Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

A Nearly-Linear Time Framework for Graph-Structured Sparsity

Chinmay Hegde
Iowa State University
chinmay@iastate.edu

Piotr Indyk
MIT
indyk@mit.edu

Ludwig Schmidt*
MIT
ludwigs@mit.edu

Abstract

We introduce a framework for sparsity structures defined via graphs. Our approach is flexible and generalizes several previously studied sparsity models. Moreover, we provide efficient projection algorithms for our sparsity model that run in nearly-linear time. In the context of sparse recovery, our framework achieves an information-theoretically optimal sample complexity for a wide range of parameters. We complement our theoretical analysis with experiments showing that our algorithms also improve on prior work in practice.

1 Introduction

Over the past decade, sparsity has emerged as an important tool in several fields including signal processing, statistics, and machine learning. In compressive sensing, sparsity reduces the sample complexity of measuring a signal, and statistics utilizes sparsity for high-dimensional inference tasks. In many settings, sparsity is a useful ingredient because it enables us to model structure in high-dimensional data while still remaining a mathematically tractable concept. For instance, natural images are often sparse when represented in a wavelet basis, and objects in a classification task usually belong to only a small number of classes.

Due to the success of sparsity, a natural question is how we can refine the notion of sparsity in order to capture more complex *structures*. There are many examples where such an approach is applicable: (i) large wavelet coefficients of natural images tend to form connected *trees*, (ii) active genes can be arranged in functional *groups*, and (iii) approximate point sources in astronomical data often form *clusters*. In such cases, exploiting this additional structure can lead to improved compression ratio for images, better multi-label classification, or smaller sample complexity in compressive sensing and statistics. Hence an important question is the following: how can we model such sparsity structures, and how can we make effective use of this additional information in a computationally efficient manner?

There has been a wide range of work addressing these questions, e.g., [Yuan and Lin, 2006; Jacob *et al.*, 2009; He and Carin, 2009; Kim and Xing, 2010; Bi and Kwok, 2011; Huang *et al.*, 2011; Duarte and Eldar, 2011; Bach *et al.*, 2012b; Rao *et al.*, 2012; Negahban *et al.*, 2012; Simon *et al.*, 2013; El Halabi and Cevher, 2015]. Usually, the proposed solutions offer a trade-off between the following conflicting goals:

Generality What range of sparsity structures does the approach apply to?

Statistical efficiency What statistical performance improvements does the use of structure enable?

Computational efficiency How fast are the resulting algorithms?

In this paper, we introduce a framework for sparsity models defined through *graphs*, and we show that it achieves a compelling trade-off between the goals outlined above. At a high level, our approach applies to data with an underlying graph structure in which the large coefficients form a small number of connected components (optionally with additional constraints on the edges). Our approach offers three main features: (i) *Generality*: the framework encompasses several previously studied sparsity models, e.g., tree sparsity and cluster sparsity. (ii) *Statistical efficiency*: our sparsity model leads to reduced sample complexity in sparse recovery and achieves the information-theoretic optimum for a wide range of parameters. (iii) *Computational efficiency*: we give a nearly-linear time algorithm for our sparsity model, significantly improving on prior work both in theory and in practice. Due to the growing size of data sets encountered in science and engineering, algorithms with (nearly-)linear running time are becoming increasingly important.

We achieve these goals by connecting our sparsity model to the *prize collecting Steiner tree* (PCST) problem, which has been studied in combinatorial optimization and approximation algorithms. To establish this connection, we introduce a generalized version of the PCST problem and give a nearly-linear time algorithm for our variant. We believe that our sparsity model and the underlying algorithms are useful beyond sparse recovery, and we have already obtained results in this direction. To keep the presentation in this paper coherent, we focus on our results for sparse recovery and briefly mention further applications in Section 4.

We give an overview of our theoretical results in Section 2 and refer the reader to the full version of this paper for proofs

*Authors ordered alphabetically.

and further details [Hegde *et al.*, 2015b]. In Section 3, we complement our theoretical results with an empirical evaluation on both synthetic and real data (a background-subtracted image, a cerebral angiogram, and an image of text).

