Learning Sparse Graphs via Sub-Gradient Descent

Chinmay Hegde, *Senior Member, IEEE*

***Abstract*—We consider the problem of reconstructing sparse graphs from a small number of observed cuts. This problem naturally arises in applications involving dynamically varying graphs where the goal is to learn the evolution of edge structure over a set of nodes. Since this is a combinatorial, non-convex problem, previous approaches for this problem have relied on approaches such as convex relaxation. In this paper, we describe a fast iterative algorithm for solving this problem that is a variant of projected sub-gradient descent; to our knowledge, this is the first such graph reconstruction method that exhibits both near-optimal sample complexity as well as linear convergence. As a side benefit, we also point out potentially interesting connections with learning sparse neural networks.**

***Index Terms*—Sparse recovery, graph sketching, sub-gradient descent, linear convergence.**

I. Introduction

ANALYSIS of massive graphs is central to several applications involving social networks, e-commerce, and biological networks. Unfortunately, in real-world applications, the sheer size of the underlying graphs can make them cumbersome to monitor, store, and process. These computational challenges are exacerbated if we are interested in tracking the evolution of graph structure over an extended period of time.

Fortunately, in several applications the difference graph over consecutive time instants only involves the modification of a few edges or nodes. Mathematically, such difference graphs can be assumed to be very *sparse*. This type of sparseness property has been leveraged in learning the graph evolution via a series of techniques known as *graph sketching*. The key idea is to store a short summary (or sketch) of the graph at each time instant. It is known that a small number of such sketches is sufficient to reconstruct sparse graphs [1].

One way to sketch the graph evolution is to record a small number of *cuts* of the graph chosen uniformly at random. Following the work of [2], several recent works in the machine learning literature have proposed algorithms for reconstructing sparse graphs from cut sketches, borrowing techniques from compressive sensing [3], [4], [5].

However, the aforementioned approaches suffer from two limitations. First, most such results resort to convex (L1-norm) relaxation for the reconstruction problem, which leads to high running time; this can be a problem for very large graphs. Second, sparseness assumptions implicitly assume that the graph evolution obeys the Erdős-Rényi model, i.e., that the presence/absence of edges are equally likely (with probability proportional to the cardinality of the edge set) and independent of other edges. This is not particularly suitable for modeling real-world networks; for example, the Web exhibits well-defined *hubs* (star-like subgraphs) where a single node is connected to several other nodes. Our goal in this paper is to provide an iterative graph reconstruction algorithm that achieves fast (ideally, exponential) convergence as well as can be extended to interesting graph structures beyond sparsity.

This work was supported by the National Science Foundation, a faculty fellowship from the Black and Veatch Foundation, and a GPU grant from the NVIDIA Corporation.

CH is with the Tandon School of Engineering at New York University, Brooklyn, NY 11201 USA (e-mail: chinmay.h@nyu.edu).

A. Setup

Consider a simple undirected graph $G = (V, E)$ defined over $|V| = p$ nodes. Let $W \in \mathbb{R}^{p \times p}$ be the symmetric (weighted) adjacency matrix representing the *evolution* (difference graph) over V at any given time instant. We will assume that the difference graph contains only s edges. Consider any $S \subseteq V$. The observed value of the *cut* corresponding to S, c_S, is defined as:

$$c_S = \sum_{i,j} W_{ij} + e, \quad i \in S, j \in S^c.$$

Here, e denotes any noise or error incurred during the observation procedure.

We rewrite the graph sketching problem as follows. For the i^{th} cut, define a vector $a_i \in \{\pm 1\}^p$ such that $a_i(S) = 1$ and $a_i(S^c) = -1$. Suppose that the mean value of the evolution W is zero. (If not, we can subtract the mean by assuming that the total weight of the evolution is also observed). Then, the cut observations can be reformulated as:

$$y_i = \frac{1}{2} a_i^T W a_i + e_i = \langle a_i \otimes a_i, W \rangle + e_i \quad i = 1, 2, \ldots, n. \quad (1)$$

In other words, each cut sketch can be viewed as a noisy version of the Hilbert-Schmidt inner product $y_i = \langle A_i, W \rangle$ of the adjacency matrix W with the rank-one tensor form $A_i = a_i \otimes a_i$. The goal is to accurately reconstruct the edges (and weights) of the difference graph under such an observation model. Two natural questions emerge in this context:

1) *Sample complexity*: How must the cut sets S be chosen in order to enable reconstruction of the evolution W from as few observations as possible?
2) *Computational complexity*: How can we efficiently reconstruct the evolution W?

