} x_i$ in the polynomial. Let $\|p\|_1 = \sum_I |\hat{p}(I)|$.
- Let $\partial_i p(x) = \sum_{J: J \not\ni i} \hat{p}(J \cup \{i\}) \prod_{j\in J} x_j$ denote the partial derivative of p with respect to x_i.
- We say $I \subseteq$ is a *maximal monomial* of p if $\hat{p}(J) = 0$ for all $J \supset I$ (i.e., there is no non-zero monomial that strictly contains I).

3 Learning Sparse Generalized Linear Models

We first describe our *Sparsitron* algorithm for learning sparse GLMs. In the next section we show how to learn MRFs using this algorithm. The main theorem of this section is the following:

Theorem 3.1. *Let $\mathcal{D}$ be a distribution on $[-1,1]^n \times \{0,1\}$ where for $(X,Y) \sim \mathcal{D}$, $E[Y|X = x] = u(w \cdot x)$ for a non-decreasing 1-Lipschitz function $u : \mathbb{R} \to [0,1]$. Suppose that $\|w\|_1 \leq \lambda$ for a known $\lambda \geq 0$. Then, there exists an algorithm that for all $\varepsilon, \delta \in [0,1]$ given $T = O(\lambda^2(\ln(n/\varepsilon)/\varepsilon^2) \cdot (\ln(1/\delta)))$ independent examples from $\mathcal{D}$, produces a vector $v \in \mathbb{R}^n$ such that with probability at least $1-\delta$,*

$$\mathop{\mathbb{E}}_{(X,Y)\leftarrow\mathcal{D}}[(u(v \cdot X) - u(w \cdot Y))^2] \leq \varepsilon. \tag{3.1}$$

The run-time of the algorithm is $O(nT)$. Moreover, it can be run in an online manner.

Proof. We assume without loss of generality that $w_i \geq 0$ for all i and that $\|w\|_1 = \lambda$; if not, we can map examples (x,y) to $((x,-x,0),y)$ and work in the new space. For any vector $v \in \mathbb{R}^n$, let $\varepsilon(v) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}[(u(v \cdot X) - u(w \cdot X))^2]$. Let $\mathbf{1}$ denote the all 1's vector.

Our approach is to use the regret bound for the *Hedge* algorithm of Freund and Schapire [FS97]. Let $T \geq 1$, $\beta \in [0,1]$ be parameters to be chosen later and $M = C'(T \ln T)/(\ln n)$ for a constant C' to be chosen later. The algorithm is shown in Algorithm 2. The inputs to the algorithm are $T + M$ independent examples $(x^1, y^1,), \ldots, (x^T, y^T)$ and $(a^1, b^1), \ldots, (a^M, b^M)$ drawn from $\mathcal{D}$.

Algorithm 2 SPARSITRON

Initialize $w^0 = \mathbf{1}/n$.
for $t = 1, \ldots, T$ **do**
  Let $p^t = w^{t-1}/\|w^{t-1}\|_1$.
  Define $\ell^t \in \mathbb{R}^n$ by $\ell^t = (1/2)(\mathbf{1} + (u(\lambda p^t \cdot x^t) - y^t)x^t)$.
  Update the weight vectors w^t: for each $i \in [n]$, set $w_i^t = w_i^{t-1} \cdot \beta^{\ell_i^t}$.
for $t = 1, \ldots, T$ **do**
  Compute the *empirical error*

$$\hat{\varepsilon}(\lambda p^t) = (1/M) \sum_{j=1}^{M} \left(u(\lambda p^t \cdot a^j) - b^j\right)^2.$$

RETURN $v = \lambda p^j$ for $j = \arg\min_{t\in[T]} \hat{\varepsilon}(\lambda p^t)$.

[2] We next analyze our algorithm and show that for suitable parameters β, T, M, it achieves a guarantee as in Equation 3.2 with probability at least $1/2$. We can then repeat the algorithm for $\log(1/\delta)$ times to get the high probability guarantee.

Observe that $\ell^t \in [0,1]^n$ and associate each $i = 1, \ldots, n$ with an expert and then apply the analysis of Freund and Schapire (c.f. [FS97], Theorem 5). In particular, setting $\beta = 1/(1 + \sqrt{T/(\ln n)})$, we get that

$$\sum_{t=1}^{T} p^t \cdot \ell^t \leq \min_{i \in [n]} \sum_{t=1}^{T} \ell_i^t + O(\sqrt{T \ln n} + (\ln n)). \tag{3.2}$$

Now, for a fixed $(x^1, y^1), \ldots, (x^{t-1}, y^{t-1})$, taking expectation with respect to (x^t, y^t), we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x^t,y^t)} \left[(p^t - (1/\lambda)w) \cdot \ell^t\right] &= (1/2) \mathop{\mathbb{E}}_{(x^t,y^t)} \left[(p^t - (1/\lambda)w) \cdot (u(\lambda p^t \cdot x^t) - y^t) x^t\right] \\
&= (1/2\lambda) \mathop{\mathbb{E}}_{x^t} \left[(\lambda p^t \cdot x^t - w \cdot x^t)(u(\lambda p^t \cdot x^t) - y^t)\right] \\
&\geq (1/2\lambda) \mathop{\mathbb{E}}_{x^t} \left[(u(\lambda p^t \cdot x^t) - u(w \cdot x^t))^2\right] \\
&\quad \text{(for all } a, b \in \mathbb{R},\ (a-b)(u(a) - u(b)) \geq (u(a) - u(b))^2\text{)}. \\
&= (1/2\lambda) \cdot \varepsilon(\lambda p^t).
\end{aligned}$$

Therefore, for a fixed $(x^1, y^1), \ldots, (x^{t-1}, y^{t-1})$, we have

$$(1/2\lambda) \mathop{\mathbb{E}}_{(x^t,y^t)} [\varepsilon(\lambda p^t)] \leq \mathop{\mathbb{E}}_{(x^t,y^t)} [p^t \cdot \ell^t - (1/\lambda) w \cdot \ell^t].$$

Plugging this into Equation 3.2, and taking expectation with respect to $(x^1, y^1), \ldots, (x^T, y^T)$, we get

$$(1/2\lambda) \sum_{t=1}^{T} \mathbb{E}[\varepsilon(\lambda p^t)] \leq \mathbb{E}\left[\min_{i \in [n]} \sum_{t=1}^{T} \ell_i^t - \sum_{t=1}^{T} (1/\lambda) w \cdot \ell^t\right] + O(\sqrt{T \ln n} + (\ln n))$$

Now, let $L = \sum_{t=1}^{T} \ell^t$. Then,

$$\min_{i \in [n]} \sum_{t=1}^{T} \ell_i^t - \sum_{t=1}^{T} (1/\lambda) w \cdot \ell^t = \min_{i \in [n]} L_i - (w/\lambda) \cdot L \leq 0,$$

where the last inequality follows as $\|w\|_1 = \lambda$. Therefore,

$$\sum_{t=1}^{T} \mathbb{E}[\varepsilon(\lambda p^t)] \leq O(\lambda) \cdot (\sqrt{T \ln n} + (\ln n)).$$

In particular, for $T \geq \ln n$,

$$\mathbb{E}\left[\min_{t \in [T]} \varepsilon(\lambda p^t)\right] \leq O(\lambda) \left(\sqrt{\frac{\ln n}{T}} + \frac{\ln n}{T}\right) = O(\lambda) \cdot \sqrt{\frac{\ln n}{T}}.$$

[2] We add the **1** in Step 4 of Algorithm 2 to be consistent with [FS97] who work with loss vectors in $[0,1]^n$.