Basic notation Let $[d]$ be the set $\{1, 2, \ldots, d\}$. We say that a vector $\beta \in \mathbb{R}^d$ is s-sparse if at most s of its coefficients are nonzero. The support of β contains the indices corresponding to nonzero entries in β, i.e., $\text{supp}(\beta) = \{i \in [d] \,|\, \beta_i \neq 0\}$. Given a subset $S \subseteq [d]$, we write β_S for the restriction of β to indices in S: we have $(\beta_S)_i = \beta_i$ for $i \in S$ and $(\beta_S)_i = 0$ otherwise. The ℓ_2-norm of β is $\|\beta\| = \sqrt{\sum_{i \in [d]} \beta_i^2}$.

Sparsity models In some cases, we have more information about a vector than only "standard" s-sparsity. A natural way of encoding such additional structure is via *sparsity models* [Baraniuk *et al.*, 2010]: let $\mathbb{M}$ be a family of supports, i.e., $\mathbb{M} = \{S_1, S_2, \ldots, S_L\}$ where $S_i \subseteq [d]$. Then the corresponding sparsity model $\mathcal{M}$ is the set of vectors supported on one of the S_i:

$$\mathcal{M} \;=\; \{\beta \in \mathbb{R}^d \,|\, \text{supp}(\beta) \subseteq S \text{ for some } S \in \mathbb{M}\} \;. \qquad (1)$$

2 Our contributions

We state our main contributions in the context of sparse recovery (see Section 4 for further applications). Here, the goal is to estimate an unknown s-sparse vector $\beta \in \mathbb{R}^d$ from observations of the form

$$y \;=\; X\beta + e \;, \qquad (2)$$

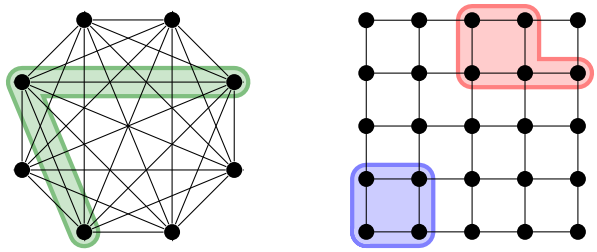
where $X \in \mathbb{R}^{n \times d}$ is the design matrix, $y \in \mathbb{R}^n$ are the observations, and $e \in \mathbb{R}^n$ is an observation noise vector. By imposing various assumptions on X and e, sparse recovery encompasses problems such as sparse linear regression and compressive sensing.

2.1 Weighted graph model (WGM)

The core of our framework for structured sparsity is a novel, **general** sparsity model which we call the *weighted graph model*. In the WGM, we use an underlying graph $G = (V, E)$ defined on the coefficients of the unknown vector β, i.e., $V = [d]$. Moreover, the graph is weighted and we denote the edge weights with $w : E \to \mathbb{N}$. We identify supports $S \subseteq [d]$ with subgraphs in G that are *forests* (unions of individual trees). Intuitively, the WGM captures sparsity structures with a small number of connected components in G. In order to control the sparsity patterns, the WGM offers three parameters:

- s, the total sparsity of S.
- g, the maximum number of connected components formed by the forest F corresponding to S.
- B, the bound on the total weight $w(F)$ of edges in the forest F corresponding to S.

More formally, let $\gamma(H)$ be the number of connected components in a graph H. Then we can define the WGM:



(a) s-sparsity (b) Cluster sparsity

Figure 1: Two examples of the weighted graph model. (a) In a complete graph, any s-sparse support can be mapped to a single tree ($g = 1$). (b) Using a grid graph, we can model a small number of clusters in an image by setting g accordingly. For simplicity, we use unit edge weights and set $B = s - g$ in both examples.

Definition 1. *The (G, s, g, B)-WGM is the set of supports*

$$\mathbb{M} = \{S \subseteq [d] \mid |S| = s \text{ and there is a } F \subseteq G \text{ with } V_F = S,\ \gamma(F) = g, \text{ and } w(F) \leq B\} \;. \qquad (3)$$

Fig. 1 shows how two sparsity structures can be encoded with the WGM. Since our sparsity model applies to *arbitrary* graphs G, it can describe a wide range of structures. In particular, the model generalizes several previously studied sparsity models, including 1D-clusters, (wavelet) tree hierarchies, the Earth Mover Distance (EMD) model, and the unweighted graph model (see the full paper [Hegde *et al.*, 2015b] for a detailed comparison).

