Published in final edited form as:
IEEE Trans Automat Contr. 2014 November ; 59(11): 2946–2961. doi:10.1109/TAC.2014.2351712.

Compressive Network Analysis

Xiaoye Jiang [Member, IEEE],
Stanford University, Stanford, CA 94305 USA

Yuan Yao [Member, IEEE],
Peking University, Beijing 100871, China

Han Liu [Member, IEEE], and
Princeton University, Princeton, NJ 08540 USA

Leonidas Guibas [Fellow, IEEE]
Stanford University, Stanford, CA 94305 USA

Yuan Yao: yuany@math.pku.edu.cn

Abstract

Modern data acquisition routinely produces massive amounts of network data. Though many methods and models have been proposed to analyze such data, the research of network data is largely disconnected with the classical theory of statistical learning and signal processing. In this paper, we present a new framework for modeling network data, which connects two seemingly different areas: *network data analysis* and *compressed sensing*. From a nonparametric perspective, we model an observed network using a large dictionary. In particular, we consider the network clique detection problem and show connections between our formulation with a new algebraic tool, namely *Randon basis pursuit in homogeneous spaces*. Such a connection allows us to identify rigorous recovery conditions for clique detection problems. Though this paper is mainly conceptual, we also develop practical approximation algorithms for solving empirical problems and demonstrate their usefulness on real-world datasets.

Index Terms

Clique detection; compressive sensing; network data analysis; Radon basis pursuit; restricted isometry property

I. Introduction

In the past decade, the research of network data has increased dramatically. Examples include scientific studies involving web data or hyper text documents connected via hyperlinks, social networks or user profiles connected via friend links, co-authorship and citation network connected by collaboration or citation relationships, gene or protein

networks connected by regulatory relationships, and much more. Such data appear frequently in modern application domains and have led to numerous high-impact applications. For instance, detecting anomaly in ad-hoc information network is vital for corporate and government security; exploring hidden community structures helps us to better conduct online advertising and marketing; inferring large-scale gene regulatory network is crucial for new drug design and disease control. Due to the increasing importance of network data, principled analytical and modeling tools are crucially needed.

Towards this goal, researchers from the network modeling community have proposed many models to explore and predict the network data. These models roughly fall into two categories: static and dynamic models. For the static model, there is only one single snapshot of the network being observed. In contrast, dynamic models can be applied to analyze datasets that contain many snapshots of the network indexed by different time points. Examples of the static network models include the Erdös-Rényi-Gilbert random graph model [18], [19], the p_1 [27], p_2 [17] and more general exponential random graph (or p^*) model [48], latent space model [26], block model [35], stochastic blockmodel [5], [47], and mixed membership stochastic block-model [2]. Examples of the dynamic network models include the preferential attachment model [4], the small-world model [49], duplication-attachment model [29], [32], continuous time Markov model [43], and dynamic latent space model [41]. A comprehensive review of these models is provided in [24].

Though many methods and models have been proposed, the research of network data analysis is largely disconnected with the classical theory of statistical learning and signal processing. The main reason is that, unlike the usual scientific data for which independent measurements can be repeatedly collected, network data are in general collected in one single realization and the nodes within the network may interact with each other through the network linkages. Such a disconnection prevents us from directly exploiting the state-of-the-art statistical learning methods and theory to analyze network data. To bridge this gap, we present a novel framework to model network data. Our framework assumes that the observed network has a sparse representation with respect to some dictionary. Once the dictionary is given, we formulate the network modeling problem into a *compressed sensing* problem. Compressed sensing, also known as compressive sensing and compressive sampling, is a technique for finding sparse solutions to underdetermined linear systems. In statistical machine learning, it is related to reconstructing a signal which has a sparse representation in a large dictionary. The field of compressed sensing has existed for decades, but recently it has exploded due to the important contributions of [8]–[10], [46]. It is believed that the compressed sensing will have far reaching implications. For example, it may suggests the possibility of new data acquisition protocols which can translate analog information into digital form with fewer sensors than what was considered necessary. Moreover, this technique can be used for phase retrieval where the phases of measurements from a linear system is omitted. In our scenario, by viewing the observed network adjacency matrix as the output of an underlying function evaluated on a discrete domain of network nodes, we can formulate the network modeling problem into a compressed sensing problem.

Specifically, we consider the network clique detection problem within this novel framework. Nevertheless, the network clique detection problem has been studied in the context of

computer science algorithms since many years ago. For example, [6] studied and compared several basic algorithms for finding the maximal complete subgraph (clique) of an undirected graph which is NP-hard in general; [21] developed two new algorithms by making use of special tree search algorithms for determining all maximal complete subgraphs of a finite undirected graph; [3] showed that spectral method can find cliques of sizes larger than $c\sqrt{n}$ (n is the number of vertices) in a polynomial time which is improved by [14] with $c = 1.261$ and by [15] with $c = (1 + \varepsilon)/\sqrt{e}$ in nearly linear time. Moreover, these problems have been studied in a more generalized context, e.g., [1] provided techniques that are useful for the detection of dense subgraphs (quasi-cliques) in massive sparse graphs. However, in this paper, we adopt a completely new framework to formulate the network clique detection problem inspired by modern statistical learning theory.

In our formulation, we assume we are given a network with its nodes representing players, items, or characters, and edge weights summarizing the observed pairwise interactions. The basic problem is to determine communities or cliques within the network by observing the frequencies of low order interactions, since in reality such low order interactions are often governed by a considerably smaller number of high order communities or cliques. Here, we use the term “low order” to describe data summarizing network individual or pairwise interaction statistics. Generally, one can interpret “low order” information as a set of quantities, each of which corresponds to a small group of nodes in the network. On the other hand, we will refer data on large cliques as “high order” information. Under such a formulation, we use a generative model in which the observed adjacency matrix that represent the network data is assumed to have a sparse representation in a large dictionary where each basis corresponds to a clique. With such a formulation, we connect our framework with a new algebraic tool, namely *Randon basis pursuit in homogeneous spaces*, which is essentially a *compressed sensing* problem of cliques in large networks. Our problem can thus be regarded as an extension of the work in [28] which studies sparse recovery of functions on permutation groups. The difference between our approach and [28] is that we reconstruct functions on k-sets (cliques, a set with cardinality k), often called the *homogeneous space* associated with a permutation group in the literature [16]. In general, our formulation can be viewed as a combinatorial version of the compressed sensing problem, where the basis matrix is constructed using the Radon bases. Before rigorously formulating the problem, we provide three motivating examples as a glimpse of typical situations which can be addressed within the framework in this paper.

Example 1 (Tracking Team Identities)

We consider the scenario of multiple targets moving in an environment monitored by sensors. We assume every moving target has an identity and they each belong to some teams or groups. However, we can only obtain partial interaction information due to the measurement structure. For example, watching a grey-scale video of a basketball game (when it may be hard to tell apart the two teams), sensors may observe ball passes or collaboratively offensive/defensive interactions between teammates. The observations are partial due to the fact that players mostly exhibit to sensors low order interactions in basketball games. It is difficult to observe a single event which involves all team members.

Our objective is to infer membership information (which team the players belong to) from such partially observed interactions.

Example 2 (Detecting Communities in Social Networks)

Detecting communities in social networks is of extraordinary importance. It can be used to understand the organization or collaboration structure of a social network. However, we do not have direct mechanisms to sense social communities. Instead, we have partial, low order interaction information. For example, we observe pairwise or triple-wise co-appearance among people who hang out for some leisure activities together. We hope to detect those social communities in the network from such partially observation data. It turns out that community structures are ubiquitous in social networks. However, there is no consistent definition of a "community". In the majority of research studies, community detections based on partitions of nodes in a network. Among these works, the most famous one is based on the modularity [38] of a partition of the nodes in a group. A shortcoming in partition-based methods is that they do not allow overlapping communities, which occur frequently in practice. Recently there has been growing interest in studying overlapping community structures [12], [31], [33]. The relevance of cliques to overlapping communities was probably first addressed in the clique percolation method [40]. In this work, communities were modeled as maximal connected components of cliques in a graph where two k-cliques are said to be connected if they share $k - 1$ nodes.

Example 3 (Inferring High Order Partial Rankings)

The problem of clique identification also arises in ranking problems. Consider a collection of items which are to be ranked by a set of users. Each user can propose the set of his or her j most favorite items (say top 3 items) but without specifying a relative preference within this set. We then wish to infer what are the top $k > j$ most favorite items (say top 5 items). This problem requires us to infer high order partial rankings from low order observations.

In these examples we are typically given a network with some nodes representing players, items, or characters, and edge weights summarizing the observed pairwise interactions. Triple-wise and other low order information can be further exploited if we consider complete sub-graphs or cliques in the networks. In this paper, we aim to formulate our problem as compressed sensing of cliques in networks. In the next section, we demonstrate that the Radon basis will be an appropriate representation for such a purpose which allows the sparse recovery by a simple linear programming reconstruction approach. With the choice of Radon basis, we develop new theories and algorithms which guarantee exact, sparse, and stable recovery in this paper. These theories have deep roots in Basis Pursuit [11] and its extensions with uniformly bounded noise. Though this paper is mainly conceptual: showing the connection between network modeling and compressed sensing, we also provide some rigorous theoretical analysis and practical algorithms on the clique recovery problem to illustrate the usefulness of our framework.

We summarize the main contributions of the paper as follows:

- We establish a compressive network analysis framework, which connects the network data analysis to the modern statistical learning and signal processing theories.
- Under the compressive network analysis framework, we study the network clique detection problem. We formulate such a problem into a compressed sensing problem, and provide theoretical guarantees under our formulation. We show that our formulated optimization problem can reliably recover the underlying network cliques in a variety of scenarios. In particular, we establish Theorems 3.5, 3.8, and 3.10 to describe exact recovery conditions for cliques of equal sizes, in the hierarchical setting, and in the incomplete Radon dictionary, respectively. Those theorems are followed by two extensions, Theorem 3.11 studies stable recovery under noise and Proposition 3.13 studies recovery for cliques of mixed sizes.
- We provide a polynomial time approximation algorithm to solve the network clique detection problem in our formulation, which can avoid the explicit construction of the basis matrix involved in the optimization problem. Due to the sparsity assumption, the solution of the optimization problem can be searched efficiently, by utilizing the cutting plane methods.
- We show the validity of our formulation and algorithm with numerous application examples.

An outline of this paper is given as follows. Section II presents the general framework of compressive network analysis. In particular, we present the Radon basis pursuit scheme under the general framework, which connects the Radon basis in algebraic combinatorics to the clique detection problem. Section III collects theoretical aspects of the Radon basis pursuit, which characterizes the exact and stable recovery under various scenarios. Section IV introduces a polynomial time approximation algorithm to solve the Radon basis pursuit that only exploits subsets of all the cliques in a greedy way. Section V demonstrates several successful application examples with both synthetic data and realistic network data. Section VI concludes the paper.