Therefore, by Markov's inequality, for a sufficiently big constant $C > 0$,

$$\mathsf{Pr}\left[\min_{t\in[T]} \varepsilon(\lambda p^t) \geq C\lambda\left(\sqrt{\frac{\ln n}{T}}\right)\right] \leq 1/4.$$

Further, by applying Fact 3.2 with $\rho = 1/4T$ and $\gamma = \sqrt{\frac{\ln n}{T}}$, for $M = C'(T \ln T)/(\ln n)$ we get that with probability at least $3/4$, for all $t \in [T]$, $|\varepsilon(\lambda p^t) - \hat{\varepsilon}(\lambda p^t)| \leq \gamma$. Combining the above, we get that for the vector v returned by the algorithm, for a sufficiently big constant C'',

$$\mathsf{Pr}\left[\varepsilon(v) \geq C''\lambda\left(\sqrt{\frac{\ln n}{T}}\right)\right] \leq 1/2.$$

We now set $T = C'''\lambda^2(\ln n)/\varepsilon^2$ for a sufficiently big constant C''', so that $\mathsf{Pr}[\varepsilon(v) \geq \varepsilon] \leq 1/2$. Note that the number of examples is $T + M = O(\lambda^2(\ln(n/\varepsilon))/\varepsilon^2)$. The claim now follows by repeating the algorithm $O(\ln(1/\delta))$ times and choosing the one with best empirical performance. □

Fact 3.2 (Chernoff Bound)**.** *There exists a constant $C > 0$ such that the following holds. Let $v \in \mathbb{R}^n$ and let $(a^1, b^1), \ldots, (a^M, b^M)$ be independent examples from $\mathcal{D}$. Then, for all $\rho, \gamma \geq 0$, and $M \geq C\ln(1/\rho)/\gamma^2$,*

$$\mathsf{Pr}\left[\left|(1/M)\left(\sum_{j=1}^{M}(u(v \cdot a^j) - b^j)^2\right) - \varepsilon(v)\right| \geq \gamma\right] \leq \rho.$$

4 Recovering affine functions from ℓ_2 minimization

In this section we show that running the Sparsitron algorithm with sufficiently low error parameter ε will result in an ℓ_∞ approximation to the unknown weight vector. We will use this strong approximation to reconstruct the dependency graphs of Ising models as well as the edge weights.

Our analysis relies on the following important definition:

Definition 4.1. *We say distribution $\mathcal{D}$ on $\{1, -1\}^n$ is δ-unbiased if for any $i \in [n]$, and any partial assignment x to $(X_j : j \neq i)$,*

$$\min(\mathsf{Pr}[X_i = 1 | X_{-i} = x], Pr[X_i = -1 | X_{-i} = x]) \geq \delta.$$

We will use the following elementary property of sigmoid.

Claim 4.2. *For $a, b \in R$,*

$$|\sigma(a) - \sigma(b)| \geq e^{-|a|-3} \cdot \min(1, |a - b|).$$

Proof. Fix $a \in R$ and let $\gamma = \min(1, |a - b|)$. Then, since σ is monotonic

$$|\sigma(a) - \sigma(b)| \geq \min(\sigma(a + \gamma) - \sigma(a), \sigma(a) - \sigma(a - \gamma)).$$

Now, it is easy to check by a case-analysis that for all $a, a' \in \mathbb{R}$,

$$|\sigma(a) - \sigma(a')| \geq \min(\sigma'(a), \sigma'(a')) \cdot |a - a'|.$$

Further, for any t, $\sigma'(t) = 1/(2 + e^t + e^{-t}) \geq e^{-|t|}/4$. Combining the above two, we get that

$$\sigma(a+\gamma) - \sigma(a) \geq (1/4)\min(e^{-|a+\gamma|}, e^{-|a|}) \cdot \gamma \geq (1/4)e^{(-|a|-\gamma)}\gamma.$$

Similarly, we get

$$\sigma(a) - \sigma(a-\gamma) \geq 4\min(e^{-|a-\gamma|}, e^{-|a|}) \cdot \gamma \geq (1/4)e^{(-|a|-\gamma)}\gamma.$$

The claim now follows by substituting $\gamma = \min(1, |a-b|)$ (and noting that $1/4 \geq e^{-2}$). □

Lemma 4.3. *Let D be a δ-unbiased distribution on $\{1,-1\}^n$. Suppose that for two vectors $v, w \in \mathbb{R}^n$ and $\alpha, \beta \in \mathbb{R}$, $\mathbb{E}_{X\sim D}[(\sigma(w \cdot X + \alpha) - \sigma(v \cdot X + \beta))^2] \leq \varepsilon$ where $\varepsilon < \delta \cdot \exp(-2\|w\|_1 - 2|\alpha| - 6)$. Then,*

$$\|v - w\|_\infty \leq O(1) \cdot e^{\|w\|_1 + |\alpha|} \cdot \sqrt{\varepsilon/\delta}.$$

Proof. For brevity, let $p(x) = w \cdot x + \alpha$, and $q(x) = v \cdot x + \beta$. Fix an index $i \in [n]$. Then, for any X, by Claim 4.2,

$$|\sigma(p(X)) - \sigma(q(X))| \geq e^{-\|w\|_1 - |\alpha| - 3} \cdot \min\left(1, |p(X) - q(X)|\right).$$

Let $X^{i,+} \in \{1,-1\}^n$ (respectively $X^{i,-}$) denote the vector obtained from X by setting $X_i = 1$ (respectively $X_i = -1$). Note that $p(X^{i,+}) - p(X^{i,-}) = 2w_i$ and $q(X^{i,+}) - q(X^{i,-}) = 2v_i$. Therefore,

$$p(X^{i,+}) - q(X^{i,+}) - (p(X^{i,-}) - q(X^{i,-})) = 2(w_i - v_i).$$

Thus,

$$\max\left(\left|p(X^{i,+}) - q(X^{i,+})\right|, \left|p(X^{i,-}) - q(X^{i,-})\right|\right) \geq |w_i - v_i|.$$