2.2 Recovery of vectors in the WGM

We analyze the **statistical efficiency** of our framework in the context of sparse recovery. In particular, we prove that the sample complexity of recovering vectors in the WGM is provably smaller than the sample complexity for "standard" s-sparse vectors. To formally state this result, we first introduce a key property of graphs.

Definition 2. *Let $G = (V, E)$ be a weighted graph with edge weights $w : E \to \mathbb{N}$. Then the weight-degree $\rho(v)$ of a node v is the largest number of adjacent nodes connected by edges with the same weight, i.e.,*

$$\rho(v) \;=\; \max_{b \in \mathbb{N}} |\{(v', v) \in E \,|\, w(v', v) = b\}| \;. \qquad (4)$$

We define the weight-degree of G to be the maximum weight-degree of any $v \in V$.

Note that for graphs with uniform edge weights, the weight-degree of G is the same as the maximum node degree. Intuitively, the (weight) degree of a graph is an important property for quantifying the sample complexity of the WGM because the degree determines how restrictive the bound on the number of components g is. In the extreme case of a complete graph, any support can be formed with only a single connected component (see Figure 1). Using Definitions 1 and 2, we now state our sparse recovery result (see the full version of this paper for a more general statement of this result [Hegde *et al.*, 2015b]).

Theorem 3. *Let $\beta \in \mathbb{R}^d$ be in the (G, s, g, B)-WGM. Then*

$$n = O\left(s\left(\log \rho(G) + \log \frac{B}{s}\right) + g \log \frac{d}{g}\right) \tag{5}$$

i.i.d. Gaussian observations suffice to estimate β. More precisely, let $e \in \mathbb{R}^n$ be an arbitrary noise vector and let $y \in \mathbb{R}^n$ be defined as in Eq. 2 where X is an i.i.d. Gaussian matrix. Then we can efficiently find an estimate $\widehat{\beta}$ such that

$$\left\|\beta - \widehat{\beta}\right\| \ \leq \ C\|e\| \ , \tag{6}$$

where C is a constant independent of all variables above.

Note that in the noiseless case ($e = 0$), we are guaranteed to recover β exactly. Moreover, our estimate $\widehat{\beta}$ is in a slightly enlarged WGM for any amount of noise. Our bound (5) can be instantiated to recover previous sample complexity results, e.g., the $n = O(s \log \frac{d}{s})$ bound for "standard" sparse recovery, which is tight [Do Ba *et al.*, 2010]. For the image grid graph example in Figure 1, Equation (5) becomes $n = O(s + g \log \frac{d}{g})$, which matches the information-theoretic optimum $n = O(s)$ as long as the number of clusters is not too large, i.e., $g = O(s/\log d)$.

2.3 Efficient projection into the WGM

The algorithmic core of our sparsity framework is a **computationally efficient** procedure for projecting arbitrary vectors into the WGM. More precisely, the model-projection problem is the following: given a vector $b \in \mathbb{R}^d$ and a WGM $\mathcal{M}$, find the best approximation to b in $\mathcal{M}$, i.e.,

$$P_{\mathcal{M}}(b) \ = \ \arg\min_{b' \in \mathcal{M}} \|b - b'\| \ . \tag{7}$$

If such a model-projection algorithm is available, one can instantiate the framework of [Baraniuk *et al.*, 2010] in order to get an algorithm for sparse recovery with the respective sparsity model. However, solving Problem (7) exactly is NP-hard for the WGM due to a reduction from the classical Steiner tree problem. To circumvent this hardness result, we use the *approximation-tolerant* framework of [Hegde *et al.*, 2015a]. Instead of solving (7) exactly, the framework requires *two* algorithms with the following complementary approximation guarantees.