II. General Framework

In this section we present the general framework of compressive network analysis with a nonparametric view. We start with an introduction of notations: let * denote vector or matrix transpose, $u = (u_1, \ldots, u_d)^* \in \mathbb{R}^d$ be a vector, $I(\cdot)$ be the indicator function, and $\delta_{ij} = I(i = j)$. We denote

$$\|u\|_0 \equiv \sum_{j=1}^{d} I(u_j \neq 0), \|u\|_2 \equiv \sqrt{\sum_{j=1}^{d} u_j^2}, \|u\|_\infty \equiv \max_j |u_j|.$$

We also denote by $\langle \cdot, \cdot \rangle$ the Euclidean inner product and $\text{sign}(u) = (\text{sign}(u_1), \ldots, \text{sign}(u_d))^*$, where

$$\operatorname{sign}(u_j)=\begin{cases}+1 & \text{if } u_j>0;\\ -1 & \text{if } u_j<0;\\ 0 & \text{if } u_j=0.\end{cases}$$

We represent a network as a graph $G = (V, E)$, where $V = \{1,\ldots,n\}$ is the set of nodes and $E \subset V \times V$ is the set of directed edges. Let $B \in \mathbb{R}^{n\times n}$ be a matrix for the observed network whose element $B(i, j) \in \mathbb{R}$ represents a quantity associated with nodes i and j.

As a bivariate function $B : V \times V \to \mathbb{R}$, we consider the following representation:

$$B(i,j)=\sum_k c_k\phi_k(i,j)+z(i,j) \quad (2.1)$$

where each $\varphi_k : V \times V \to \mathbb{R}$ is a basis function. In our framework, we assume that (2.1) has a sparse representation with respect to a dictionary $A = [\varphi_1,\ldots, \varphi_N]$ and z accouts for the noise with such a representation. In other words, there exists a subset $S \subset \{1,\ldots, N\}$ with cardinality $|S| \ll N$, such that $c_k = 0$ for $k \notin S$. As N grows, we can view φ_k as evaluating a possibly infinite-dimensional function on $V \times V$, whence the model (2.1) is intrinsically nonparametric for arbitrary static networks.

Different choices of φ_k may lead to the following important cases.

- Tensor product: $\varphi_k(i, j) = u_k(i)v_k(j)$, where $u_k, v_k \in \mathbb{R}^V$, such a representation may lead to matrix factorization of B, such as singular vector decompositions (SVD), etc.;
- Subgraph indicator: φ_k is representing a subgraph of G, which can be learned from data, such as the motif in biological networks;
- Clique indicator: in particular, consider a complete subgraph or clique of G which has a set of nodes K. Let φ_K be the adjacency matrix of clique K, i.e. $\varphi_K(i, j) = 1$ if $i, j \in K$, $i \neq j$ and 0 otherwise. Such a basis function leads to an important case to be studied in this paper, the *clique detection problem*.

In the sequel, without loss of generality, we assume that B is symmetric: $B = B^*$ and $\operatorname{diag}(B) = 0$. With these assumptions, to model B, it is sufficient to model its upper-triangle because the lower-triangle is a identical copy and entries on the diagonal are trivial. For notational simplicity, we squeeze B into a vector $b \in \mathbb{R}^M$ where $M = n(n - 1)/2$ is the number of upper-triangle elements in B, i.e., we concatenate the $n, n - 1,\ldots, 1$ entries on the upper-triangle of B into a vector b. In this case each basis function becomes a vector $\varphi_k \in \mathbb{R}^M$ and $A = [\varphi_1, \ldots, \varphi_N]$ becomes a M-by-N matrix. We denote by A_{st} the element on the s-th row and t-th column of A. Here s indexes a pair of different nodes and t indexes a basis φ_t.

Given the dictionary A, we can recover the sparse representation in (2.1). We try to reconstruct $x = (x_1,\ldots, x_N)^*$ from the following problem:

$$(\mathrm{P}_0)\ \min \|x\|_0 \mathbf{s.t.} \|b - Ax\|_z \leq \delta. \quad (2.2)$$

In the above formulation, $\|\cdot\|_z$ is a vector norm constructed using the knowledge of z (here z is the noise term of (2.1) under the squeezed graph setting). However, the problem in (2.2) is non-convex. In the sparse learning literature, a convex relaxation of (2.2) can be written as

$$(\mathrm{P}_1)\ \min \|x\|_1 \mathbf{s.t.} \|b - Ax\|_z \leq \delta. \quad (2.3)$$

In later sections, we will detail the conditions which guarantee the equivalence of solutions to the above optimization problems.

Besides the clique detection problem in which only second-order interactions are needed, our method can also be extended to handle higher-order interactions. In this case, we need to extend the bivariate function $B : V^2 \to \mathbb{R}$ to higher order functions $V^j \to \mathbb{R}$, e.g., we collect top j element sets in partial order ranking as we discussed in Example 3. To pursue sparse representation similar as in (2.2) and (2.3), the dictionary A may be higher order tensors or coincidence matrices of cliques. In the rest of this paper, we only focus on the clique detection problem due to its notational simplicity, while the theory can be applied to more general settings.

A. Mathematical Formulation

Under the general framework in (2.1), we formulate the clique detection problem into a compressed sensing problem named *Radon basis pursuit*. For this, we construct a dictionary A so that each column of A corresponds to one clique. The intuition of such a construction is that we assume there are several hidden cliques within the network, which are perhaps of different sizes and may have overlaps. Every clique has certain weights, which contain meaningful information such as interactions between network nodes, partial ranking votes as we discussed in Examples 1–3. The observed adjacency matrix B (or equivalently, its vectorized version b) is a linear combination of many clique basis contaminated by a noise vector, which is the z-term in the general setting as in (2.1).

For simplicity, we first restrict ourselves to the case that all the cliques are of the same size k ($2 < k < n$). The case with mixed sizes will be discussed later. Let $C_1, C_2, \ldots, C_N$ be all the cliques of size k and each $C_j \subset V$. We have $N = \binom{n}{k}$. For each $t \in \{1, \ldots, N\}$, we construct the dictionary A as the following:

$$A_{st} = \begin{cases} 1 & \text{if the } s\text{-th pair of nodes both lie in } C_t; \\ 0 & \text{otherwise.} \end{cases}$$

The above formulation can be generalized to the case where b is a vector of length $\binom{n}{j}$ ($j \geq 2$) with the s-th entry in b characterizing a quantity associated with a j-set (a set with

cardinality j). The dictionary A will then be a binary matrix $R^{j,k}$ with entries indicating whether a j-set is a subset of a k-set (or k-clique), i.e.,

$$R_{st}^{j,k} = \begin{cases} 1 & \text{if the } s-\text{th } j-\text{set of nodes all lie in the } k-\text{clique } C_t; \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, the case where b is the vector of length $\binom{n}{2}$ corresponds to a special case where $A = R^{2,k}$. Our algorithms and theory hold for general $R^{j,k}$ with $j < k$.

Now we provide two concrete reconstruction programs for the clique identification problems

$$(\mathscr{P}_1)\ \min \|x\|_1 \mathrm{s.t.} b = Ax$$

$$(\mathscr{P}_{1,\delta})\ \min \|x\|_1 \mathrm{s.t.} \|Ax - b\|_\infty \leq \delta.$$

$\mathcal{P}_1$ is known as Basis Pursuit [11], where we consider an ideal case that the noise level is zero. For robust reconstruction against noise, we consider the robust formulation $\mathcal{P}_{1,\delta}$. The program in $\mathcal{P}_{1,\delta}$ differs from the Dantzig selector [10] which uses the constraint in the form $\|A^*(Ax - b)\|_\infty \leq \delta$. The reason why we choose the bounded noise in the formulation $\mathcal{P}_{1,\delta}$ lies in the fact that it is a natural noise model for network data, due to the homogenous property of the matrix A. Moreover, if we use other noise assumptions, e.g., the Gaussian noise model, the optimization formulation would have higher complexity. In our formulation, however, the bouded noise assumption leads to the linear programming formulation of $\mathcal{P}_{1,\delta}$, which enables practical computation for large scale problems.

B. Network Data Sparse Representation

Let $G = (V, E)$ be the network we are trying to model. The set of vertices V represents individual identities such as people in the social network. Each edge in E is associated with some weights which represent interaction frequency information.

We assume that there are several common interest groups or communities within the network, represented by cliques (or complete sub-graphs) within graph G, which are perhaps of different sizes and may have overlaps. Every community has certain interaction frequency which can be viewed as a function on cliques. Such interaction frequency can be interpreted as the number of co-appearance as we discussed in Example 2. However, we only receive partial measurements consisting of low order interaction frequency on subsets in a clique. For example, in the simplest case we may only observe pairwise interactions represented by edge weights. Our problem is to reconstruct the function on cliques from such partially observed data. A graphical illustration of this idea is provided in Fig. 1, in which we see an observed network can be written as a linear combination of several overlapped cliques. As a

side note, the problem of factorizing graphs into cliques relates to the Hammersley-Clifford Theorem in probability theory [25].

One application scenario is to identify two basketball teams from pairwise interactions among players as discussed in the introduction. Suppose we have x_0 which is a signal on all 5-sets of a 10-player set. We assume it is sparsely concentrated on two 5-sets which correspond to the two teams with nonzero weights. Assume we have observations b of pairwise interactions $b = Ax_0 + z$, where z is uniform random noise defined on $[-\varepsilon, \varepsilon]$. We solve $\mathcal{P}_{1,\delta}$ with $\delta = \varepsilon$, which is a linear program over $x \in \mathbb{R}^{\binom{10}{5}} = \mathbb{R}^{252}$ with parameters $A \in \mathbb{R}^{\binom{10}{2} \times \binom{10}{5}} = \mathbb{R}^{45 \times 252}$ and $b \in \mathbb{R}^{45}$.

Throughout all the application examples discussed in this paper, we are typically interested in identifying cliques with positive weights. However, the theories and examples developed in this paper apply generally for arbitrary weights, either positive or negative. On the other hand, when we apply our algorithm to real application examples, we do see a few negative weights arise. Those negative weights can be explained as offsets of the detected cliques which have over-estimated positive weights. Thus, in the optimization formulation $\mathcal{P}_1$ and $\mathcal{P}_{1,\delta}$ there's no need for an extra non-negative constraint on the variables.

C. Connection With Radon Basis

Let V_k (V_j) denote the set of all k-sets (j-sets) of $V = \{1,\ldots, n\}$ and $\mathcal{H}^k$ ($\mathcal{H}^j$) be the set of real-valued functions on V_k (V_j). We build a matrix $R^{j,k} : \mathcal{H}^k \to \mathcal{H}^j$ ($k > j$) as a mapping from functions on all k-sets of V to functions on all j-sets of V as the following. Each row of $R^{j,k}$ represents a j-set and each column represents a k-set. The entries of $R^{j,k}$ are either 0 or 1 indicating whether the j-set is a subset of the k-set. Note that every column of $R^{j,k}$ has $\binom{k}{j}$ ones. Lacking *a priori* information, we assume that every j-set of a particular k-set has equal interaction probability, so that we choose the same constant 1 for each column. To summarize, we have

$$R^{j,k}_{st} = \begin{cases} 1 & \text{if} \sigma_s \subset \tau_t \\ 0 & \text{otherwise} \end{cases}$$

where σ_s is a j-set corresponding to the index s and τ_t is a k-set corresponding to the index t where $1 \le s \le \binom{n}{j}$ and $1 \le t \le \binom{n}{k}$. As we will see, this construction leads to a canonical basis associated with the discrete Radon transform. The size of matrix $R^{j,k}$ clearly depends on the total number of items $n = |V|$. We omit n as its meaning will be clear from the context.