Therefore, for any fixing of X_{-i}, as X is δ-unbiased,

$$\Pr_{X_i | X_{-i}}\left[|p(X) - q(X)| \geq |w_i - v_i|\right] \geq \delta.$$

Hence, combining the above inequalities,

$$\varepsilon \geq \mathop{\mathbb{E}}_{X}\left[(\sigma(p(X)) - \sigma(q(X)))^2\right] \geq e^{-2\|w\|_1 - 2|\alpha| - 6} \cdot \delta \cdot \min\left(1, |w_i - v_i|^2\right).$$

As $\varepsilon < e^{-2\|w\|_1 - 2|\alpha| - 6}\delta$, the above inequality can only hold if $|w_i - v_i| < 1$ so that

$$|w_i - v_i| < e^{\|w\|_1 + |\alpha| + 3} \cdot \sqrt{\varepsilon/\delta}.$$

The claim now follows.

□

5 Learning Ising Models

Definition 5.1. *Let $A \in \mathbb{R}^{n\times n}$ be a* weight matrix *and $\theta \in \mathbb{R}^n$ be a* mean-field *vector. The associated n-variable* Ising model *is a distribution $\mathcal{D}(A, \theta)$ on $\{1,-1\}^n$ given by the condition*

$$\Pr_{Z \leftarrow \mathcal{D}(A,\theta)}[Z = z] \propto \exp\left(\sum_{i \neq j \in [n]} A_{ij} z_i z_j + \sum_i \theta_i z_i\right).$$

We define $\lambda(A, \theta) = \max_i(\sum_j |A_{ij}| + |\theta_i|)$ to be the width *of the model.*

We give a simple, sample-efficient, and online algorithm for recovering the parameters of an Ising model.

Theorem 5.2. *Let $\mathcal{D}(A,\theta)$ be an n-variable Ising model with width $\lambda(A,\theta) \le \lambda$. There exists an algorithm that given λ, $\varepsilon, \rho \in (0,1)$, and $N = O(\lambda^2 \exp(O(\lambda))/\varepsilon^4)\cdot(\log(n/\varepsilon))(\log(n/\rho))$ independent samples $Z^1, \dots, Z^N \leftarrow \mathcal{D}(A,\theta)$ produces $\hat{A}$ such that with probability at least $1-\rho$,*

$$\|A - \hat{A}\|_\infty \le \varepsilon.$$

The run-time of the algorithm is $O(n^2 N)$. Moreover, the algorithm can be run in an online manner.

Proof. The starting point for our algorithm is the following observation. Let $Z \leftarrow \mathcal{D}(A,\theta)$. Then, for any $i \in [n]$ and any $x \in \{1,-1\}^{[n]\setminus\{i\}}$,

$$\mathsf{Pr}[Z_i = -1 | Z_{-i} = x] = \frac{1}{1+\exp(2\sum_{j\neq i} A_{ij}x_j + \theta_i)} = \sigma(w(i)\cdot x + \theta_i),$$

where we define $w(i) \in R^{[n]\setminus\{i\}}$ with $w(i)_j = -2A_{ij}$ for $j \neq i$. This allows us to use our Sparsitron algorithm for learning GLMs.

For simplicity, we describe our algorithm to infer the coefficients A_{nj} for $j \neq n$; it extends straightforwardly to recover the weights $\{A_{ij} : j \neq i\}$ for each i. Let $Z \leftarrow \mathcal{D}(A,\theta)$ and let $X \equiv (Z_1, \dots, Z_{n-1}, 1)$, and $Y = (1-Z_n)/2$. Then, from the above we have that

$$\mathbb{E}[Y|X] = \sigma(w(n)\cdot X),$$

where $w(n) \in \mathbb{R}^n$ with $w(n)_j = -2A_{nj}$ for $j < n$, and $w(n)_n = \theta_i$. Note that $\|w(n)\|_1 \le 2\lambda$. Further, σ is a monotone 1-Lipschitz function. Let $\gamma \in (0,1)$ be a parameter to be chosen later. We now apply the Sparsitron algorithm to compute a vector $v(n) \in \mathbb{R}^n$ so that with probability at least $1-\rho/n$,

$$\mathbb{E}[(\sigma(w(n)\cdot X) - \sigma(v(n)\cdot X))^2] \le \gamma. \tag{5.1}$$

We set $\hat{A}_{nj} = -(v(n)_j)/2$ for $j < n$. We next argue that Equation 5.1 in fact implies $\|w(n) - v(n)\|_\infty \ll 1$. To this end, we will use the following easy fact (see e.g. Bresler [Bre15]):

Fact 5.3. *For $Z \leftarrow \mathcal{D}(A,\theta)$, $i \in [n]$, and any partial assignment x to Z_{-i},*

$$\min\left(\mathsf{Pr}[Z_i = -1 | Z_{-i} = x], \mathsf{Pr}[Z_i = 1 | Z_{-i} = x]\right) \ge (1/2)e^{-2\lambda(A,\theta)} \ge (1/2)e^{-2\lambda}.$$

That is, the distribution Z is δ-unbiased for $\delta = (1/2)e^{-2\lambda}$. Note that $w(n)\cdot X = \sum_{j<n} w(n)_j Z_j + w(n)_n$ and $v(n)\cdot X = \sum_{j<n} v(n)_j Z_j + v(n)_n$. Therefore, as $(Z_1, \dots, Z_{n-1})$ is δ-unbiased, by Lemma 4.3 and Equation 5.1, we get

$$\max_{j<n} |v(n)_j - w(n)_j| \le O(1)\exp(2\lambda)\cdot\sqrt{\gamma/\delta},$$

if $\gamma \le c\delta\cdot\exp(-4\lambda) \le c\exp(-5\lambda)$ for a sufficiently small c. Thus, if we set $\gamma = c'\exp(-5\lambda)\varepsilon^2$ for a sufficiently small constant c', then we get

$$\max_{j<n} |A_{nj} - \hat{A}_{nj}| = (1/2)\|v(n) - w(n)\|_\infty \le \varepsilon.$$

By a similar argument for $i = 1, \ldots, n-1$ and taking a union bound, we get estimates $\hat{A}_{ij}$ for all $i \neq j$ so that with probability at least $1 - \rho$,

$$\max_{i \neq j} |A_{ij} - \hat{A}_{ij}| \leq \varepsilon.$$

Note that by Theorem 3.1, the number of samples needed to satisfy Equation 5.1 is

$$O((\lambda/\gamma)^2 \cdot (\log(n/\gamma))(\log(n/\rho))) = O(\lambda^2 \exp(10\lambda)/\varepsilon^4) \cdot (\log(n/\varepsilon))(\log(n/\rho)).$$

This proves the theorem. □

The theorem immediately implies an algorithm for recovering the dependency graph of an Ising model with nearly optimal sample complexity.