Tail approximation: Find an $S \in \mathbb{M}$ such that

$$\|b - b_S\| \leq c_T \cdot \min_{S' \in \mathbb{M}} \|b - b_{S'}\| \ . \tag{8}$$

Head approximation: Find an $S \in \mathbb{M}$ such that

$$\|b_S\| \geq c_H \cdot \max_{S' \in \mathbb{M}} \|b_{S'}\| \ . \tag{9}$$

Here, $c_T > 1$ and $c_H < 1$ are arbitrary, fixed constants. Note that a head approximation guarantee does not imply a tail guarantee (and vice versa). In fact, stable recovery is not possible with only one type of approximate projection guarantee [Hegde *et al.*, 2015a]. We provide two algorithms for solving (8) and (9) (one per guarantee) which both run in *nearly-linear time*.

Our model-projection algorithms are based on a connection to the prize-collecting Steiner tree problem (PCST), which is a generalization of the classical Steiner tree problem. Instead of finding the cheapest way to connect *all* terminal nodes in a given weighted graph, we can instead omit some terminals from the solution and pay a specific price for each omitted node. The goal is to find a subtree with the optimal trade-off between the cost paid for edges used to connect a subset of the nodes and the price of the remaining, unconnected nodes (see [Goemans and Williamson, 1995] for a formal definition).

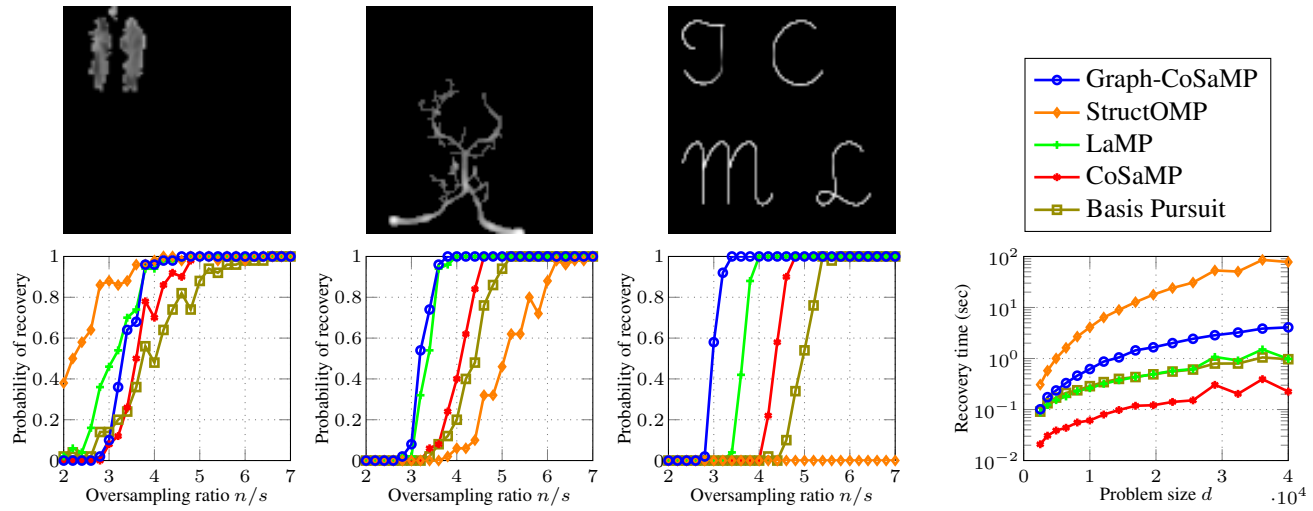
We make the following three main algorithmic contributions. Due to the wide applicability of the PCST problem, we believe that these algorithms can be of independent interest (see Section 4).

- We introduce a variant of the PCST problem in which the goal is to find a set of g trees instead of a single tree. We call this variant the prize-collecting Steiner forest (PCSF) problem and adapt the algorithm of [Goemans and Williamson, 1995] for this variant.
- We reduce the projection problems (8) and (9) to a small set of adaptively constructed PCSF instances.
- We give a nearly-linear time algorithm for the PCSF problem and hence also the model projection problem.