Following [2], [3], [4], we assume that the cut sets S are chosen uniformly at random. This can be implemented by assuming that each a_i is a vector whose entries are i.i.d. Rademacher random variables with distribution $P(1) = P(-1) = 1/2$.[1]

We propose a new algorithm for reconstructing the evolution W from cut sketches of the form (1). Our method fundamentally differs from several of the aforementioned graph reconstruction methods: as opposed to using convex

[1]This scheme differs from the works [6], [7] who construct the sketch vectors by sampling from certain other distributions.

optimization, our method is non-convex purely uses simple first-order techniques. Our approach offers two key benefits: (i) the algorithm exhibits *linear* convergence, and (ii) the framework is flexible and can model several different families of graph evolutions.

The key assumption underlying our algorithm is that the support of the adjacency matrix of the difference graph (representing the set of edges that have evolved) obeys a natural *structured sparsity* model. It is reasonable that in certain applications, the edges in the difference graph must be *clustered* in some fashion. Examples of such cluster-like structure can be manifested in block-sparse innovations (i.e, only a small block of edges has changed [8]) and node-perturbed innovations (i.e., only edges from a few nodes have changed [9]).

For such cases, we prove that the underlying matrix W can be reconstructed by recording merely $n = O(s \log p)$ non-adaptive cut queries. As a consequence of our analysis, the sample complexity of our algorithm is only weakly dependent on the number of nodes in the underlying graph and scales according to the number of *evolved* edges; this can be important in very large graphs whose edges evolve slowly.

As a consequence of the above link to structured sparsity models, we can derive our algorithm exhibits $O(p^2 \text{ polylog } p)$ running time, which is much faster than the methods of [2], [3] for large p, but is slower than the active learning approach of [5] (which, unfortunately, is only applicable to sparse graphs without any additional structure). Closing this gap is an interesting direction for future research.

B. Techniques

At a high level, our algorithm merges two well-known lines of work in the literature. The first line of work includes techniques to solve the graph reconstruction problem (1) using techniques from compressive sensing [2], [3], [4]. The second line of work goes in the reverse direction, integrating techniques for combinatorial optimization into sparse recovery methods [10], [11], [12], [13], [14].

The main technical ingredient in our proof is the use of a concentration property of the cut observation operator Eq. (1), following [15], [16], [17], [6]). This property lends to a new first-order graph reconstruction algorithm that provably exhibits linear convergence. This method is (essentially) equivalent to (projected) *sub*-gradient descent, but its use does not seem to have been proposed in either the graph recovery, or the structured sparsity, literature.

C. Learning sparse polynomial neural networks

Before proceeding, let us take a brief detour to highlight a connection of the aforementioned graph reconstruction setup to a different class of learning problems. Consider a shallow (two-layer) neural network comprising p input nodes, a single hidden layer with r neurons with quadratic activation function $\sigma(z) = z^2$, first layer weights $\{w_j\}_{j=1}^r \subset \mathbb{R}^p$, and an output layer comprising of a single node and weights $\{\alpha_j\}_{j=1}^r \subset \mathbb{R}$. Then the above network is called a *polynomial neural network* [18], [19], [20], whose input-output relation is given by:

$$\hat{y} = \sum_{j=1}^{r} \alpha_j \sigma(w_j^T x) = \sum_{j=1}^{r} \alpha_j \langle w_j, x, \rangle^2 = x^T W x$$

where $W = \sum_{j=1}^r \alpha_j w_j w_j^T$ is a $p \times p$ matrix encoding the weights of the network. Therefore, if W is s-sparse, then the graph reconstruction problem 1 is similar to the problem of learning a *sparse* ground truth polynomial neural network with one hidden layer when excited with random binary inputs.