Corollary 5.4. *Let $\mathcal{D}(A, \theta)$ be an n-variable Ising model with width $\lambda(A, \theta) \leq \lambda$ and each non-zero entry of A at least $\eta > 0$ in absolute value. There exists an algorithm that given λ, $\eta, \rho \in (0, 1)$, and $N = O(\exp(O(\lambda))/\eta^4) \cdot (\log(n/\eta)) \log(n/\rho)$ independent samples $Z^1, \ldots, Z^N \leftarrow \mathcal{D}(A, \theta)$ recovers the underlying dependency graph of $\mathcal{D}(A, \theta)$ with probability at least $1 - \rho$. The run-time of the algorithm is $O(n^2 N)$. Moreover, the algorithm can be run in an online manner.*

Proof. The claim follows immediately from Theorem 5.2 by setting $\varepsilon = \eta/2$ to compute $\hat{A}$ and taking the edges E to be $E = \{\{i, j\} : |\hat{A}_{ij}| \geq \eta/2\}$. □

It is instructive to compare the upper bounds from Corollary 5.4 with known, unconditional lower bounds on the sample complexity of learning Ising models with n vertices due to Santhanam and Wainwright [SW12]. They prove that, even if the weights of the underlying graph are known, any algorithm for learning the graph structure must use $\Omega(\frac{2^{\lambda/4} \cdot \log n}{\eta \cdot 2^{3\eta}})$ samples. Hence, the sample complexity of our algorithm is near the best-known information-theoretic lower bound.

6 Recovering polynomials from ℓ_2 minimization

We next prove that for any polynomial $p : \mathbb{R}^n \to \mathbb{R}$, minimizing the ℓ_2-loss with respect to a sigmoid under a δ-unbiased distribution $\mathcal{D}$ also implies closeness as a polynomial. That is, if we find a polynomial $q : \mathbb{R}^n \to \mathbb{R}$, such that $\mathbb{E}_{X \sim \mathcal{D}}[(\sigma(p(X)) - \sigma(q(X)))^2]$ is sufficiently small, then $\|p - q\|_1 \ll 1$. The exact bound depends on the ℓ_1-norm of p and the degree of the polynomial. This will allow us to recover the underlying graph from a graphical model when combined with Sparsitron.

The main claim of this section is

Lemma 6.1. *Let D be a δ-unbiased distribution on $\{1, -1\}^n$. Let p, q be two multilinear polynomials $p, q : \mathbb{R}^n \to \mathbb{R}$ of degree t such that $\mathbb{E}_{X \sim D}[(\sigma(p(x)) - \sigma(q(x))^2] \leq \varepsilon$ where $\varepsilon < e^{2\|p\|_1 + 6} \delta^t$. Then,*

$$\|p - q\|_1 = O(1) \cdot (2t)^t e^{\|p\|_1} \cdot \sqrt{\varepsilon/\delta^t} \cdot \binom{n}{t}.$$

The proof follows the same outline as that of Lemma 4.3 but is a bit more subtle. To start with, we need the following property of δ-unbiased distributions which says that low-degree polynomials are not too small with non-trivial probability (aka *anti-concentration*) under δ-unbiased distributions.

Lemma 6.2. *There is a constant $c > 0$ such that the following holds. Let D be a δ-unbiased distribution on $\{1,-1\}^n$. Then, for any multilinear polynomial $s : \mathbb{R}^n \to \mathbb{R}$, and any maximal monomial $I \neq \emptyset \subseteq [n]$ in s,*

$$\Pr_{X\sim D}[|s(X)| \geq |\hat{s}(I)|] \geq \delta^{|I|}.$$

Proof. We prove the claim by induction on $|I|$. For an $i \in [n]$, Let $X^{i,+} \in \{1,-1\}^n$ (respectively $X^{i,-}$) denote the vector obtained from X by setting $X_i = 1$ (respectively $X_i = -1$).

Suppose $I = \{i\}$ so that $s(X) = \hat{s}(\{i\})X_i + s'(X_{-i})$ for some polynomial s' that only depends on X_{-i}. Then, for any $i \in [n]$, $\max(|s(X^{i,+})|, |s(X^{i,-})|) \geq |\hat{s}(\{i\})|$. Therefore, for any fixing of X_{-i}, as X is δ-unbiased,

$$\Pr_{X_i|X_{-i}}[|s(X)| \geq |\hat{s}(\{i\})|] \geq \delta.$$

Now, suppose $|I| = \ell \geq 2$ and that the claim is true for all polynomials and all monomials of size at most $\ell - 1$. Let $i \in I$. Then, $s(X) = X_i \cdot \partial_i(s(X_{-i})) + s'(X_{-i})$ for some polynomial s' that only depends on X_{-i}. Thus, $\max(|s(X^{i,+})|, |s(X^{i,-})|) \geq |\partial_i s(X_{-i})|$. Therefore, for any fixing of X_{-i}, as X is δ-unbiased,

$$\Pr_{X_i|X_{-i}}[|s(X)| \geq |\partial_i s(X_{-i})|] \geq \delta.$$

Now, let $J = I \setminus \{i\}$ and observe that J is a maximal monomial in $r(X_{-i}) \equiv \partial_i s(X_{-i})$ with $\hat{r}(J) = \hat{s}(I)$. Therefore, by the induction hypothesis,

$$\Pr_{X_{-i}}[|\partial_i s(X_{-i})| \geq |\hat{s}(I)|] \geq \delta^{\ell-1}.$$

Combining the last two inequalities, we get that $\Pr[|s(X)| \geq \hat{s}(I)] \geq \delta^{\ell}$. The claim now follows by induction. □

The next claim shows that under the assumptions of Lemma 6.1, the highest degree monomials of p, q are close to each other.