The matrix $R^{j,k}$ constructed above is related to discrete Radon transforms on homogeneous space $\mathcal{M}^k$. In fact, up to a constant, the adjoint operator $(R^{j,k})^*$ is called the discrete Radon transform from homogeneous space $\mathcal{M}^j$ to $\mathcal{M}^k$ in [16]. Here all the k-sets form a homogeneous space. The collection of all row vectors of $R^{j,k}$ is called as the j-th order *Radon basis* for $\mathcal{M}^k$. Our usage here is to exploit the transpose matrix of the Radon transform to construct an over-complete dictionary for $\mathcal{M}^j$, so that the observation b can be represented by a possibly sparse function $x \in \mathcal{M}^k$ ($k \geq j$).

The Radon basis was proposed as an efficient way to study partially ranked data in [16], where it was shown that by looking at low order Radon coefficients of a function on $\mathcal{M}^k$, we usually get useful and interpretable information. The approach here adds a reversal of this perspective, i.e., the reconstruction of sparse high order functions from low order Radon coefficients. We will discuss this in the sequel with a connection to the compressive sensing [9], [11].

In the remainder of the paper, when we use the matrix $R^{j,k}$ (sometimes, we denote it as A) in the optimization problems, we assume that it is normalized so that each column's ℓ_2-norm equals to 1. In other words, we have

$$R_{st}^{j,k} = \begin{cases} \frac{1}{\sqrt{\binom{k}{j}}}, & \text{if} \sigma_s \subset \tau_t; \\ 0, & \text{otherwise} \end{cases}$$

where again, σ_s is a j-set corresponding to the index s and τ_t is a k-set corresponding to the index t.

Before we proceed, we would like to gather some of the most important facts about the matrix $R^{j,k}$. These facts would be useful for understanding theorems and algorithms presented in later sections.

- The matrix $R^{j,k}$ is of size $\binom{n}{j}$-by-$\binom{n}{k}$.
- Each row of the matrix $R^{j,k}$ can be indexed by a j-set; each column of the matrix $R^{j,k}$ can be indexed by a k-set. Sometimes, we will use j-sets and k-sets to index the rows and columns of $R^{j,k}$ in later sections.
- The entries in $R^{j,k}$ are either $1/\sqrt{\binom{k}{j}}$ or 0. Every row in the matrix $R^{j,k}$ has the $\binom{n-j}{k-j}$ non-zero entries; every column in the matrix $R^{j,k}$ has $\binom{k}{j}$ non-zero entries.
- Two columns indexed by τ_1 and τ_2 (where τ_1 and τ_2 are two k-sets) are orthogonal if and only if $|\tau_1 \bigcap \tau_2| \leq j - 1$.

III. A Theory of Recovery

In this section, we detail our new framework on compressive network analysis theories, which enable rigorous theoretical analysis of the corresponding convex programs.

A. Failure of Universal Recovery

Recently it was shown by [8] and [9] that $\mathcal{P}_1$ has a unique sparse solution x_0, if the matrix A satisfies the *Restricted Isometry Property* (RIP), i.e., for every subset of columns $T \subset \{1,\ldots, N\}$ with $|T| \leq s$, there exists a certain universal constant $\delta_s \in [0, \sqrt{2} - 1)$ such that

$$(1-\delta_s)\|x\|_2^2 \leq \|A_T x\|_2^2 \leq (1+\delta_s)\|x\|_2^2, \forall x \in \mathbb{R}^{|T|}$$

where A_T is the sub-matrix of A with columns indexed by T. Then exact recovery holds for all s-sparse signals x_0 (i.e., x_0 has at most s non-zero components), whence called the *universal recovery*.

Unfortunately, in our construction of the basis matrix A, RIP is not satisfied unless for very small s. The following theorem illustrates the failure of universal recovery in our case.

Theorem 3.1—Let $n > k + j + 1$ and $A = R^{j,k}$ with $j < k$. Unless $s < \begin{pmatrix} k+j+1 \\ k \end{pmatrix}$, there does not exist a $\delta_s < 1$ such that the inequalities

$$(1-\delta_s)\|x\|_2^2 \leq \|A_T x\|_2^2 \leq (1+\delta_s)\|x\|_2^2, \forall x \in \mathbb{R}^{|T|}$$

hold universally for every $T \subset \{1,\ldots, N\}$ with $|T| \leq s$, where $N = \begin{pmatrix} n \\ k \end{pmatrix}$.

Note that $\begin{pmatrix} k+j+1 \\ k \end{pmatrix}$ does not depend on the network size n, which will be problematic. We can only recover a constant number of cliques no matter how large the network is. The main issue here is that the RIP tries to provide exact recovery guarantees for sparse signals with arbitrary support. For many applications, such a condition is too strong to be realistic. Instead of studying such "universal" conditions, in this paper we seek conditions that secure exact recovery of a collection of sparse signals x_0, whose sparsity pattern satisfies certain conditions more appropriate to our setting. Such conditions could be more natural in reality, which will be shown in the sequel as simply requiring bounded overlaps between cliques. In other words, the conditions that we propose to guarantee exact recovery or stable recovery is the RIP restricted to a class of signals, rather than arbitrary signals.

Remark 3.2—Recall from the end of the last section that the matrix A has altogether $N=\begin{pmatrix} n \\ k \end{pmatrix}$ columns. Each column in fact corresponds to a k-clique. Therefore, we could also use a k-clique to index a column of A. In this sense, let $T = \{i_1,\ldots, i_s\} \subset \{1,\ldots, N\}$ be a subset of size s. An equivalent notation is to represent T as a class of sets: $T = \{\tau_1, \ldots, \tau_s\}$ where each $\tau_i \subset \{1,\ldots, n\}$ and $|\tau_i| = k$. We will abuse this notation method also in proofs of later results.

Proof of Theorem 3.1: We can extract a set of columns $T = \{\tau: \tau \subset \{1,2, \ldots, k + j + 1\}$ and $|\tau| = k\}$ (τ is interpreted as a k-set) and form a submatrix A_T. Recall that A has altogether $\begin{pmatrix} n \\ j \end{pmatrix}$ rows, combined with the condition that $n > k + j + 1$ and the fact that the number of nonzero rows of A_T should be exactly $\begin{pmatrix} k+j+1 \\ j \end{pmatrix}$, we know that there exist rows in A_T which only contains zeroes.

By discarding rows with all zero entries, it is easy to show that the rank of A_T is at most $\begin{pmatrix} k+j+1 \\ j \end{pmatrix}$, which is less than the number of columns. To see that the rank of A_T is at most $\begin{pmatrix} k+j+1 \\ j \end{pmatrix}$, we need to use the fact that $j < k$ which implies that

$$\begin{pmatrix} k+j+1 \\ j \end{pmatrix} < \begin{pmatrix} k+j+1 \\ k \end{pmatrix} \quad (3.1)$$

from which we can see that the number of nonzero rows of A_T is smaller than the number of columns.

Thus, the columns in A_T must be linearly dependent. In other words, there exist a nonzero vector $h \in \mathbb{R}^N$ where $\text{supp}(h) \subset T$ such that $Ah = 0$. When $s \geq \begin{pmatrix} k+j+1 \\ k \end{pmatrix}$, since $|\text{supp}(h)| \leq |T| < s$, we can not expect universal sparse recovery for all s-sparse signals.

B. Exact Recovery Conditions

Here, we present our exact recovery conditions for x_0 from the observed data b by solving the linear program $\mathcal{P}_1$. Suppose A is an M-by-N matrix and x_0 is a sparse signal. Let $T = \text{supp}(x_0)$, T^c be the complement of T, and A_T (or A_{T^c}) be the submatrix of A where we only extract column set T (or T^c, respectively). The following proposition from [9] characterizes the conditions that $\mathcal{P}_1$ has a unique condition.

Proposition 3.3—Let $x_0 = (x_{01},\ldots, x_{0N})^*$, we assume that $A_T^* A_T$ is invertible and there exists a vector $w \in \mathbb{R}^M$ such that:

1. $\langle A_j, w\rangle = \text{sign}(x_{0j}), \forall j \in T$;
2. $|\langle A_j, w\rangle| < 1, \forall j \in T^c$.

Then x_0 is the unique solution for $\mathcal{P}_1$ [9].

The above proposition (see proofs in [9]) points out the necessary and sufficient condition that in the noise-free setting $\mathcal{P}_1$ exactly recover the sparse signal x_0. The necessity and sufficiency comes from the KKT condition in convex optimization theory [9]. However, this condition is difficult to check due to the presence of w. If we further assume that w lies in the column span of A_T, the condition in Proposition 3.3 reduces to the following condition.

Irrepresentable Condition (IRR)—The matrix A satisfies the IRR condition with respect to $T = \text{supp}(x_0)$, if $A_T^* A_T$ is invertible and

$$\|A_{T^c}^* A_T (A_T^* A_T)^{-1}\|_\infty < 1$$

or, equivalently

$$\|(A_T^* A_T)^{-1} A_T^* A_{T^c}\|_1 < 1$$

where $\|\cdot\|_\infty$ and $\|\cdot\|_1$ stand for the matrix norms, i.e., $\|A\|_\infty := \max_i \Sigma_j |A_{ij}|$ and $\|A\|_1 = \max_j \Sigma_i |A_{ij}|$. In other words, the $\|\cdot\|_\infty$ is the maximum absolute row sum of the matrix, while $\|\cdot\|_1$ is the maximum absolute column sum of the matrix. In the literature of sparse representation, there is a notion called "Exact Recovery Coefficient" proposed by [45], which can be shown to be equivalent to the irrepresentable condition.

Proposition 3.4—By restricting that w lies in the image of A_T, the conditions in proposition 3.3 reduce to the IRR condition.

Before we provide the proof for Proposition 3.4, we note that such a proposition has been discussed in [20].

Proof of Proposition 3.4: Since w lies in the image of A_T, we can write $w = A_T v$. To make sure that the first condition in Proposition 3.3 holds, we must have $v = (A_T^* A_T)^{-1} \text{sign}(x_0)$, so

$$w = A_T (A_T^* A_T)^{-1} \text{sign}(x_0).$$

Now the second condition in proposition 3.3 can be equivalently written as

$$\|A_{T^c}^* A_T (A_T^* A_T)^{-1}\|_\infty < 1$$

under the assumption that $w = A_T (A_T^* A_T)^{-1} \mathrm{sign}(x_0)$, which is exactly the IRR condition.