2.4 Comparison to related work

In addition to "point-solutions" for individual sparsity models, there has been a wide range of work on general frameworks for utilizing structure in sparse recovery. The approach most similar to ours is [Baraniuk *et al.*, 2010], which gives a framework underlying several recovery algorithms for individual sparsity models. However, the framework has one important drawback: it does not come with a full recovery algorithm. Instead, the authors only give a recovery scheme that assumes the existence of a model-projection algorithm satisfying (7). Such an algorithm must be constructed from scratch for each model, and the techniques that have been used for various models so far are quite different. Our contribution can be seen as complementing the framework of [Baraniuk *et al.*, 2010] with a nearly-linear time projection algorithm that is applicable to a wide range of sparsity structures. This answers a question raised by the authors of [Huang *et al.*, 2011], who also give a framework for structured sparsity with a universal and complete recovery algorithm. Their framework is applicable to a wide range of sparsity models, but the corresponding algorithm is significantly slower than ours, both in theory as well as in practice, as demonstrated by our experiments in Section 3. Moreover, our recovery algorithm shows more robust performance across different shapes of graph clusters.

Both of the approaches mentioned above use iterative *greedy* algorithms for sparse recovery. There is also a large body of work on combining M-estimators with *convex* regularizers that induce structured sparsity, e.g., see the surveys [Bach *et al.*, 2012a] and [Wainwright, 2014]. The work closest to ours is [Jacob *et al.*, 2009], which uses an overlapping group Lasso to enforce graph-structured sparsity (graph Lasso). In contrast to their approach, our algorithm gives more fine-grained control over the number of clusters in the graph. Moreover, our algorithm has better computational complexity, and to the best of our knowledge there are no



(a) Background-subtracted image (b) Angiogram (c) Text (d) Running times

Figure 2: Sparse recovery experiments. The images in the top row are the original images β.

formal results relating the graph structure to the sample complexity of the graph Lasso. Empirically, our algorithm recovers an unknown vector with graph structure faster and from fewer observations than the graph Lasso (see the supplementary material of the full paper [Hegde *et al.*, 2015b]).

3 Experiments

We focus on the performance of our proposed algorithm (Graph-CoSaMP) for the task of recovering 2D data with clustered sparsity. Multiple methods have been proposed for this problem, and our theoretical analysis shows that our algorithm should improve upon the state of the art. We compare our results to StructOMP [Huang *et al.*, 2011] and the heuristic Lattice Matching Pursuit (LaMP) [Cevher *et al.*, 2009]. The implementations were supplied by the authors and we used the default parameter settings. Moreover, we ran two common recovery algorithms for "standard" s-sparsity: Basis Pursuit [Candès *et al.*, 2006] and CoSaMP [Needell and Tropp, 2009].

We follow a standard evaluation procedure for sparse recovery / compressive sensing: we record n observations $y = X\beta$ of the (vectorized) image $\beta \in \mathbb{R}^d$ using a subsampled Fourier matrix X. We assume that all algorithms possess prior knowledge of the sparsity s and the number of connected-components g in the true support of the image β. We declare a trial successful if the squared ℓ_2-norm of the recovery error is at most 5% of the squared ℓ_2-norm of the original vector β. The probability of successful recovery is then estimated by averaging over 50 trials. We perform several experiments with varying oversampling ratios n/s and three different images. See the supplementary material of the full paper [Hegde *et al.*, 2015b] for a description of the dataset, experiments with noise, and a comparison with the graph Lasso.

Figure 2 demonstrates that Graph-CoSaMP yields consistently competitive phase transitions and exhibits the best sample complexity for images with "long" connected clusters, such as the angiogram image (b) and the text image (c). While StructOMP performs well on "blob"-like images such as the background-subtracted image (a), its performance is poor in our other test cases. For example, it can successfully recover the text image only for oversampling ratios $n/s > 15$. Note that the performance of Graph-CoSaMP is very consistent: in all three examples, the phase transition occurs between oversampling ratios 3 and 4. Other methods show significantly more variability.