Our theory shows that only $O(s \log p)$ training samples suffice to learn the new network, and that (projected) sub-gradient descent converges at a linear rate. This suggests a considerable improvement over the results in [19], [20] — where the sample complexity scales as $\Omega(pr)$ — for the case when the underlying ground network is very sparse (this, for instance, can arise if the weights of the network are slowly evolving). A deeper dive into further connections between sparse recovery, graph sketching, and polynomial neural network learning is outside the scope of this paper but could be of potential future interest.

II. Algorithm and Analysis

A. Basics

Let us start with a somewhat more general problem formulation than what we need. Suppose $a \in \{\pm 1\}^p$ denotes a vertex of the hypercube. For each subset $J \subseteq [p]$, we define the multivariate polynomial basis functions:

$$\chi_J(a) = \begin{cases} \prod_{i \in J} a_i, & J \neq \emptyset, \\ 1, & J = \emptyset. \end{cases}$$

We assume the following generative model for our queries. We obtain data samples $(a_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$ such that

$$y_i = f(a_i).$$

Expanding in term of the polynomial basis, we can express f in terms of its (Fourier) basis coefficients

$$y_i = \sum_{J \subseteq [n]} d_J \chi_J(a).$$

In the following, we will restrict our attention to *quadratic* polynomials where $|J| \leq 2$. Observe that this setting exactly matches that of Eq. (1). Moreover, we will assume that only a few of the coefficients d_J are non-zero; specifically, the number of nonzero basis coefficients is given by $s \ll p^2$.

We re-write Equation (1) in terms of a linear operator mapping the coefficients of the polynomial to the observations. Suppose we represent the Fourier coefficients of the function f in the form of a $p^2 \times 1$ vector $z = \text{vec}(Z)$ such that

$$Z_{ij} = d_{ij}, \quad i, j \in [p].$$

Overall, the forward mapping can be written as:

$$y_i = \mathcal{A}(z) + e_i \quad i = 1, 2, \ldots, n. \tag{2}$$

Similar to sparse recovery, the goal will be to recover z from y.

B. Sub-gradient descent

Our goal is to learn the entries of z given a minimal number of samples. We achieve this by performing *projected sub-gradient descent* (PsGD). Define the loss function

$$\psi(z) = \|y - \mathcal{A}(z)\|_1.$$

and pose the estimate of z to the solution to the non-convex constrained optimization problem:

$$\widehat{z} = \underset{\|z\|_0 \leq s}{\arg\min}\, \psi(z).$$

As opposed to ordinary sparse recovery approaches, the objective function is non-differentiable and therefore conventional iterative hard-thresholding schemes do not directly apply. Instead, starting from the zero vector $z_0 = 0$, we update the estimated coefficients via projected sub-gradient descent:

$$z_{t+1} = P_{\kappa s}(z_t - \eta_t \partial\psi(z_t)), \tag{3}$$

where $z_0 := 0$, $\partial\psi(x)$ is a sub-gradient of ψ at x, and κ and η_t are constants to be specified later in the proof. By properties of the absolute value function, one can see that a suitable sub-gradient is the *sign* function:

$$\partial\psi(z) = \mathcal{A}^* \text{sgn}(y - \mathcal{A}(z)).$$

We will prove below that with sufficiently many samples, the PsGD algorithm converges *linearly*, and at termination, provides an accurate estimate of the true polynomial coefficients. To our knowledge, this proof is novel. This proof complements the theoretical analysis of [6] with a simple, practical algorithm with provably fast convergence guarantees; the method suggested in that paper used convex relaxation and didn't specify convergence rates. In particular, we obtain the following:

Theorem 1: Suppose that the linear map $\mathcal{A}$ is constructed using $n = O(s \log p)$ cut queries of the form 1. Then, the projected sub-gradient descent algorithm produces a sequence of estimates $\widehat{z}_t$ for $t = 1, 2, \ldots$ such that:

$$\|\widehat{z}_{t+1} - z\|_2 \leq \gamma \|\widehat{z}_t - z\|_2 + C\|e\|_1,$$

where $0 < \gamma < 1$ is a constant that depends on α, β, κ, s, and $C > 0$ is a constant. Moreover, the algorithm provides an error of ε within $O(\log(1/\varepsilon))$ iterations.