Intuitively, the IRR condition requires that, for the ground truth sparsity signal x_0, the relevant bases in A_T are not highly correlated with irrelevant bases in A_T^c. Note that this condition only depends on A and the support of x_0, which is easier to check. The assumption that w lies in the column span of A_T is mild [10], [44], [51], [52]. Without this assumption, the IRR condition can be relaxed to the existence of $\omega \in \ker(A_T^*)$ such that $\|A_{T^c}^* A_T (A_T^* A_T)^{-1} + A_{T^c}^* \omega\|_\infty < 1$, where one can choose $\omega = 0$ to avoid the worst-case.

C. Detecting Cliques of Equal Size

In this subsection, we present sufficient conditions of IRR which can be easily verified. We consider the case that $A = R^{j,k}$ with $j < k$. Given data b about all j-sets, we want to infer important k-cliques. Suppose x_0 is a sparse signal on all k-cliques. We have the following theorem, which is a direct result of Lemma 3.6.

Theorem 3.5—Let $T = \mathrm{supp}(x_0)$, if we enforce the overlaps among k-cliques in T to be no larger than r, then $r \leq j - 2$ guarantees the IRR condition.

The above theorem is a direct result from the following lemma.

Lemma 3.6—Let $T = \mathrm{supp}(x_0)$ and $j \geq 2$. Suppose for any $s_1, s_2 \in T$, the two k-sets σ_1, σ_2 corresponding to s_1, s_2 have overlaps no larger than r, we have:

1. If $r \leq j - 2$, then $\|A_{T^c}^* A_T (A_T^* A_T)^{-1}\|_\infty < 1$;
2. If $r = j - 1$, then $\|A_{T^c}^* A_T (A_T^* A_T)^{-1}\|_\infty \leq 1$ where equality holds with certain examples;
3. If $r = j$, then there exist examples such that $\|A_{T^c}^* A_T (A_T^* A_T)^{-1}\|_\infty > 1$.

One thing to note is that Theorem 3.5 is only an easy-to-verify condition based on the worst-case analysis, which is sufficient but not necessary. In fact, what really matters is the IRR condition. It uses a simple characterization of allowed clique overlaps which guarantees the IRR Condition. Specifically, clique overlaps no larger than $j - 2$ is sufficient to guarantee the exact sparse recovery by $\mathcal{P}_1$, while larger overlaps may violate the IRR Condition. Since this theorem is based on a worst-case analysis, in real applications, one may encounter examples which have overlaps larger than $j - 2$ while $\mathcal{P}_1$ still works.

In summary, IRR is sufficient and almost necessary to guarantee exact recovery. Theorem 3.5 tells us the intuition behind the IRR is that *overlaps among cliques must be small enough*, which is easier to check. In the next subsection, we show that IRR is also sufficient to guarantee stable recovery with noises.

Remark 3.7—We need to make a clear distinction between two statements: "σ, τ have no overlaps" and "σ, τ have no overlaps on any j-sets" (σ, τ are two k-cliques). The first statement is equivalent to $\sigma \bigcap \tau = \emptyset$, while the second statement is not. However, if σ, τ have no overlaps on any j-sets, then the columns in the matrix A correspond to σ and τ are orthogonal.

In the following, we construct explicit conditions which allow large overlaps while the IRR still holds, as long as such heavy overlaps do not occur too often among the cliques in T. The existence of a partition of T in the next theorem is a reasonable assumption in the network settings where network hierarchies exist. In social networks, it has been observed by [22] that communities themselves also join together to form meta-communities. The assumptions that we made in the next theorem where the allowed overlaps are dependent on whether the two communities are from the same meta-community characterize such a scenario.

Theorem 3.8—Assume $(k + 1)/2 \leq j < k$. Let $T = \text{supp}(x_0)$. Suppose there exist a partition $T = T_1 \bigcup T_2 \bigcup \ldots \bigcup T_m$ with each T_i satisfies $|T_i| \leq K$, such that:

- for any $s_1, s_2 \in T$ indexed by σ_i, σ_j belong to the same partition, $|\sigma_i \bigcap \sigma_j| \leq r$;
- for any $s_1, s_2 \in T$ indexed by σ_i, σ_j belong to different partitions, $|\sigma_i \bigcap \sigma_j| \leq 2j - k - 1$.

If $C_1(K, k, j, r) < 1/4$ and $C_2(K, k, j, r) \leq 3/4$, then IRR holds. Here

$$C_1(K,k,j,r)=(K-1)\begin{pmatrix} r \\ j \end{pmatrix} / \begin{pmatrix} k \\ j \end{pmatrix}$$

$$C_2(K,k,j,r)=\left(\begin{pmatrix} k-1 \\ j \end{pmatrix}+(K-1)\begin{pmatrix} (k+r)/2 \\ j \end{pmatrix}\right) / \begin{pmatrix} k \\ j \end{pmatrix}.$$

Remark 3.9—In the theorem above, there is no constraint on m in order that IRR holds. Such a result means that we can potentially recover sparse network cliques within an arbitrarily large network under the condition of Theorem 3.8.

Proof of Theorem 3.8: We will show the following two inequalities hold.

$$\|A_T^* A_T - I\|_\infty \leq C_1(K,k,j,r), \|A_{T^c}^* A_T\|_\infty \leq C_2(K,k,j,r).$$

We first bound the sup-norm of $A_T^* A_T - I$. Note that when σ_i and σ_j belong to different partitions of T, then $|\sigma_i \bigcap \sigma_j| = 0$ because their overlap is no larger than $2j - k - 1$ which is strictly smaller than j. So $A_T^* A_T$ is a block diagonal matrix with block sizes $|T_1|, |T_2|, \ldots, |T_m|$, and each diagonal entry of $A_T^* A_T$ is one.

Thus, for any $\sigma \in T$, only cliques from the same partition as σ may have overlaps with σ greater than j. Thus, the row sum of $A_T^* A_T - I$ can be bounded by $C_1(K, k, j, r)$. So the first inequality is now established.

To prove the second inequality, we observe that for a fixed $\tau \in T^c$, $|\tau \bigcap \sigma_i| \geq j$ and $|\tau \bigcap \sigma_j| \geq j$ can not hold at the same time for any σ_i and σ_j belong to different partitions. This is because otherwise, we will have

$$|\tau| \geq |\tau \cap (\sigma_i \cup \sigma_j)| = |\tau \cap \sigma_i| + |\tau \cap \sigma_j| - |\tau \cap \sigma_i \cap \sigma_j| \geq j + j - (2j - k - 1) = k + 1.$$

Thus, all σ's such that $|\sigma \bigcap \tau| \geq j$ must lie in the same partition of T.

For the same reason, we can show that for a fixed $\tau \in T^c$, $|\tau \bigcap \sigma_i| \geq (k + r + 1)/2$ and $|\tau \bigcap \sigma_j| \geq (k + r + 1)/2$ can not hold at the same time for σ_i and σ_j belong to the same partition of T. This is because otherwise, we will have

$$|\tau| \geq |\tau \cap (\sigma_i \cup \sigma_j)| = |\tau \cap \sigma_i| + |\tau \cap \sigma_j| - |\tau \cap \sigma_i \cap \sigma_j| \geq (k + r + 1)/2 + (k + r + 1)/2 - r = k + 1.$$

Thus we know the maximum row sum of $A_{T^c}^* A_T$ is bounded from above by $C_2(K, k, j, r)$.

Now if $C_1(K, k, j, r) < 1/4$ and $C_2(K, k, j, r) \leq 3/4$, then we have

$$\|A_T^* A_T - I\|_\infty < 1/4, \|A_{T^c}^* A_T\|_\infty < 3/4.$$

Thus

$$\begin{aligned} \|A_{T^c}^* A_T (A_T^* A_T)^{-1}\|_\infty &\leq \|A_{T^c}^* A_T\|_\infty \|(A_T^* A_T)^{-1}\|_\infty \leq \|A_{T^c}^* A_T\|_\infty (1 \\ &+ \sum_{i=1}^{\infty} \| \left(A_T^* A_T - I \right) \|_\infty^i < 3/4 \, (1 \\ &+ \sum_{i=1}^{\infty} (1/4)^i \Big) = 1 \end{aligned}$$

where the second inequality follows from a matrix geometric expansion. In general, for any invertable matrix M, we have

$$\|M^{-1}\|_\infty = \|(I - (I - M))^{-1}\|_\infty = \|I + \sum_{i=1}^{\infty} (I - M)^i\|_\infty \leq 1 + \sum_{i=1}^{\infty} \|I - M\|_\infty^i.$$

So IRR holds under our conditions.

The basis matrix $A = R^{j,k}$ have $\binom{n}{k}$ bases, which is not polynomial with respect to k. As we will see from later sections, a practical implementation of the Radon basis pursuit for the clique detection problem works on a subset of bases among all $\binom{n}{k}$ bases. In that case, we are actually solving $\mathcal{P}_1$ and $\mathcal{P}_{1,\delta}$ with the basis matrix $\bar{A}$, which is only a submatrix of A with a subset of column bases extracted. If we work with the submatrix $\bar{A}$, then we can potentially impose stronger conditions regarding column overlappings. In such a scenario, we can impose stronger conditions on $\bar{A}$ to obtain successful recovery theorems. We have the following theorem regarding this scenario.

Theorem 3.10—Denote the set of all cliques for columns in $\bar{A}$ by S and $T = \mathrm{supp}(x_0) \subset S$, where $\bar{A}$ is a submatrix of A. Assume any two k-cliques corresponding to column indices of $\bar{A}$ have intersections at most r. Then for the basis matrix $\bar{A}$, IRR holds if

$$r \leq \left(\frac{1}{|T|(1+\sqrt{|T|})} \right)^{\frac{1}{j}} k. \quad (3.2)$$

Proof of Theorem 3.10: Let T^c be the complement of T with respect to S. Note that

$$\|\overline{A}^*_{T^c}\overline{A}_T(\overline{A}^*_T\overline{A}_T)^{-1}\|_\infty \leq \|\overline{A}^*_{T^c}\overline{A}_T\|_\infty \|(\overline{A}^*_T\overline{A}_T)^{-1}\|_\infty \leq \|\overline{A}^*_{T^c}\overline{A}_T\|_\infty \cdot \sqrt{|T|}\|(\overline{A}^*_T\overline{A}_T)^{-1}\|_2$$

where in the last inequality $\|\cdot\|_2$ is the matrix spectral norm. In general, for a matrix M, $\|M\|_2$ is the square root of the largest eigenvalue of the positive-semidefinite matrix M^*M. So it suffices to show

$$\|\overline{A}^*_{T^c}\overline{A}_T\|_\infty \cdot \sqrt{|T|}\|(\overline{A}^*_T\overline{A}_T)^{-1}\|_2 < 1$$

under condition (3.2).