Lemma 6.3. *Let $\mathcal{D}$ be a δ-unbiased distribution on $\{1,-1\}^n$. Let p, q be two multilinear polynomials $p, q : \mathbb{R}^n \to \mathbb{R}$ such that $\mathbb{E}_{X\sim D}[(\sigma(p(x)) - \sigma(q(x))^2] \leq \varepsilon$ where $\varepsilon < e^{-2\|p\|_1-6}\delta^{|I|}$. Then, for every maximal monomial $I \subseteq [n]$ of $(p-q)$,*

$$|\hat{p}(I) - \hat{q}(I)| \leq e^{\|p\|_1+3} \cdot \sqrt{\varepsilon/\delta^{|I|}}.$$

Proof. Fix a maximal monomial $I \subseteq [n]$ in $(p-q)$. Now, for any X, by Claim 4.2,

$$|\sigma(p(X)) - \sigma(q(X))| \geq e^{-\|p\|_1-3} \cdot \min(1, |p(X) - q(X)|).$$

On the other hand, as X is δ-unbiased, by Lemma 6.2, with probability at least $\delta^{|I|}$, $|p(X) - q(X)| \geq |\hat{p}(I) - \hat{q}(I)|$. Therefore,

$$\varepsilon \geq \mathop{\mathbb{E}}_{X}\left[(\sigma(p(X)) - \sigma(q(X)))^2\right] \geq e^{-2\|p\|_1-6} \cdot \delta^{|I|} \cdot \min\left(1, |\hat{p}(I) - \hat{q}(I)|^2\right).$$

As $\varepsilon < e^{-2\|p\|_1-6}\delta^{|I|}$, the above inequality can only hold if $|\hat{p}(I) - \hat{q}(I)| < 1$ so that

$$|\hat{p}(I) - \hat{q}(I)| < e^{\|p\|_1+3}\sqrt{\varepsilon/\delta^{|I|}}.$$

The claim follows. □

We are ready to prove the main claim of this section - Lemma 6.1.

Proof of Lemma 6.1. For a polynomial $s : \mathbb{R}^n \to \mathbb{R}$ of degree at most t, and $\ell \leq t$, let $s_{\leq \ell}$ denote the polynomial obtained from s by only taking monomials of degree at most ℓ and let $s_{=\ell}$ denote the polynomial obtained from s by only taking monomials of degree exactly ℓ.

For brevity, let $r = p - q$, and for $\ell \leq t$, let $\rho_\ell = \|r_{=\ell}\|_1 = \|p_{=\ell} - q_{=\ell}\|_1$. We will inductively bound $\rho_t, \rho_{t-1}, \ldots, \rho_1$.

From Lemma 6.3 applied to the polynomials p, q, we immediately get that

$$\rho_t = \|r_{=t}\|_1 \leq e^{\|p\|_1+3} \cdot \sqrt{\varepsilon/\delta^t} \cdot \binom{n}{t} \equiv \varepsilon_0. \tag{6.1}$$

Now consider $I \subseteq [n]$ with $|I| = \ell$. Then, by an averaging argument, there is some fixing of the variables not in X_I so that for the polynomials p_I, q_I obtained by this fixing, and for the resulting distribution D_I on $\{1, -1\}^I$,

$$\mathop{\mathbb{E}}_{Y \sim D_I}[(\sigma(p_I(Y)) - \sigma(q_I(Y)))^2] \leq \varepsilon.$$

Note that D_I is also δ-unbiased. Therefore, by Lemma 6.3 applied to the polynomials p, q, letting $r_I = p_I - q_I$, we get that

$$|\hat{r}_I(I)| = |\hat{p_I}(I) - \hat{q}_I(I)| \leq e^{\|p\|_1+3} \cdot \sqrt{\varepsilon/\delta^{|I|}}.$$

We next relate the coefficients of r_I to that of r. As the polynomial r_I is obtained from r by fixing the variables not in I to some values in $\{1, -1\}$,

$$|\hat{r}_I(I)| \geq |\hat{r}(I)| - \sum_{J : J \supset I} |\hat{r}(J)|\,.$$

Combining the above two inequalities, we get that

$$|\hat{r}(I)| \leq e^{\|p\|_1+3} \cdot \sqrt{\varepsilon/\delta^\ell} + \sum_{J \supset I} |\hat{r}(J)|\,.$$

Summing the above equation over all I of size exactly ℓ, we get

$$\begin{aligned}
\|r_{=\ell}\|_1 = \sum_{I : |I| = \ell} |\hat{r}(I)| &\leq e^{\|p\|_1+3} \cdot \sqrt{\varepsilon/\delta^\ell} \cdot \binom{n}{\ell} + \sum_{I : |I| = \ell} \left(\sum_{J \supset I} |\hat{r}(J)| \right) \\
&\leq \varepsilon_0 + \sum_{I : |I| = \ell} \left(\sum_{J \supset I} |\hat{r}(J)| \right) \\
&= \varepsilon_0 + \sum_{j=\ell+1}^{t} \binom{j}{\ell} \cdot \left(\sum_{J : |J| = j} |\hat{r}(J)| \right) = \varepsilon_0 + \sum_{j=\ell+1}^{t} \binom{j}{\ell} \|r_{=j}\|_1.
\end{aligned}$$

Therefore, we get the recurrence,

$$\rho_\ell \leq \varepsilon_0 + \sum_{j=\ell+1}^{t} \binom{j}{\ell} \rho_j. \tag{6.2}$$

We can solve the above recurrence by induction on ℓ. Specifically, we claim that the above implies $\rho_j \le (2t)^{t-j} \cdot \varepsilon_0$. For $j = t$, the claim follows from Equation 6.1. Now, suppose the inequality holds for all $j > \ell$. Then, by Equation 6.2, as $\binom{j}{\ell} \le j^{j-\ell}$,

$$\rho_\ell \le \varepsilon_0 + \sum_{j=\ell+1}^{t} j^{j-\ell}(2t)^{t-j}\varepsilon_0 \le \varepsilon_0 + \sum_{j=\ell+1}^{t} t^{j-\ell}(2t)^{t-j}\varepsilon_0$$
$$\le t^{t-\ell} \cdot \varepsilon_0 \cdot \left(1 + \sum_{j=\ell+1}^{t} 2^{t-j}\right) = t^{t-\ell} \cdot \varepsilon_0 \cdot 2^{t-\ell}.$$

Therefore,

$$\|r\|_1 = \sum_{\ell=0}^{t} \|r_{=\ell}\|_1 \le \sum_{\ell=0}^{t} (2t)^{t-\ell}\varepsilon_0 \le \varepsilon_0 \cdot 2^{t+1}t^t.$$

The lemma now follows by plugging in the value of ε_0. □

7 Learning Markov Random Fields

We can now describe how to apply the Sparsitron algorithm to recover the parameters of binary t-wise MRFs. We in fact show how to recover the parameters of a *log-polynomial* density defined as follows:

Definition 7.1. *A distribution $\mathcal{D}$ on $\{1, -1\}^n$ is said to be a log-polynomial distribution of degree t if for some multilinear polynomial $p : \mathbb{R}^n \to \mathbb{R}$,*

$$\Pr_{X \sim \mathcal{D}}[X = x] \propto \exp(p(x)).$$

We give an efficient algorithm that given samples from a log-polynomial density recovers the underlying polynomial. We can then use this to learn MRFs via the Hammersley-Clifford theorem.