The running time of PsGD is given by $\tilde{O}(sp^2)$, where the tilde hides polylogarithmic factors. This theorem also affirmatively addresses an open question (Remark 12) of [21], which suggest the possibility of non-convex algorithms to solve recovery problems of the form (1).

C. Proof of linear convergence

In this section, we prove Theorem 1 in the noiseless case where $e = 0$. The proof in the presence of noise follows in an identical manner (but with somewhat more tedious algebra), so we omit it here.

In [6], it is shown for several distributions from which a_i are sampled, the (debiased form of) the linear operator $\mathcal{A}$ satisfies a form of *mixed-norm* restricted isometry property. Specifically, if we consider consecutive samples a_{2i-1}, a_{2i}, then we "debias" the linear mapping $\mathcal{A}$ by defining $\mathcal{B}$ such that:

$$\mathcal{B} : d \to \sum_{J \subset [n]} d_J \left(\chi_J(a_1) - \chi_J(a_2)\right).$$

This can be simulated by taking consecutive differences of observed samples. Moreover, this linear mapping satisfies the so-called *RIP*-(ℓ_2, ℓ_1). Formally, the RIP-(ℓ_2, ℓ_1) says that there exist constants α, β such that $0 < \alpha < \beta$ that depend only on the problem size parameters p, n, s, such that for *any* vector z with $\|z\|_0 \leq 2s$, the following holds:

$$\alpha\|z\|_2 \leq \|\mathcal{B}(z)\|_1 \leq \beta\|z\|_2. \tag{4}$$

Henceforth, we use the notation $\mathcal{A}$, while keeping in mind that we are really taking about is really a debiased version of the definition given in (2).

We now show that if the linear map $\mathcal{A}$ satisfies (4), then PsGD converges linearly. Our approach follows that of [22], who demonstrate convergence of a similar algorithm using sub-exponential Gaussian observations; our proof is somewhat different and relatively shorter. We rely on the following auxiliary geometric lemma by [23], which states that an orthogonal projection onto the set of sparse vectors behaves like a *near-contraction*.

Lemma 2: For any $z \in \mathbb{R}^n$ and s-sparse $w \in \mathbb{R}^n$ and for any integer $\kappa > 1$, the following holds:

$$\begin{aligned}\|P_{\kappa s}(z) - w\|_2^2 &\leq \left(1 + 2\sqrt{\frac{1}{\kappa - 1}}\right)\|z - w\|_2^2 \\ &:= \nu\|z - w\|_2^2.\end{aligned}$$

The "near-contraction" parameter ν can be made arbitrarily close to 1, provided we increase the parameter κ accordingly. We now are ready to prove our main theorem. Proof. Define $\Omega = \text{supp}(z) \bigcup \text{supp}(z_t) \bigcup \text{supp}(z_{t+1})$. Let $g_t = \partial\psi(z_t) = \mathcal{A}^*\text{sgn}(y - \mathcal{A}(z_t))$. Define $b - z_t - \eta_t g_t$. Since z_{t+1} is the best sparse approximation to b, it also happens to be the best sparse approximation to b^Ω. We apply Lemma 2 to $z = b^\Omega$ and $w = z_{t+1}$ to get:

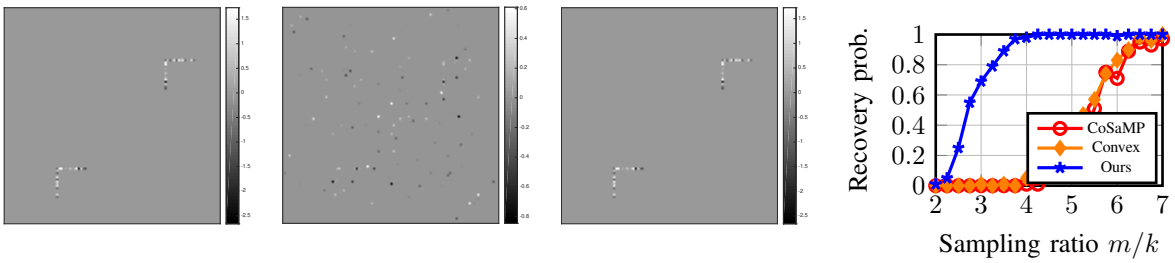
$$\|z_{t+1} - z\|_2^2 \leq \nu\left\|z - b^\Omega\right\|_2^2.$$