Firstly

$$\|\overline{A}^*_{T^c}\overline{A}_T\|_\infty = \max_{\tau \in T^c} \sum_{\sigma \in T} \frac{\binom{|\tau \cap \sigma|}{j}}{\binom{k}{j}} \leq \frac{|T|\binom{\tau}{j}}{\binom{k}{j}}$$

since $|\tau \cap \sigma| \leq r$. So at least we need

$$|T|\binom{r}{j}/\binom{k}{j}<1. \quad (3.3)$$

Secondly, let $K=\overline{A}_T^*\overline{A}_T$, then

$$K_{ii}=1$$

and since $\forall\ \sigma_i,\ \sigma_j \in T$, $|\sigma_i \bigcap \sigma_j| \leq r$, we have

$$K_{ij} \leq \frac{\binom{r}{j}}{\binom{k}{j}}.$$

Under condition (3.3), K is diagonal dominant, i.e.,

$$K_{ii}>\sum_{j\neq i}|K_{ij}|.$$

Then by Gershgorin Circle Theorem,

$$\lambda_{\min} \geq 1-\sum_{j\neq i}|K_{ij}| \geq 1-(|T|-1)\binom{r}{j}/\binom{k}{j} \geq 1-|T|\binom{r}{j}/\binom{k}{j}.$$

Therefore, it suffices to have

$$\frac{|T|\binom{r}{j}}{\binom{k}{j}}\frac{\sqrt{|T|}}{1-|T|\binom{r}{j}/\binom{k}{j}}<1$$

which gives

$$\binom{r}{j}<\frac{1}{|T|\left(1+\sqrt{|T|}\right)}\binom{k}{j}.$$

This is equivalent to

$$r(r-1)\cdots(r-j+1)<\frac{k(k-1)\cdots(k-j+1)}{|T|\left(1+\sqrt{|T|}\right)}$$

i.e.,

$$\frac{r(r-1)\cdots(r-j+1)}{k(k-1)\cdots(k-j+1)}<\frac{1}{|T|\left(1+\sqrt{|T|}\right)}.$$

It is sufficient if

$$r<\left(\frac{1}{|T|\left(1+\sqrt{|T|}\right)}\right)^{\frac{1}{j}}k$$

because then we would have

$$\frac{r(r-1)\cdots(r-j+1)}{k(k-1)\cdots(k-j+1)}\leq\frac{r^j}{k^j}<\frac{1}{|T|\left(1+\sqrt{|T|}\right)}.$$

D. Stable Recovery Theorems

In applications, one always encounters examples with noise such that exact sparse recovery is impossible. In this setting, $\mathcal{P}_{1,\delta}$ will be a good replacement of $\mathcal{P}_1$ as a robust reconstruction program. Here we present stable recovery theorem of $\mathcal{P}_{1,\delta}$ with bounded noise, which can be viewed as a result of implications of restricted isometry property [8] with noises.

Theorem 3.11—Under the general framework (2.1), we assume that $\|z\|_\infty \leq \epsilon$, $|T| = s$, and the IRR

$$\left\|A_{T^c}^* A_T (A_T^* A_T)^{-1}\right\|_\infty \leq \alpha \leq 1/s.$$

Then the following error bound holds for any solution $x_{\hat{\delta}}$ of $\mathcal{P}_{1,\delta}$

$$\left\|\hat{x}_\delta - x_0\right\|_1 \leq \frac{2s(\epsilon+\delta)}{1-\alpha s}\left\|A_T (A_T^* A_T)^{-1}\right\|_1. \quad (3.4)$$

Proof of Theorem 3.11: Let $h = x_{\hat{\delta}} - x_0$. Note that $\|Ax_{\hat{\delta}} - b\|_\infty \leq \delta$ and $z = Ax_0 - b$ with $\|z\|_\infty \leq \epsilon$. Then

$$\|Ah\|_\infty = \|A\hat{x}_\delta - Ax_0\|_\infty = \|A\hat{x}_\delta - b + b - Ax_0\|_\infty \le \|A\hat{x}_\delta - b\|_\infty + \|z\|_\infty \le \delta + \epsilon. \quad (3.5)$$

We denote $x_{\delta|T}$ as constraining $x_{\hat{\delta}}$ on the support T, i.e. all the entries of $x_{\hat{\delta}}$ corresponding to T^c will be set to zero. From the optimization problem in ($\mathcal{P}_{1,\delta}$), we know that $\|x_0\|_1 \ge \|x_{\delta|T}\|_1$

$$\|h_T\|_1 = \|x_0 - \hat{x}_\delta|T\|_1 \ge \|x_0\|_1 - \|\hat{x}_\delta|T\|_1 \ge \|\hat{x}_\delta\|_1 - \|\hat{x}_{\delta|T}\|_1 = \|\hat{x}\delta|T^c\|_1 = \|h_{T^c}\|_1. \quad (3.6)$$

Therefore

$$\begin{aligned}
&\left|\left\langle Ah, A_T(A_T^*A_T)^{-1}h_T\right\rangle\right| \\
&\quad = \left|\left\langle A_T h_T, A_T(A_T^*A_T)^{-1}h_T\right\rangle + \left\langle A_{T^c}h_{T^c}, A_T(A_T^*A_T)^{-1}h_T\right\rangle\right| \ge \|h_T\|_2^2 \\
&\quad - \left|\left\langle h_{T^c}, A_{T^c}^*A_T(A_T^*A_T)^{-1}h_T\right\rangle\right| \ge \|h_T\|_2^2 \\
&\quad - \|h_{T^c}\|_1 \|A_{T^c}^*A_T(A_T^*A_T)^{-1}h_T\|_\infty \ge \frac{1}{s}\|h_T\|_1^2 \\
&\quad - \alpha\|h_{T^c}\|_1\|h_T\|_\infty \ge \frac{1}{s}\|h_T\|_1^2 \\
&\quad - \alpha\|h_{T^c}\|_1\|h_T\|_1 \ge \left(\frac{1}{s} - \alpha\right)\|h_T\|_1^2
\end{aligned}$$

where the last step is due to $\|h_T\|_1 \ge \|h_{Tc}\|_1$ in the inequality (3.6). On the other hand, using (3.5) we know

$$\left|\left\langle Ah, A_T(A_T^*A_T)^{-1}h_T\right\rangle\right| \le \|Ah\|_\infty \|A_T(A_T^*A_T)^{-1}h_T\|_1 \le (\delta+\epsilon)\|A_T(A_T^*A_T)^{-1}\|_1\|h_T\|_1.$$

Combining these two inequalities yields

$$\|h_T\|_1 \le \frac{s(\delta+\epsilon)}{1-\alpha s}\|A_T(A_T^*A_T)^{-1}\|_1$$

as desired.

In the special case where $k = j + 1$, we have:

Corollary 3.12—Let $k = j + 1$, $|T| = s$, and for any $s_1, s_2 \in T$ indexed by σ_1, σ_2, the two cliques σ_1 and σ_2 have overlaps no larger than $j - 2$. Then we have $\|A_{T^c}^*A_T(A_T^*A_T)^{-1}\|_\infty \le 1/(j+1)$, and thus the following error bound for solution $x_{\hat{\delta}}$ of $\mathcal{P}_{1,\delta}$ holds:

$$\|\hat{x}_\delta - x_0\|_1 \le \frac{2s(\epsilon+\delta)}{1-\frac{s}{j+1}}\sqrt{j+1}, s<j+1.$$

Proof of Corollary 3.12: This corollary follows from the previous theorem. Note that when the conditions in Theorem 2 hold, $A_T^* A_T = I$ and $\|A_T\|_1 \le \sqrt{\binom{k}{j}} = \sqrt{j+1}$.

Now it suffice to establish the fact that in this special case, we have

$$\|A_{T^c}^* A_T (A_T^* A_T)^{-1}\|_\infty \le \frac{1}{j+1} < 1.$$

Note that since any $\sigma_1, \sigma_2 \in T$ satisfy $|\sigma_1 \bigcap \sigma_2| \le j-2$, we have $A_T^* A_T$ is an identity matrix. So $\|A_{T^c}^* A_T (A_T^* A_T)^{-1}\|_\infty = \|A_{T^c}^* A_T\|_\infty$. Now assume $\tau \in T^c$, let $S_\tau = \{\sigma : |\sigma \bigcap \tau| \ge j, \sigma \in T\}$, then $|S_\tau| \le 1$. This is because otherwise, suppose $\{\sigma_1, \sigma_2\} \subset S_\tau$ such that $|S_\tau| \ge 2$, then we have

$$|\tau| \ge |\tau \cap (\sigma_1 \cup \sigma_2)| = |\tau \cap \sigma_1| + |t \cap \sigma_2| - |t \cap \sigma_1 \cap \sigma_2| \ge j + j - (j-2) = j+2$$

which contradicts with the fact that τ is a $j+1$-set. So there exist at most one $\sigma_0 \in T$ such that $|\tau \bigcap \sigma_0| \ge j$. However, since $|\tau \bigcap \sigma| \le j$, it follows that whenever there exist such a σ_0, the equality holds. Under such a scenario, let v_τ be the row vector of $A_{T^c}^* A_T$ with row index correspond to τ. Then $\|v_\tau\|_\infty \le \binom{j}{j} / \binom{j+1}{j} = 1/(j+1) < 1$.

We note that in applications, recovering the support is very important because the support represents a meaningful set of cliques that explains the observation data. In the exact recovery setting, successful recovery of signals naturally infers that we successfully recovered the support of signals. In Theorem 3.11, we uses the ℓ_1-norm to bound the recovered signal $x_{\hat{\delta}}$ and the groudtruth signal x_0. When the signal strenghth is strong enough, this implies that we can identify the support of the ground truth signal x_0. In other words, when signal to noise ratio is high, Theorem 3.11 infers successful recovery of support of x_0.

E. Identifying Cliques With Mixed Sizes

In general settings, we need to identify high order cliques of mixed sizes, i.e., cliques of sizes $k_1, k_2, \ldots, k_\ell (j \le k_1, k_2, \ldots, k_g \le k)$, based on the observed data b on all j-sets. One way to construct the basis matrix A is by concatenating $R^{j,k}$ with different k's satisfying $k > j$. We can then solve $\mathcal{P}_1$ and $\mathcal{P}_{1,\delta}$ for exact recovery and stable recovery with this newly concatenated basis matrix A. We have the following proposition:

Proposition 3.13—Suppose x_0 is a sparse signal on cliques of sizes $k_1,k_2,\ldots,k_\ell (j \leq k_1, k_2, \ldots,k_\ell \leq k)$ and $b = Ax_0$. Let $T = \text{supp}(x_0)$.

1. If the cliques corresponding to T have no overlaps, then they can be identified by $\mathcal{P}_1$.
2. Moreover, if the data $b = Ax_0 + z$ is contaminated by the noise z, $\mathcal{P}_{1,\delta}$ provides an estimate of x_0 for which the inequality in (3.4) still holds.

The above proposition provides us a sufficient condition to guarantee exact sparse recovery with concatenated bases and the stable recovery theory is also established. We note that Proposition 3.13 aims to provide sufficient conditions to guarantee successful recovery under the worst-case condition. However, in practical applications, the method works better than this worst-case guarantee since the worst-case setting may only happen with small probability. In Section V, we will see that the worst-case propositions are very conservative statements of recovering cliques.