Theorem 7.2. *Let $\mathcal{D}$ be a log-polynomial distribution of degree at most t on $\{1, -1\}^n$ with the associated polynomial $p : \mathbb{R}^n \to \mathbb{R}$ such that $\max_i \|\partial_i p\|_1 \le \lambda$. There exists an algorithm that given λ, and $\varepsilon, \rho \in (0, 1)$ and*

$$N = \frac{(2t)^{O(t)} \cdot e^{O(\lambda t)}}{\varepsilon^4} \cdot (\log(n/\varepsilon))(\log(n/\rho)),$$

independent samples $Z^1, \ldots, Z^N \leftarrow \mathcal{D}$ produces a multilinear polynomial $q : \mathbb{R}^n \to \mathbb{R}$ such that with probability at least $1 - \rho$,

$$\|p - q\|_1 \le \varepsilon \cdot \binom{n}{t}.$$

Moreover, for monomials I of highest degree t, we get the stronger error bound of $|\hat{p}(I) - \hat{q}(I)| \le \varepsilon$. The run-time of the algorithm is $O(N \cdot n^t)$ and the algorithm can be run in an online manner.

Before proving the theorem we first show how it implies learning of t-wise MRFs. We will use the characterization of MRFs via the Hammersley-Clifford theorem. Given a graph $G = (V, E)$ on n vertices, let $C_t(G)$ denote all cliques of size at most t in G. A binary t-wise MRF with dependency

graph G is a distribution $\mathcal{D}$ on $\{1,-1\}^n$ where the probability density function of $\mathcal{D}$ can be written as

$$\Pr_{Z\sim\mathcal{D}}[Z=x] \propto \exp\left(\sum_{I\in\mathcal{S}} \psi_I(x)\right),$$

where $\mathcal{S} \subseteq C_t(G)$ and each $\psi_I : \mathbb{R}^n \to \mathbb{R}$ is a function that depends only on the variables in I. Note that if $t = 2$, this corresponds exactly to the Ising model. We call $\psi(x) = \sum_{I\in\mathcal{S}} \psi_I(x)$ the *factorization polynomial* of the MRF.

Note that the factorization polynomial is a polynomial of degree at most t. However, different graphs and factorizations (i.e., functions $\{\psi_i\}$) could potentially lead to the same polynomial. To get around this we enforce the following non-degeneracy condition:

Definition 7.3. *For a t-wise MRF $\mathcal{D}$ on $\{1,-1\}^n$ we say an associated dependency graph G and factorization*

$$\Pr_{Z\sim\mathcal{D}}[Z=x] \propto \exp\left(\sum_{I\in\mathcal{S}} \psi_I(x)\right),$$

for $\mathcal{S} \subseteq C_t(G)$ is η-identifiable if for every maximal monomial *J in $\psi(x) = \sum_{I\in\mathcal{S}} \psi_I(x)$, $\left|\hat{\psi}(J)\right| \geq \eta$ and every edge in G is covered by a non-zero monomial of ψ.*

We now state our main theorem for t-wise MRFs. Roughly speaking, using $M = 2^{O(\lambda t)} \cdot n^{O(t)}/\varepsilon^4$ samples, where λ is the maximum ℓ_1-norm of the derivatives of the polynomial, and run-time $O(M \cdot n^t)$, we can compute a t-wise MRF that is ε-close in pointwise-approximation to the original probability density function. Moreover, if the MRF is η-identifiable, we can recover the underlying dependency graph exactly.

Corollary 7.4. *Let $\mathcal{D}$ be a t-wise MRF on $\{1,-1\}^n$ with underlying dependency graph G and factorization polynomial $\psi(x) = \sum_{I\in C_t(G)} \psi_I(x)$ with $\max_i \|\partial_i \psi\|_1 \leq \lambda$. There exists an algorithm that given λ, and $\varepsilon, \rho \in (0, 1/2)$, and*

$$N = \frac{(2t)^{O(t)} e^{O(\lambda t)}}{\varepsilon^4} \cdot n^{4t} \cdot (\log(n/\varepsilon)) \log(n/\rho)$$

independent samples $Z^1, \ldots, Z^N \leftarrow \mathcal{D}$ produces a t-wise MRF $\mathcal{D}'$ with dependency graph H and a factorization polynomial $\varphi(x) = \sum_{I\in C_t(H)} \varphi_I(x)$ such that with probability at least $1-\rho$:

$$\forall x,\ \Pr_{Z\sim\mathcal{D}}[Z=x] = (1\pm\varepsilon) \Pr_{Z\sim\mathcal{D}'}[Z=x].$$

Moreover, if $\mathcal{D}$ is η-identifiable, then by setting $\varepsilon = \eta/2$ in the above we can ensure exact recovery of the dependency graph G. The algorithm runs in time $O(Nn^t)$ and can be run in an online manner.

Note that $\max_i \|\partial_i \psi\|_1$ is analogous to the notion of width for Ising models. Thus, exponential dependence on it is necessary as in the Ising model.

Proof. We apply Theorem 7.2 with error $\varepsilon' = \varepsilon n^{-t}$ to samples from $\mathcal{D}$ to obtain a polynomial $\varphi : \mathbb{R}^n \to \mathbb{R}$ such that $\|\psi - \varphi\|_1 \leq \varepsilon$. We build a new graph H as follows: For each monomial $I \subseteq [n]$

with $\hat{\varphi}(I) \neq 0$, add all the edges in I to H. Let $\mathcal{D}'$ denote the t-wise MRF with dependency graph H and factorization polynomial φ. Since, $\|\psi - \varphi\|_1 \leq \varepsilon$, it follows that for all x, $|\psi(x) - \varphi(x)| < \varepsilon$. Therefore, for all x,

$$\exp(\psi(x)) = \exp(\varphi(x) \pm \varepsilon) = (1 \pm 2\varepsilon)\exp(\varphi(x)).$$

The first part of the claim now follows.

For the second part, we follow a similar approach with $\varepsilon' = \eta n^{-t}/2$ to obtain a polynomial $\varphi : \mathbb{R}^n \to \mathbb{R}$ such that $\|\psi - \varphi\|_1 \leq \eta/2$. We then build a new graph H as follows: For each monomial $I \subseteq [n]$ with $|\hat{\varphi}(I)| \geq \eta/2$, add all the edges in I to H.