We now simplify the right hand side as follows:

$$\begin{aligned}&\left\|z - b^\Omega\right\|_2^2 \\ &= \|z - z_t - \eta_t \mathcal{A}_\Omega^* \text{sgn}(\mathcal{A}(z - z_t))\|_2^2 \\ &= \|z - z_t\|_2^2 - 2\eta_t\langle z - z_t, \mathcal{A}_\Omega^*\text{sgn}(\mathcal{A}(z - z_t))\rangle + \eta_t^2\left\|g_t^\Omega\right\|_2^2 \\ &= \|z - z_t\|_2^2 - 2\eta_t\langle \mathcal{A}_\Omega(z - z_t), \text{sgn}(\mathcal{A}(z - z_t))\rangle + \eta_t^2\left\|g_t^\Omega\right\|_2^2 \\ &= \|z - z_t\|_2^2 - 2\eta_t\|\mathcal{A}(z - z_t)\|_1 + \eta_t^2\left\|g_t^\Omega\right\|_2^2. \end{aligned} \tag{5}$$

We now upper bound $\left\|g_t^\Omega\right\|$ as follows. Observe that

$$\begin{aligned}\left\|g_t^\Omega\right\|_2^2 &= \langle \mathcal{A}_\Omega^*\text{sgn}(y - \mathcal{A}(z_t)), \mathcal{A}_\Omega^*\text{sgn}(y - \mathcal{A}(z_t))\rangle \\ &= \langle \text{sgn}(y - \mathcal{A}(z_t)), \mathcal{A}_\Omega\mathcal{A}_\Omega^*\text{sgn}(y - \mathcal{A}(z_t))\rangle \\ &\leq \|\mathcal{A}_\Omega\mathcal{A}_\Omega^*\text{sgn}(y - \mathcal{A}(z_t))\|_1 \\ &\leq \beta\|\mathcal{A}_\Omega^*\text{sgn}(y - \mathcal{A}(z_t))\|_2 = \beta\left\|g_t^\Omega\right\|_2.\end{aligned}$$



Original graph Convex opt. [2] Our method Numerical comparison

Fig. 1: *Simulated graph recovery results using various algorithms. Images represent adjacency matrices with weights depicted in grayscale. Our algorithm improves sample-complexity by at least 2.5× over existing approaches.*

where the last inequality follows from the upper bound in (4). Therefore, we get:

$$\left\|g_t^{\Omega}\right\| \leq \beta.$$

Plugging this into (5) and setting $\eta_t = \|y - Az_t\|/\beta^2$ we get:

$$\begin{aligned}
&\left\|z - b^{\Omega}\right\|_2^2 \\
&\leq \|z - z_t\|_2^2 - 2\frac{\|\mathcal{A}(z - z_t)\|_1^2}{\beta^2} + \frac{\|\mathcal{A}(z - z_t)\|_1^2}{\beta^2} \\
&= \|z - z_t\|_2^2 - \frac{\|\mathcal{A}(z - z_t)\|_1^2}{\beta^2} \\
&\leq \|z - z_t\|_2^2 - \frac{\alpha^2}{\beta^2}\|z - z_t\|_2^2 = \left(1 - \frac{\alpha^2}{\beta^2}\right)\|z - z_t\|_2^2.
\end{aligned}$$

Therefore, we get:

$$\|z_{t+1} - z\|_2^2 \leq \nu\left(1 - \frac{\alpha^2}{\beta^2}\right)\|z - z_t\|_2^2,$$

which gives us the noiseless version of Theorem 1 with convergence parameter $\gamma = \sqrt{\nu}\sqrt{1 - \alpha^2/\beta^2}$. Of course, convergence happens only when $\gamma < 1$, so we need to choose κ accordingly. If we define $\Delta := \beta/\alpha$ then $\kappa = 1$ suffices as long as $\Delta < \sqrt{4/3}$. Otherwise, we set:

$$\kappa > \frac{1}{\left(\frac{\Delta}{\sqrt{\Delta^2 - 1}} - 1\right)^2}.$$

This completes the proof.