IV. A Polynomial Approximation Algorithm

In practical applications, we have pairwise interaction data in a network with n nodes and we wish to infer high order cliques up to size k. Directly constructing A by concatenating Radon basis matrices $R^{j,j}$, $R^{j,j+1}\ldots$, $R^{j,k}$ and solving $\mathcal{P}_{1,\delta}$ would incur exponential complexity since A has exponentially many columns with respect to k. This would be intractable for inferring high order cliques in large networks. In this section, we describe a polynomial time (with respect to both n and k) approximation algorithm for solving $\mathcal{P}_{1,\delta}$. Recall that the primal and dual programs $\mathcal{P}_{1,\delta}$ and $\mathcal{D}_{1,\delta}$ are

$$(\mathscr{P}_{1,\delta})\ \min \|x\|_1 \mathbf{s.t.} \|Ax - b\|_\infty \leq \delta$$

$$(\mathscr{D}_{1,\delta})\ \max -\delta\|\gamma\|_1 - b^*\gamma \mathbf{s.t.} \|A^*\gamma\|_\infty \leq 1.$$

The key of our algorithm is that we use a polynomial number of variables and constraints to approximate both programs, yielding an approximate solution for $\mathcal{P}_{1,\delta}$. More precisely, we apply a sequential primal-dual interior point method to solve the relaxed programs:

$$(\mathscr{P}_{1,\delta,T})\ \min \|x\|_1 \mathbf{s.t.} \|Ax - b\|_\infty \leq \delta$$

$$(\mathscr{D}_{1,\delta,T})\ \max -\delta\|\gamma\|_1 - b^*\gamma \mathbf{s.t.} \|A_T^*\gamma\|_\infty \leq 1.$$

Here, A_T is a submatrix of A where we extract a subset of columns T. We approximate the solution to the original programs by solving the above relaxed programs where we only use polynomially many columns indexed by T. In particular, we want to find an interior point γ for $\mathcal{D}_{1,\delta,T}$ which is also feasible for $\mathcal{D}_{1,\delta}$. With this γ available, we can use duality gaps to

check convergence because the current dual objective provides a lower bound for $\mathcal{D}_{1,\delta}$ and any interior point for $\mathcal{P}_{1,\delta,T}$ provides an upper bound for $\mathcal{P}_{1,\delta}$.

Let A_i be the i-th column of A. We need to sequentially update the column set T. When we have a solution γ (which is called the approximate analytic center) for the relaxed program $\mathcal{D}_{1,\delta,T}$, we need to find a new column A_i ($i \in T^c$) which is not feasible in $\mathcal{D}_{1,\delta,T}$. By incorporating A_i into T, the feasible region of $\mathcal{D}_{1,\delta,T}$ is reduced to better approximate that of $\mathcal{D}_{1,\delta}$. When the current solution γ has no violated constraint, i.e., γ is feasible for $\mathcal{D}_{1,\delta}$, we use interior point methods to find a series of interior points which converge to the solution of $\mathcal{D}_{1,\delta,T}$. However, we may obtain a new interior point γ which is not feasible for $\mathcal{D}_{1,\delta}$. We then go back and add violated constraints. When we output the solution, we also use a threshold to figure out the support of sparse signal, which we make sure to satisfy the IRR condition. A formal description is provided in Algorithm 1.

Algorithm 1 Cutting Plane Method for Solving $\mathcal{P}_{1,\delta}$

Initialize $A = I$, $x = b$, $\gamma = (1, 1, \ldots, 1)^*$.

while TRUE **do**

  if $\exists |A_i^* \gamma| > 1$ where $i \in T^c$ **then**

    $T \leftarrow T \bigcup \{i\}$, formulate new $\mathcal{D}_{1,\delta,T}$ and $\mathcal{P}_{1,\delta,T}$.

    Find new interior points γ and x for $\mathcal{D}_{1,\delta,T}$ and $\mathcal{P}_{1,\delta,T}$ respectively.

  else if the duality gap is small **then**

    Get the dual solution x, ̂do the IRR check and stop.

  else

    Find a new interior point γ for $\mathcal{D}_{1,\delta,T}$, which optimizes the dual objective.

  end if

end while

In Algorithm 1, the first IF statement involves a problem of finding a violated dual constraint for the current relaxed program. In the special case where γ are dual variables associated with edges, the problem becomes the *maximum edge weight clique problem*, which is known to be NP-hard. We use a simple greedy heuristic algorithm, which iteratively adds new nodes in order to maximize summation of edge weights to solve this problem [36], which runs in $\mathcal{O}(nk^2)$ time and can return a 0.94-approximate solution in the average case. Note that, if γ is feasible for the dual relaxation problem with no additional violated constraints, then 0.94γ must be feasible for $\mathcal{D}_{1,\delta}$ whose objective is discounted by 0.94. Thus, we will terminate with an 0.94-approximate solution.

Let η be the threshold to check the duality gap. Algorithm 1 can also be understood as the column generation method [13], since adding a new inequality constraint in the dual program adds a variable to the primal program and thus adds a column to the basis matrix. For more details of the algorithm, see [37] and [50]. Theoretically, if one is able to find a violated constraint in constant time and uses interior point methods to locate approximate centers of the primal-dual feasible regions, then Algorithm 1 has computational complexity $\mathcal{O}(M/\eta^2)$, where M is the number of dual variables [37], [50]. In our case, $M \asymp n^2$ and find a violated constraint has complexity $\mathcal{O}(nk^2)$, thus algorithm 1 has complexity $\mathcal{O}(n^{3k2}/\eta^2)$.

Finally, we note that other iterative algorithms, e.g., Bregman iterations, which have guaranteed convergence rates [7] can be used to find solutions of linear program relaxations in our algorithms. We also note that, in practice, we never need to explicitly construct the matrix A because there are many combinatorial structures within the basis matrix to exploit. For example, operations such as evaluating inner products between the bases can be evaluated efficiently by directly comparing two sets.

V. Application Examples

In this section, we provide four application examples to illustrate the effectiveness of the proposed framework in this paper. We test these examples with the proposed algorithm described in the last section. We note that whether the IRR condition holds is an important criteria in the execution of our algorithm, which makes sure that our theory are well connected with the experiments. As we will see, our clique-based model can deal with overlaps between cliques which gives us more community structural information compared against using purely clustering methods and the state-of-the-art clique percolation method. In these examples, we use the clique *volume* and *conductance*, which arguably are the simplest evaluation criteria of clustering quality, to evaluate different algorithms. The clique volume is the sum of edge weights inside the clique, while the clique conductance is the ratio between the number of weights leaving the clique and the clique volume [34].

More precisely, let B_{uv} be the element on the u-th row and v-th column of the adjacency matrix B. The *conductance* $\varphi(S)$ of a set of nodes S is defined as

$$\phi(S)=\frac{\sum_{\{(u,v):u\in S,v\notin S\}} B_{uv}}{\min(\mathrm{Vol}(S),\mathrm{Vol}(V\backslash S))}$$

and *volume* is $\mathrm{Vol}(S) = \Sigma_{\{u,v\in S\}} B_{uv}$.

Before we move on to the details of experiments, we would like to briefly describe the clique percolation method which we aim to compare against. In the simplest case, the clique percolation method builds up the communities from k-cliques, which correspond to complete (fully connected) subgraphs of k nodes. Two k -cliques are considered adjacent if they share $k - 1$ nodes. A community is defined as the maximal union of k cliques that can be reached from each other through a series of adjacent k-cliques. Since even small networks can contain a vast number of k-cliques, the implementation of this approach is based on locating the maximal cliques rather than the individual k-cliques. Thus the complexity of this approach in practice is equivalent to the NP-complete maximal clique finding [40]. In the generalied cases, researchers have developed the weighted clique percolation method. In such a case, a weighted k-clique is a complete subgraph with k nodes such that the geometric mean of the $k(k - 1)/2$ edge weights is greater than a prescribed threshold. The weighted clique percolation method defines weighted network communities as the percolation clusters of weighted k-cliques [39].

A. Basketball Team Detection

Detecting two basketball teams from pairwise interactions among plays is an ideal scenario since the two teams do not overlap. This example is discussed in the Example 1 in the introduction section.

Suppose we have x_0 which is the true signal indicating the two teams among all 5-sets of the 10-player set, i.e., it is sparsely concentrated on two 5-sets which correspond to the two teams with magnitudes both equal to one. Assume we have observations b of pairwise interactions, i.e., $b = Ax_0 + z$, where z is bounded random noise uniformly distributed in $[-\varepsilon, \varepsilon]$. We solve $\mathcal{P}_{1,\delta}$ with $\delta = \in$, which is a linear programming search over $x \in \mathbb{R}^{\binom{10}{5}} = \mathbb{R}^{252}$ with a parameter matrix $A \in \mathbb{R}^{\binom{10}{2} \times \binom{10}{5}} = \mathbb{R}^{45 \times 252}$ and $b \in \subset^{45}$. We note that in this example, we set $\delta = \varepsilon$, pretending that we already know the tight bounds of noises. However, in real cases, we need to solve for different δs to get a solution path, so as to decide the optimal solution. We refer the reader to the ranking example described later for more detailed discussion.

In this problem, we solve $\mathcal{P}_{1,\delta}$ directly without using the algorithm provided in the last section due to the small size of the problem. The results of the solution to $\mathcal{P}_{1,\delta}$ are shown in Fig. 2. In Fig. 2(a), we see that the two basketball teams are perfectly detected as expected. Since the two 5-sets correspond to the two teams have no overlap, hence satisfy the irrepresentable Condition (IRR). In Fig. 2(b), we try to detect the two teams under different noise levels $\varepsilon \in [0,1]$. The two basketball teams can be detected under fairly large noise levels. This example can also be dealt with using spectral clustering techniques where we normalize the pairwise interaction data to get the transition matrix, followed by spectral clustering on eigenspaces [42]. We observe that both our method and spectral clustering works very well under noise level less than 0.8 (i.e., $|\varepsilon| < 0.8$).

B. The Social Network of Les Misérables

As we have discussed in the introduction section (Example 2) of detecting communities in social network, we consider the social network of 33 characters in Victor Hugo's novel *Les Misérables* [30]. We represent this social network using a weighted graph [Fig. 3(a)]. The edge weights are the co-appearance frequencies of the two corresponding characters. Table I illustrates several social communities formed by relationships including *friendships, street gangs, kinships*, etc. The underlying social community, regarded as the ground truth for the data, is summarized in Fig. 3(a) where several social communities arise. Fig. 3(b) shows the spectral clustering result in which the first three red cuts are reasonable while the next three blue cuts destroy a lot of community structures within the network.

We aim to detect cliques up to size 9. We construct the matrix A using a concatenation of matrices $R^{2,3}, R^{2,4}, \cdots, R^{2,9}$. In this case, $b \in \mathbb{R}^{\binom{33}{2}} = \mathbb{R}^{528}$. Because of the large size of the problem, we use algorithm described in the last section to solve the problem.