As we have $\left|\hat{\psi}(I) - \hat{\varphi}(I)\right| \leq \eta/2$ for every monomial I, we should in particular have $|\hat{\varphi}(I)| \geq \eta/2$ for every maximal monomial of ψ. Further, from the definition of identifiability, every edge in G is part of some maximal monomial so that we correctly recover all the edges of G and $H = G$. This completes the proof of the corollary. □

We next sketch the proof of Theorem 7.2. The proof is similar to that Theorem 5.2, wherein instead of using Sparsitron directly, we run the algorithm in an expanded feauture space where we linearize multilinear polynomials. The analysis relies on Lemma 6.1 instead of Lemma 4.3 as in the proof of Theorem 5.2.

Proof of Theorem 7.2. For each i, we will show how to recover a polynomial q_i such that $\|\partial_i p - q_i\|_1 < \varepsilon \cdot \binom{n}{t-1}$. We can then combine these polynomials to obtain a polynomial q. One way to do so is as follows: For each $I \subseteq [n]$, let $i = \arg\min(I)$, and define $\hat{q}(I) = \hat{q}_i(I \setminus \{i\})$. Then,

$$\begin{aligned}\|p - q\|_1 &= \sum_I |\hat{p}(I) - \hat{q}(I)| = \sum_{i=1}^{n} \sum_{I:\arg\min(I)=i} |\hat{p}(I) - \hat{q}(I)| \\ &\leq \sum_{i=1}^{n} \|\partial_i p - q_i\| \leq \varepsilon \cdot n \cdot \binom{n}{t-1}.\end{aligned}$$

Here we show how to find a polynomial q_n such that with probability at least $1 - \rho/n$,

$$\|\partial_n p - q_n\|_1 < \varepsilon \cdot \binom{n}{t-1}. \tag{7.1}$$

The other cases can be handled similarly and the theorem then follows from the above argument.

The starting point for our algorithm is the following observation. Let $Z \sim \mathcal{D}$. Then, for any $x \in \{1, -1\}^{n-1}$,

$$\frac{\mathsf{Pr}[Z_n = 1 | Z_{-n} = x]}{\mathsf{Pr}[Z_n = -1 | Z_{-n} = x]} = \exp(2\partial_n p(x)).$$

Thus,

$$\mathsf{Pr}[Z_n = -1 | Z_{-n} = x] = \sigma(-2\partial_n p(x)).$$

This allows us to use our Sparsitron algorithm for learning GLMs via *feature expansion*. Concretely, let $p' = -2\partial_n p$ and $\boldsymbol{p'} = (\hat{p'}(I) : I \subseteq [n-1], |I| \leq t-1)$. Similarly, for $x \in \{1, -1\}^{n-1}$, let $\boldsymbol{v}(x) = (\prod_{i \in I} x_i : I \subseteq [n-1], |I| \leq t-1)$. Let X be the distribution of $\boldsymbol{v}(x)$ and let $Y = (1 - Z_n)/2$. Then, from the above arguments, we have

$$\mathbb{E}[Y|X] = \sigma(\boldsymbol{p'} \cdot X).$$

Note that $\|\boldsymbol{p'}\|_1 = 2\|\partial_n p\|_1 \leq 2\lambda$. Let $\gamma \in (0,1)$ be a parameter to be chosen later. We now apply the Sparsitron algorithm as in Theorem 3.1 to compute a vector $\boldsymbol{q'} \in \mathbb{R}^n$ such that with probability at least $1 - \rho/n$,

$$\mathbb{E}[(\sigma(\boldsymbol{p'} \cdot X) - \sigma(\boldsymbol{q'} \cdot X))^2] \leq \gamma.$$

We define polynomial q_n by setting $\hat{q_n}(I) = (-1/2) \cdot \boldsymbol{q'}_I$ for all $I \subseteq [n-1]$. Then, the above implies that

$$\mathop{\mathbb{E}}_{Z'=(Z_1,\ldots,Z_{n-1})} \left[\left(\sigma(-2\partial_n p(Z')) - \sigma(-2q_n(Z')) \right)^2 \right] \leq \gamma. \tag{7.2}$$

Finally, just as in the Ising model, for each i, and any partial assignment x to Z_{-i},

$$\min\left(\mathsf{Pr}[Z_i = -1 | Z_{-i} = x], \mathsf{Pr}[Z_i = 1 | Z_{-i} = x]\right) =$$
$$\min\left(\sigma(-2\partial_i p(x)), 1 - \sigma(-2\partial_i p(x))\right) \geq (1/2)e^{-2\|\partial_i p\|_1} \geq (1/2)e^{-2\lambda}.$$

That is, the distribution Z is δ-unbiased for $\delta = (1/2)e^{-2\lambda}$. Therefore, by Equation 7.2, and Lemma 6.1, for $\gamma < c\exp(-4\lambda) \cdot \delta^{-t}$ for a sufficiently small constant c, we get

$$\|\partial_n p - q_n\|_1 \leq O(1)(2t)^t \cdot e^{2\lambda} \cdot \sqrt{\gamma/\delta^t} \cdot \binom{n}{t-1} \leq \varepsilon \cdot \binom{n}{t-1},$$

where $\gamma = \varepsilon^2 \cdot \exp(-C\lambda t)/C(2t)^{2t}$ for a sufficiently large constant $C > 0$. Note that by Theorem 3.1, the number of samples needed to satisfy Equation 5.1 is

$$O((\lambda/\gamma)^2 \cdot (\log(n/\gamma))(\log(n/\rho))) = \frac{(2t)^{O(t)} \cdot e^{O(\lambda t)}}{\varepsilon^4} \cdot (\log(n/\varepsilon))(\log(n/\rho))).$$

This proves Equation 7.1 and hence the main part of the theorem. The moreover part follows from looking at the proof of Lemma 6.1 and is omitted here. □

References

[AKN06] Pieter Abbeel, Daphne Koller, and Andrew Y. Ng. Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, 7:1743–1788, 2006.

[ATHW11] Animashree Anandkumar, Vincent Y. F. Tan, Furong Huang, and Alan S. Willsky. High-dimensional structure estimation in ising models: Local separation criterion. 40(3), August 20 2011. Comment: Published in at http://dx.doi.org/10.1214/12-AOS1009 the Annals of Statistics (http://www.imstat.org/aos/) by the Institute of Mathematical Statistics (http://www.imstat.org).

[BGS14] Guy Bresler, David Gamarnik, and Devavrat Shah. Structure learning of antiferromagnetic ising models. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *NIPS*, pages 2852–2860, 2014.

[BMS13] Guy Bresler, Elchanan Mossel, and Allan Sly. Reconstruction of markov random fields from samples: Some observations and algorithms. *SIAM J. Comput*, 42(2):563–578, 2013.

[Bre15] Guy Bresler. Efficiently learning ising models on arbitrary graphs. In *STOC*, pages 771–782. ACM, 2015.

[CL68] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Information Theory*, 14(11):462–467, November 1968.