III. Numerical Validation

Our primary contribution in this paper is conceptual. Nonetheless, we provide preliminary numerical evidence that our algorithm indeed succeeds as advertised, while acknowledging that more extensive numerical evaluations of our method are necessary.

We demonstrate the benefits of our approach in comparison with existing techniques. Figure 1 demonstrates learning the structure of a graph exhibiting star-like evolutions; this test example has been proposed and experimented upon in [8]. Figure 1 also displays recovery results using a greedy pursuit algorithm (suggested in [3]) and convex optimization (suggested in [2]). The input to all methods are $n = 1140$ random cut sketches of the test adjacency matrix W with $s = 100$ edges and $p = 128$ nodes. Figure 1 also displays phase transitions of the probability of successful recovery as a function of the number of cut sketches n. We observe that our algorithm is able to accurately recover the edge structure of the evolution, while previous methods, such as ℓ_1-minimization [24] and CoSaMP [25] fail. The fact that our algorithm can effectively leverage the dependencies between edges in the target graph evolution is key to its success.

The above star-graph example is relevant for networks that exhibit well-defined *hubs* where a central node is connected to lots of other nodes. Extension to other families of graph structures (such as cliques, blocks, cycles, and trees) is straightforward; as long as there exists a routine to project onto the space of possible supports of graph evolutions, the above proof can be modified to show that PsGD exhibits linear convergence.

IV. Related work and discussion

Due to space constraints, our discussion of prior work will necessarily be brief. We refer to the seminal work of [2] for an in-depth discussion of prior work in this area.

Approaches to reconstructing graphs from linear measurements have emerged (more or less in parallel) in the machine learning literature [2], [3] as well as in the theoretical computer science literature [1], [26]. Reconstructing structured temporal differences of graphs has been subsequently studied [8].

Recent work by [6], [7] have made explicit the connections between graph sketching and sparse recovery by representing cut queries via *rank-one projections* of the adjacency matrix. We borrow this setup. However, we note that the techniques of [6], [7] involve convex relaxations and do not seem to be easily amenable to structured graphs. Our algorithm and analysis closes this gap.

Recovery of structured-sparse vectors from random linear measurements has been studied by [10], [11], [12], [13], [14]. However, all these earlier works eventually require an RIP-like assumption of the observation operator, and is not applicable to cut queries.

Finally, we note that recovering adjacency matrices from cut queries can be interpreted as a special case of learning sparse additive models from quadratic measurements [27], [21], [28]. However, these approaches seems to require carefully chosen *adaptive* queries. in contrast, our framework succeeds with random non-adaptive cut queries and is more generically applicable.

REFERENCES

[1] K. Ahn, S. Guha, and A. McGregor, "Analyzing graph structure via linear measurements," in *Proc. ACM Symp. Discrete Alg. (SODA)*, 2012, pp. 459–467.

[2] P. Stobbe and A. Krause, "Learning Fourier sparse set functions.," in *Proc. Int. Conf. Art. Intell. Stat. (AISTATS)*, 2012, pp. 1125–1133.

[3] S. Negahban and D. Shah, "Learning sparse boolean polynomials," in *Proc. Allerton Conf. on Comm., Contr., and Comp.*, 2012, pp. 2032–2036.

[4] M. Kocaoglu, K. Shanmugam, A. Dimakis, and A. Klivans, "Sparse polynomial learning and graph sketching," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2014, pp. 3122–3130.

[5] Xiao Li and Kannan Ramchandran, "An active learning framework using sparse-graph codes for sparse polynomials and graph sketching," in *Advances in Neural Information Processing Systems*, 2015, pp. 2170–2178.

[6] Y. Chen, Y. Chi, and A. Goldsmith, "Exact and stable covariance estimation from quadratic sampling via convex programming," *IEEE Trans. Inform. Theory*, vol. 61, no. 7, pp. 4034–4059, 2015.