The way of how the algorithm works in this example is that it can quickly identify tens of columns in A which are indexed by the meaningful k-sets ($2 \leq k \leq 9$) corresponding to interesting communities in the social network. Because of the existance of significant noise in the data, the algorithm will continue to search other candidate columns in A until the duality gap is within a prescribed threshold. After a solution x îs obtained, we try to figure out what is the support of x b̂y using a simple thresholding, as well as double checking to make sure that IRR is satisfied.

We use the standard spectral clustering approach as a benchmark algorithm, and we compare our method with the clique percolation method. It turns out that 23 and 19 cliques are identified respectively where our approach can identify more meaningful cliques—see Fig. 3 and Table I where we verify the ground truth from the novel. For example, our method can correctly identify two separate cliques {4, 15, 22} and {20, 21, 22}, while the clique percolation method treats {4, 15, 20, 21, 22} as a single clique. The interaction frequencies among those characters, however, show that there are relatively smaller cross-community interactions, thus those two 3-cliques should be separated. Fig. 3(c) and (d) depict important 3-cliques and 4-cliques identified by our algorithm. The sparsity patterns of those cliques satisfy the irrepresentable condition where overlaps between them are generally not large. However, they do not necessarily satisfy the condition in Lemma 3.6 which is based on a worst-case analysis. In Fig. 4, we also compare both methods in terms of clique conductances and volumes and see that the cliques identified by Radon basis pursuit have slightly lower conductances and larger volumes, which demonstrates advantages of our approach.

From the analysis of the algorithm in the last section, to detect cliques up to size k, we know that our algorithm has complexity $\mathcal{O}(n^3k^2/\eta^2)$, where η is the threshold for the duality gap. The complexity of the clique percolation method in practice is equivalent to the NP-complete maximal clique finding, which can be exponential of the network size. Though the spectral clustering algorithm performs poorly, its complexity is $\mathcal{O}(n^3)$. Thus, our algorithm has a much more reasonable complexity compared with the clique percolation method while achieves a slightly better performance. Though both our algorithm and clique percolation method do not have low complexity as the clustering method, however, their performance overwhelms the clustering based approach.

We test the running time of different algorithms on the Les Misérables dataset, see Table II. We find that the clique percolation has a very short running time despite its worst theoretical complexity. The spectral clustering approach simply performs a bipartite clustering on the network, which has the shortest running time. We also test three different algorithms of solving $\mathcal{P}_{1,\delta}$. The iterative algorithm uses the Bregman iterations to solve the optimization problem [7]. The stagewise algorithm first tries to detect 3-cliques, and then detect 4-cliques, and etc. It can be viewed as solving a series of optimization problems using interior point solvers, each with basis matrix $A = R^{2,k}$, where $k = 3, 4, \ldots, 9$. We also test the algorithm that we described in the last section to solve the problem. We note that our algorithm has better performance than the stagewise algorithm, but worse than the iterative algorithm. However, we note that when we are dealing with large scale datasets (see the next

example), we can not use iterative algorithm and stagewise algorithm, because the number of columns in the basis matrix A is extremely large.

In summary, our method obtains more abundant social structure information than the benchmark spectral clustering approach and the clique percolation techniques with reasonable complexity. It also yields social communities with overlaps which is impossible for clustering methods.

C. Coauthor ships in Network Science

Similar as the last example of detecting communities in social network of Les Misérables, we also study a medium size coauthorship network where there is a total of 1589 scientists who come from a broad variety of fields. Part of this network is shown in Fig. 6(a). There are 136 and 166 cliques identified by our approach and the clique percolation method, respectively. We also compare the two methods in terms of clique conductances and volumes.

We still aim to detect cliques up to size 9. The construction of the matrix A is the same, using a concatenation of matrices $R^{2,3}$, $R^{2,4}$, $\cdots$, $R^{2,9}$ and $b \in \mathbb{R}^{\binom{1589}{2}}$ is a sparse vector. The difference is that in this case the sizes of matrices $R^{j,k}$ are larger because the number of nodes in this coauthorship network is much larger than the social network of Les Misérables. We also use algorithm described in the last section to solve the problem.

From Fig. 5, we see that the cliques identified by Radon basis pursuit have smaller conductances and comparable clique volumes than the clique percolation method. In this example, our approach can identify the cliques up to size 9 in 564 seconds. In general, our approach can be used to identify cliques in social networks with hundreds or even thousands of nodes.

Finally, we note that clustering techniques, e.g., spectral clustering, combined with our algorithm can provide a more refined analysis of the network. We can look at the persistence of identified cliques in the binary tree decomposition of bipartite spectral clustering of the network in a bottom-up way. Cliques which persist through more levels will give us meaningful community structural information.

In Fig. 6(b), a small fraction of the binary tree decomposition of bipartite spectral clustering is depicted, where child nodes are spectral bipartition of the parent node. We can detect cliques within the child nodes. Once cliques within clusters *C, D* are identified, we then backtrack to the parent nodes *B* and *A* to see if the identified cliques still persist.

We can identify 3 cliques (c_1 = {Kumar, Raghavan, Rajagopalan, Tomkins}, c_2 = {Kumar. S, Raghavan, Rajagopalan}, c_3 = {Raghavan, Rajagopalan, Tomkins, Kumar. S}) within C and 3 cliques (d_1 = {Flake. G, Lawrence. S, Giles. C, Coetzee. F}, d_2 = {Flake. G, Lawrence. S, Giles. C, Pennock. D, Glover. E}, d_3 = {Flake. G, Lawrence. S, Giles. C}) within D which persist to parents B and A. We can identify papers whose authors are exactly

those cliques. Using only clustering will not get this result because those cliques have heavy overlaps between them.

In Fig. 6(b), for simplicity, we only show two persistent cliques: c_1 = {Kumar, Raghavan, Rajagopalan, Tomkins} and d_1 = {Flake. G, Lawrence. S, Giles. C, Coetzee. F} which are the most important cliques (having the largest weights when solving the LP program) in clusters *C* and *D* respectively. These two cliques are also the most important two cliques in cluster *B*, and if we even further back track them to clustering *A*, they are still ranked as the first and the third in terms of weights among all cliques identifiable in *A*.

D. Inferring High Order Ranking

As we have discussed in the introduction, we can infer high order partial rankings using the same methodology discussed in this paper. We use the Jester dataset [23] which contains about 24 000 users who provided their ratings to 100 jokes. Those ratings are real values ranging from −10.00 to +10.00. We extract 20 jokes from the entire dataset which have the highest mean scores. Among those 20 jokes, we consider all $\binom{20}{5}$ 5-joke tuples. For each tuple of 5-jokes, we count how many people vote this tuple as his or her top 5. Fig. 7(a) shows that there is a top 5-set, {27, 29, 35, 36, 50}, with an overwhelming votes than other tuples.

The ground truth votes on all 5 tuples can be viewed as a ground truth signal $x_0 \in \mathbb{R}^{\binom{20}{5}}$. Now suppose, we don't know x_0, but we only know votes on all 3 tuples and wonder if we can identify the most popular 5-joke group. Specifically, we construct a matrix $A = R^{3,5} \in \mathbb{R}^{\binom{20}{3} \times \binom{20}{5}}$ and $b \in \mathbb{R}^{\binom{20}{3}}$ summarizes the votes on top 3-jokes. By solving $\mathcal{P}_{1,\delta}$ we hope to be able to identify the top 5-set, {27, 29, 35, 36, 50}. It turns out that by looking at the whole regularization path by varying δ, we are capable to detect this subset [Fig. 7(b)] in a robust way, which consistently have the largest positive weights among all candidate 5-jokes).

VI. Conclusion

We propose a novel approach to connect two seemingly different areas: *network data analysis* and *compressive sensing*. By formulating problems in the framework of *Randon basis pursuit in homogeneous spaces*, we cast the network clique detection problem into a compressed sensing problem. Such a novel formulation allows us to explore and analyze network data guided by more rigorous theories. Instead of providing just another heuristic method, we aim at contributing at the fonudational level to network data analysis. We hope that our work could build a bridge connecting the research communities of network modeling and compressed sensing, so that research results and tools from one area could be ported to another to create more exciting results.

Acknowledgments

The authors also thank Z. Ma, M. Peng, M. Saunders, and Y. Ye for very helpful discussions and comments. H. Liu is thankful for a faculty supporting package from Johns Hopkins University.

This work was supported by ARO Grants W911NF-10-1-0037 and W911NF -07-2-0027, by NSF Grant CCF 1011228, by the Google Corporation, by the National Basic Research Program of China (973 Program 2011cb825501), by NSFC (61071157, 61370004, 11326038), by Microsoft Research Asia, by a professorship in the Hundred Talents Program at Peking University, by NSF Grants III-1116730, NSF III-1332109, and NSF IIS-1408910, NIH R01MH102339, NIH R01GM083084, R01HG06841, and by FDA HHSF223201000072C. This paper was previously presented at the 15th International Conference on Artificial Intelligence and Statistics (AISTATS), April 21–23, 2012, La Palma, Canary Islands, Spain, April. Recommended by Associate Editor F. Dabbene.

References

1. Abello J, Resende M, Sudarsky S. Massive quasi-clique detection. Lecture Notes in Computer Science. 2002; 2286:598–612.
2. Airoldi EM, Blei DM, Fienberg SE, Xing EP. Mixed membership stochastic blockmodels. J Mach Learn Res. 2008; 9:1981–2014. [PubMed: 21701698]
3. Alon N, Krivelevich M, Sudakov B. Finding a large hidden clique in a random graph. Proc 9th Annu ACM-SIAM Symp Discrete Algorithms, Society for Industrial and Applied Mathematics. 1998
4. Barabasi AL, Albert R. Emergence of scaling in random networks. Science. 1999; 286:509–512. [PubMed: 10521342]
5. Bickel PJ, Chen A. A nonparametric view of network models and newmanÐgirvan and other modularities. Proc Nat Acad Sci USA. 2009; 106:21068–21073. [PubMed: 19934050]
6. Bron C, Kerbosch J. Algorithm 457: Finding all cliques of an undirected graph. Commun ACM. 1973; 16:575–577.
7. Cai J, Osher S, Shen Z. Linearized bregman iterations for compressed sensing. Mathemat Comput. 2009; 78(267):1515–1536.
8. Candés EJ. The restricted isometry property and its implications for compressed sensing. Comptes Rendus de l'Acadéemie des Sciences, Paris, Série I. 2008; 346:589–592.
9. Candés EJ, Tao T. Decoding by linear programming. IEEE Trans Inform Theory. 2005; 51:4203–4215.
10. Candés EJ, Tao T. The dantzig selector: Statistical estimation when p is much larger than n. Ann Stat. 2007; 35(6):2313–2351.
11. Chen S, Donoho DL, Saunders MA. Atomic decomposition by basis pursuit. SIAM J Scientific Comput. 1999; 20:33–61.
12. Chen W, Liu Z, Sun X, Wang Y. A game-theoretic framework to identify overlapping communities in social networks. Data Mining and Knowl Discovery J, Special Issue on ECML PKDD. 2010; 21:224–240.
13. Dantzig G, Wolfe P. Decomposition principle for linear programs. Oper Res. 1960; 8:101–111.
14. Dekel, OGGY.; Peres, Y. Workshop on Analytic Algorithmics and Combinatorics. San Francisco, CA: 2011. Finding hidden cliques in linear time with high probability.
15. Deshpande Y, Montanari A. Finding hidden cliques of size $\sqrt{N/e}$ in nearly linear time. 2013 arXiv: 1304.7047.
16. Diaconis, P. Group Representations in Probability and Statistics. Institute of Mathematical Statistics; 1988.
17. Duijn MAJV, Snijders TAB, Zijlstra BJH. p_2: A random effects model with covariates for directed graphs. Statistica Neerlandica. 2004; 59:234–254.
18. Erdös P, Rényi A. On random graphs, i. Publicationes Mathematicae. 1959; 6:290–297.
19. Erdös P, Rényi A. On the evolution of random graphs. Pub Mathemat Inst Hungrian Acad, of Sci. 1960; 5:17–61.