[Cli90] P. Clifford. Markov random fields in statistics. In G. R. Grimmett and D. J. A. Welsh, editors, *Disorder in Physical Systems. A Volume in Honour of John M. Hammersley*, pages 19–32, Oxford, 1990. Clarendon Press.

[FGKP09] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM J. Comput.*, 39(2):606–645, 2009.

[FS97] Yoav Freund and Robert Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *JCSS: Journal of Computer and System Sciences*, 55, 1997.

[Gol95] Oded Goldreich. Three XOR-lemmas - an exposition. *Electronic Colloquium on Computational Complexity (ECCC)*, 2(56), 1995.

[GR09] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM J. Comput.*, 39(2):742–765, 2009.

[HS86] G. Hinton and T. Sejnowski. Learning and relearning in boltzmann machines. In Rummelhart and McClelland, editors, *Parallel Distributed Processing*, pages 283–335, 1986.

[JEMF06] Ariel Jaimovich, Gal Elidan, Hanah Margalit, and Nir Friedman. Towards an integrated protein-protein interaction network: A relational markov network approach. *Journal of Computational Biology*, 13(2):145–164, 2006.

[KF09] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[KFL01] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Information Theory*, 47(2):498–519, 2001.

[KKKS11] Sham M. Kakade, Adam Kalai, Varun Kanade, and Ohad Shamir. Efficient learning of generalized linear and single index models with isotonic regression. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *NIPS*, pages 927–935, 2011.

[KS01] Karger and Srebro. Learning markov networks: Maximum bounded tree-width graphs. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 2001.

[KS09] Adam Tauman Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT*, 2009.

[Lau98] S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1998.

[MRS13] Elchanan Mossel, Sébastien Roch, and Allan Sly. Robust estimation of latent tree graphical models: Inferring hidden states with inexact parameters. *IEEE Trans. Information Theory*, 59(7):4357–4373, 2013.

[NBSS12] Praneeth Netrapalli, Siddhartha Banerjee, Sujay Sanghavi, and Sanjay Shakkottai. Greedy learning of markov network structure, 2012.

[RSS12] Avik Ray, Sujay Sanghavi, and Sanjay Shakkottai. Greedy learning of graphical models with small girth. In *Allerton*, pages 2024–2031. IEEE, 2012.

[Sal09] Ruslan Salakhutdinov. Learning in markov random fields using tempered transitions. In Yoshua Bengio, Dale Schuurmans, John D. Lafferty, Christopher K. I. Williams, and Aron Culotta, editors, *NIPS*, pages 1598–1606. Curran Associates, Inc, 2009.

[SW12] Narayana P. Santhanam and Martin J. Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Trans. Information Theory*, 58(7):4117–4134, 2012.

[TR14] Rashish Tandon and Pradeep Ravikumar. Learning graphs with a few hubs. In *ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 602–610. JMLR.org, 2014.

[Val88] L. G. Valiant. Functionality in neural nets. In *Proceedings of the First Annual Workshop on Computational Learning Theory*, COLT '88, pages 28–39, San Francisco, CA, USA, 1988. Morgan Kaufmann Publishers Inc.

[Val15] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem. *J. ACM*, 62(2):13:1–13:45, May 2015.

[VMLC16] Marc Vuffray, Sidhant Misra, Andrey Y. Lokhov, and Michael Chertkov. Interaction screening: Efficient and sample-optimal learning of ising models. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *NIPS*, pages 2595–2603, 2016.

[WRL06] Martin J. Wainwright, Pradeep Ravikumar, and John D. Lafferty. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *NIPS*, pages 1465–1472. MIT Press, 2006.

A Hardness of Learning t-wise Markov random fields

Bresler, Gamarnik, and Shah [BGS14] showed how to embed parity learning as a Markov random field and showed that a restricted class of algorithms must take time $n^{\Omega(d)}$ to learn degree MRFs defined on degree d graphs. Here we use the same construction, and for completeness we give a proof. The conclusion is that, assuming the hardness of learning sparse parities with noise, for degree d graphs, learning t-wise MRFs ($t < d$) will require time $n^{\Omega(t)}$. Our positive results match this conditional lower bound.

- Let $\chi_S : \{-1,1\}^n \to \{-1,1\}^n$ be an unknown parity function on a subset S, $|S| \leq k$, of n inputs bits (i.e., $f(x) = \prod_{i \in S} x_i$). Let $\mathcal{C}_k$ be the concept class of all parity functions on subsets S of size at most k. Let $\mathcal{D}$ be the uniform distribution on $\{-1,1\}^n$.
- Fix an unknown $c \in \mathcal{C}_k$ and consider the following random experiment: x is drawn according to $\mathcal{D}$ and with probability $1/2 + \eta$ (for some constant η), the tuple $(x, c(x))$ is output. With probability $1/2 - \eta$, the tuple $(x, c(x)')$ is output where $c'(x)$ is the complement of $c(x)$.
- The k-LSPN problem is as follows: Given i.i.d. such tuples as described above, find h such that $\Pr_x[h(x) \neq c(x)] \leq \varepsilon$.

Now we reduce k-LSPN to learning the graph structure of a $(k+1)$-wise Markov random field. Let S denote the k indices of the unknown parity function Let G be a graph on $n+1$ vertices $X_1, \ldots, X_n, Y$ equal to a clique on the set of vertices corresponding to set S and vertex Y. Consider the probability distribution

$$\mathsf{Pr}[Z = (x_1, \ldots, x_n, y)] \propto \exp(\gamma \chi_S(x) y)$$

for some constant γ. A case analysis shows that $p_1 = \mathsf{Pr}[Y = \chi_S(X)] \propto e^{\gamma}$ and $p_2 = \mathsf{Pr}[Y \neq \chi_S(X)] \propto e^{-\gamma}$. Hence the ratio p_1/p_2 is approximately $1 + 2\gamma$. Since $p_1 + p_2 = 1$, by choosing γ to be a sufficiently small, we will have $p_1 \geq 1/2 + \eta$ and $p_2 \leq 1/2 - \eta$ for some small (but constant) η.

Further, it is easy to see that the parity of any subset of X_i's is unbiased. By the Vazirani XOR lemma [Gol95], this implies that the X_is are uniformly distributed. Therefore, the distribution encoded by this $(k+1)$-wise MRF is precisely the distribution described in the k-LSPN problem. If we could discover the underlying clique in the Markov random field, we would be able to learn the underlying sparse parity. Hence, learning $k+1$-MRFs is harder than k-LSPN.

The current best algorithm for learning k-LSPN is due to Valiant [Val15] and runs in time $n^{\Omega(0.8k)}$. Any algorithm running in time $n^{o(k)}$ would be a major breakthrough in theoretical computer science.