[7] Gautam Dasarathy, Parikshit Shah, Badri Narayan Bhaskar, and Robert D Nowak, "Sketching sparse matrices, covariances, and graphs via tensor products," *IEEE Transactions on Information Theory*, vol. 61, no. 3, pp. 1373–1388, 2015.

[8] F. Fazayeli and A. Banerjee, "Generalized direct change estimation in ising model structure," in *Proc. Int. Conf. Machine Learning (ICML)*, 2016, pp. 2281–2290.

[9] K. Tan, P. London, K. Mohan, S. Lee, M. Fazel, and D. Witten, "Learning graphical models with hubs.," *J. Machine Learning Research*, vol. 15, no. 1, pp. 3297–3331, 2014.

[10] V. Cevher, M. Duarte, C. Hegde, and R. Baraniuk, "Sparse signal recovery using Markov Random Fields," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, Dec. 2008.

[11] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inform. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[12] C. Hegde, P. Indyk, and L. Schmidt, "Approximation-tolerant model-based compressive sensing," in *Proc. ACM Symp. Discrete Alg. (SODA)*, Jan. 2014.

[13] C. Hegde, P. Indyk, and L. Schmidt, "Approximation algorithms for model-based compressive sensing," *IEEE Trans. Inform. Theory*, vol. 61, no. 9, pp. 5129–5147, Sept. 2015.

[25] Deanna Needell and Joel A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[14] C. Hegde, P. Indyk, and L. Schmidt, "A nearly linear-time framework for graph-structured sparsity," in *Proc. Int. Conf. Machine Learning (ICML)*, July 2015.

[15] R. Blei and S. Janson, "Rademacher chaos: tail estimates versus limit theorems," *Arkiv för Matematik*, vol. 42, no. 1, pp. 13–29, 2004.

[16] R. Latała, "Estimates of moments and tails of gaussian chaoses," *The Annals of Probability*, vol. 34, no. 6, pp. 2315–2331, 2006.

[17] B. Nazer and R. Nowak, "Sparse interactions: Identifying high-dimensional multilinear systems via compressed sensing," in *Proc. Allerton Conf. on Comm., Contr., and Comp.*, 2010, pp. 1589–1596.

[18] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir, "On the computational efficiency of training neural networks," in *Advances in neural information processing systems*, 2014, pp. 855–863.

[19] M. Soltani and C. Hegde, "Towards provable learning of polynomial neural networks using low-rank matrix estimation," in *Proc. Intl. Conf. Artificial Intelligence and Statistics (AISTATS)*, Apr. 2018.

[20] M. Soltani and C. Hegde, "Provable algorithms for learning two-layer polynomial neural networks," *IEEE Trans. Sig. Proc.*, vol. 67, no. 13, pp. 3361–3371, July 2019.

[21] Hemant Tyagi, Anastasios Kyrillidis, Bernd Gärtner, and Andreas Krause, "Algorithms for learning sparse additive models with interactions in high dimensions," *Information and Inference: A Journal of the IMA*, vol. 7, no. 2, pp. 183–249, 2017.

[22] Simon Foucart and Guillaume Lecué, "An iht algorithm for sparse recovery from subexponential measurements," *IEEE Signal Processing Letters*, vol. 24, no. 9, pp. 1280–1283, 2017.

[23] X. Li, T. Zhao, R. Arora, H. Liu, and J. Haupt, "Stochastic variance reduced optimization for nonconvex sparse learning," in *Proc. Int. Conf. Machine Learning (ICML)*, 2016, pp. 917–925.

[24] Simon Foucart and Holger Rauhut, *A Mathematical Introduction to Compressive Sensing*, Springer, 2013.

[26] Dmitry Kogan and Robert Krauthgamer, "Sketching cuts in graphs and hypergraphs," in *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*. ACM, 2015, pp. 367–376.

[27] Hemant Tyagi, Anastasios Kyrillidis, Bernd Gärtner, and Andreas Krause, "Learning sparse additive models with interactions in high dimensions," in *Artificial Intelligence and Statistics*, 2016, pp. 111–120.

[28] Hemant Tyagi and Jan Vybiral, "Learning non-smooth sparse additive models from point queries in high dimensions," *arXiv preprint arXiv:1801.08499*, 2018.