We also investigate the computational efficiency of Graph-CoSaMP. We consider resized versions of the angiogram image and record $n = 6s$ observations for each image size d. Figure 2(d) displays the recovery times (averaged over 50 trials) as a function of d. We observe that the runtime of Graph-CoSaMP scales nearly linearly with d, comparable to the conventional sparse recovery methods. Moreover, Graph-CoSaMP is about $20\times$ faster than StructOMP.

4 Further applications

We have introduced a general framework for structured sparsity that encompasses several previously studied sparsity models, but still allows information-theoretically optimal sample complexity and nearly-linear time algorithms.

Our algorithms have found applications beyond sparse recovery. In [Schmidt *et al.*, 2015], we have used our PCST algorithm for a feature extraction task in seismic image processing. Other potential applications include the work of [Rozenshtein *et al.*, 2014], who use a PCST algorithm for event detection in social networks.

Acknowledgements

We thank Jayadev Acharya, Stefanie Jegelka, Youssef Mroueh, Devavrat Shah, and the anonymous reviewers for many helpful comments on earlier versions of this paper. This work was supported by grants from the MITEI-Shell program, the MADALGO center, and the Simons Investigator Award.

References

[Bach *et al.*, 2012a] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

[Bach *et al.*, 2012b] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 11 2012.

[Baraniuk *et al.*, 2010] Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010.

[Bi and Kwok, 2011] Wei Bi and James T. Kwok. Multi-label classification on tree- and DAG-structured hierarchies. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 17–24, 2011.

[Candès *et al.*, 2006] Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[Cevher *et al.*, 2009] Volkan Cevher, Marco F. Duarte, Chinmay Hegde, and Richard Baraniuk. Sparse signal recovery using markov random fields. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 257–264. 2009.

[Do Ba *et al.*, 2010] Khanh Do Ba, Piotr Indyk, Eric Price, and David P. Woodruff. Lower bounds for sparse recovery. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1190–1197, 2010.

[Duarte and Eldar, 2011] Marco F. Duarte and Yonina C. Eldar. Structured compressed sensing: From theory to applications. *IEEE Transactions on Signal Processing*, 59(9):4053–4085, 2011.

[El Halabi and Cevher, 2015] Marwa El Halabi and Volkan Cevher. A totally unimodular view of structured sparsity. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[Goemans and Williamson, 1995] Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

[He and Carin, 2009] Lihan He and Lawrence Carin. Exploiting structure in wavelet-based bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 57(9):3488–3497, 2009.

[Hegde *et al.*, 2015a] Chinmay Hegde, Piotr Indyk, and Ludwig Schmidt. Approximation algorithms for model-based compressive sensing. *IEEE Transactions on Information Theory*, 61(9):5129–5147, 2015.

[Hegde *et al.*, 2015b] Chinmay Hegde, Piotr Indyk, and Ludwig Schmidt. A nearly-linear time framework for graph-structured sparsity. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 928–937, 2015.

[Huang *et al.*, 2011] Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011.

[Jacob *et al.*, 2009] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group Lasso with overlap and graph Lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 433–440, 2009.

[Kim and Xing, 2010] Seyoung Kim and Eric P. Xing. Tree-guided group Lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 543–550, 2010.

[Needell and Tropp, 2009] Deanna Needell and Joel A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[Negahban *et al.*, 2012] Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 11 2012.

[Rao *et al.*, 2012] Nikhil S. Rao, Ben Recht, and Robert D. Nowak. Universal measurement bounds for structured sparse signal recovery. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22 of *JMLR Proceedings*, pages 942–950, 2012.

[Rozenshtein *et al.*, 2014] Polina Rozenshtein, Aris Anagnostopoulos, Aristides Gionis, and Nikolaj Tatti. Event detection in activity networks. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1176–1185, 2014.

[Schmidt *et al.*, 2015] Ludwig Schmidt, Chinmay Hegde, Piotr Indyk, Ligang Lu, Xingang Chi, and Detlef Hohl. Seismic feature extraction using Steiner tree methods. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[Simon *et al.*, 2013] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group Lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[Wainwright, 2014] Martin J. Wainwright. Structured regularizers for high-dimensional problems: Statistical and computational issues. *Annual Review of Statistics and Its Application*, 1(1):233–253, 2014.

[Yuan and Lin, 2006] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.