20. Fuchs JJ. On sparse representations in arbitrary redundant bases. IEEE Trans Inform Theory. 2004; 50:1341–1344.
21. Gerhards L, Lindenberg W. Clique detection for nondirected graphs: Two new algorithms. Computing. 1979; 21:295–322.
22. Girvan M, Newman MEJ. Community structure in social and biological networks. Proc Nat Acad Sci USA. 2002; 99:7821–7826. [PubMed: 12060727]
23. Goldberg K, Roeder T, Gupta D, Perkins C. Eigentaste: A constant time collaborative filtering algorithm. Inform Retrieval. 2001; 4(2):133–151.
24. Goldenberg A, Zheng AX, Fienberg SE, Airoldi EM. A survey of statistical network models. Found and Trends in Mach Learn. 2010; 2
25. Grimmett, G.; Welsh, DJA. Disorder in Physical Systems: A Volume in Honour of John M Hammersley. New York: Oxford University Press; 1990.
26. Hoff PD, Raftery AE, Handcock MS. Latent space approaches to social network analysis. J American Statist Assoc. 2001; 97:1090–1098.
27. Holland PW, Leinhardt S. An exponential family of probability distributions for directed graphs. J American Statist Assoc. 1981; 76:33–50.
28. Jagabathula S, Shah D. Inferring rankings under constrained sensing. Neural Inform Process Syst (NIPS). 2008
29. Kleinberg JM, Kumar R, Raghavan P, Rajagopalan S, Tomkins A. The web as a graph: Measurements, models, methods. Proc Int Computing and Combinatorics Conf. 1999
30. Knuth, DE. The Stanford GraphBase: A Platform for Combinatorial Computing. Readin, MA: Addison-Wesley; 1993.
31. Kovács I, Palotai R, Szalay M, Csermely P. Community landscapes: A novel, integrative approach for the determination of overlapping network modules. PLoS ONE. 2010; 5:e12528. [PubMed: 20824084]
32. Kumar R, Raghavan P, Rajagopalan S, Sivakumar D, Tomkinsm A, Upfal E. Stochastic models for the Web graph. Proc Symp Foundations of Computer Science. 2000
33. Lancichinetti A, Fortunato S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Phys Rev E. 2009; 80(1):16118.
34. Leskovec J, Lang K, Mahoney M. Empirical comparison of algorithms for network community detection. ACM WWW Int Conf World Wide Web (WWW). 2010
35. Lorrain F, White H. Structural equivalence of individuals in social networks. Mathemat Sociol. 1971; 1:49–80.
36. Lueker GS. Maximization problems on graphs with edge weights chosen from a normal distribution. ACM Symp Theory of Comput. 1978
37. Mitchell JE. Polynomial interior point cutting plane methods. Optimiz Methods and Softw. 2003; 18:2003.
38. Newman MEJ. Modularity and community structure in networks. Proc Nat Acad Sci. 2006; 103(23):8577–8582. [PubMed: 16723398]
39. Onnela JP, Saramäki J, Kertész J, Kaski K. Intensity and coherence of motifs in weighted complex networks. Phys Rev E. 2005; 71
40. Palla G, Derényi I, Farkas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. Nature. 2005; 435(7043):814. [PubMed: 15944704]
41. Sarkar P, Moore A. Dynamic social network analysis using latent space models. SIGKDD Explorations: Special Edition on Link Mining. 2005
42. Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell. 2000; 22
43. Snijders TAB. Models for longitudinal network data. Models and Methods in Soc Network Anal. 2005
44. Tibshirani R. Regression shrinkage and selection via the lasso. J Roy Statist Soc B. 1996; 58(1): 267–288.
45. Tropp J. Just relax: Convex programming methods for identifying sparse signals in noise. IEEE Trans Inform Theory. 2006; 52:1030–1051.

46. Tsaig Y, Donoho DL. Compressed sensing. IEEE Trans Inform Theory. 2006; 52:1289–1306.
47. Wasserman S, Anderson C. Stochastic a posterior blockmodels: Construction and assessment. Social Netw. 1987; 9:1–36.
48. Wasserman S, Pattison P. Logit models and logistic regressions for social networks: I. an introduction to markov graphs and *p**. Psychometrika. 1996; 61:401–425.
49. Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. Nature. 1998; 393:409–410. [PubMed: 9623993]
50. Ye, Y. Interior Point Algorithms: Theory and Analysis. New York: Wiley; 1997.
51. Yuan M, Lin Y. On the nonnegative garrote estimator. J Roy Statist Soc B. 2007; 69(2):143–161.
52. Zhao P, Yu B. On model selection consistency of lasso. J Mach Learn Res. 2006; 7:2541–2563.

Biographies

Xiaoye Jiang (M′11) received the Ph.D. degree from Stanford University, Stanford, CA, USA, in 2012 from the Institute for Computational and Mathematical Engineering, under the supervision of Dr. Guibas.

His research focuses on machine learning and pattern recognition with applications in ranking, multi-object tracking, and social network studies.

Yuan Yao (M′13) received the B.S.E and M.S.E. degrees in control engineering from Harbin Institute of Technology, Harbin, China, in 1996 and 1998, respectively, the M.Phil. degree in mathematics from City University of Hong Kong in 2002, and the Ph.D. degree in mathematics (with Prof. Smale) from the University of California at Berkeley in 2006.

He was with Stanford University, Stanford, CA, USA, then joined the School of Mathematical Sciences, Peking University, Beijing, China, in 2009 as a Professor of Statistics in the Hundred Talents Program. His current research interests include topological and geometric methods for high dimensional data analysis and statistical machine learning, with applications in computational biology, computer vision, and information retrieval.

Dr. Yao is a member of ACM.

Han Liu (M′11) received a joint Ph.D. degree in machine learning and statistics from the Carnegie Mellon University, Pittsburgh, PA, USA, in 2011.

He is currently an Assistant Professor of Statistical Machine Learning in the Department of Operations Research and Financial Engineering at Princeton University, Princeton, NJ, USA. He is also an adjunct Professor in the Department of Biostatistics and Department of Computer Science at Johns Hopkins University. He built and is serving as the principal investigator of the Statistical Machine Learning (SMiLe) lab at Princeton University. His research interests include high dimensional semiparametric inference, statistical optimization, and Big Data inferential analysis.

Leonidas Guibas (F′12) received the Ph.D. degree from Stanford University, Stanford, CA, USA, in 1976 (under the supervision of Prof. Knuth).

He has worked for Xerox PARC, MIT, and DEC/SRC. He has been at Stanford University since 1984 as Professor of Computer Science. He has produced several Ph.D. students who are well-known in computational geometry, such as J. Hershberger, J. Snoeyink, and J. Stolfi, or in computer graphics, such as D. Salesin, E. Veach, and N. Mitra. At Stanford University, he has developed new courses in algorithms and data structures, geometric modeling, geometric algorithms, and sensor networks.

Dr. Guibas is an ACM Fellow as well as a winner of the ACM/AAAI Allen Newell award.

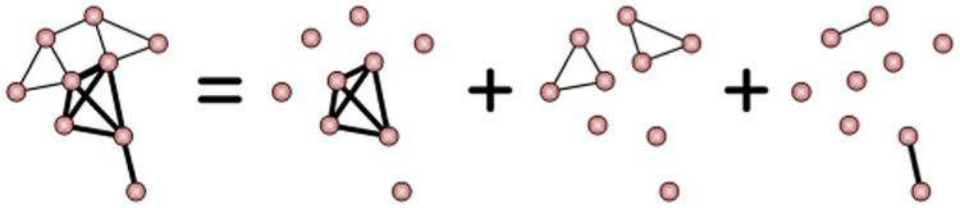

Fig. 1.
Illustrative example of the main idea: the sparse representation of a network with tetrahedra, triangles and edges.

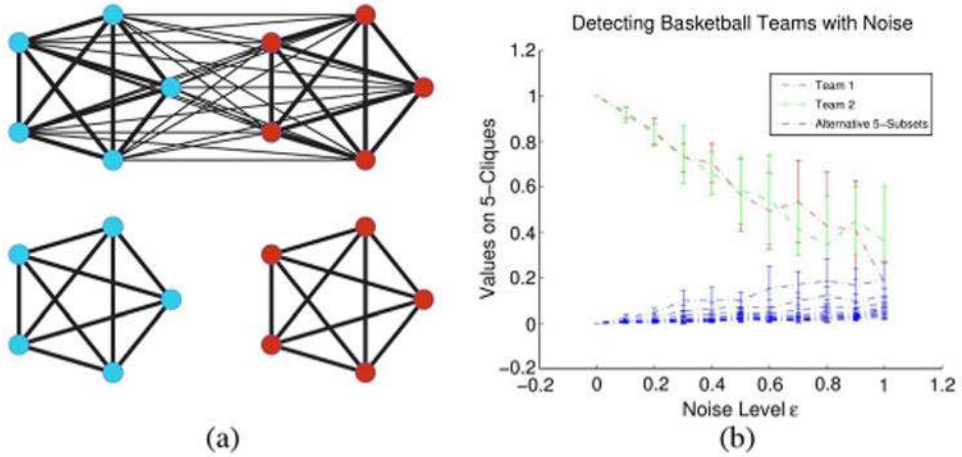

Fig. 2.

Detecting Basketball Teams with Noise. (a) Two teams in a virtual Basketball Game, with intra-team interaction 1 and cross-team interaction noise no more than $\in$, i.e., the noise term $z = b - Ax_0$ in $\mathcal{P}_{1,\delta}$ satisfies $\|z\|_\infty \leq \in$. (b) Under a large noise level $\in < 0.9$, the two teams are identifiable. For each noise level, we run 100 simulations repeatedly, whose errorbar plot of weights on cliques are shown.

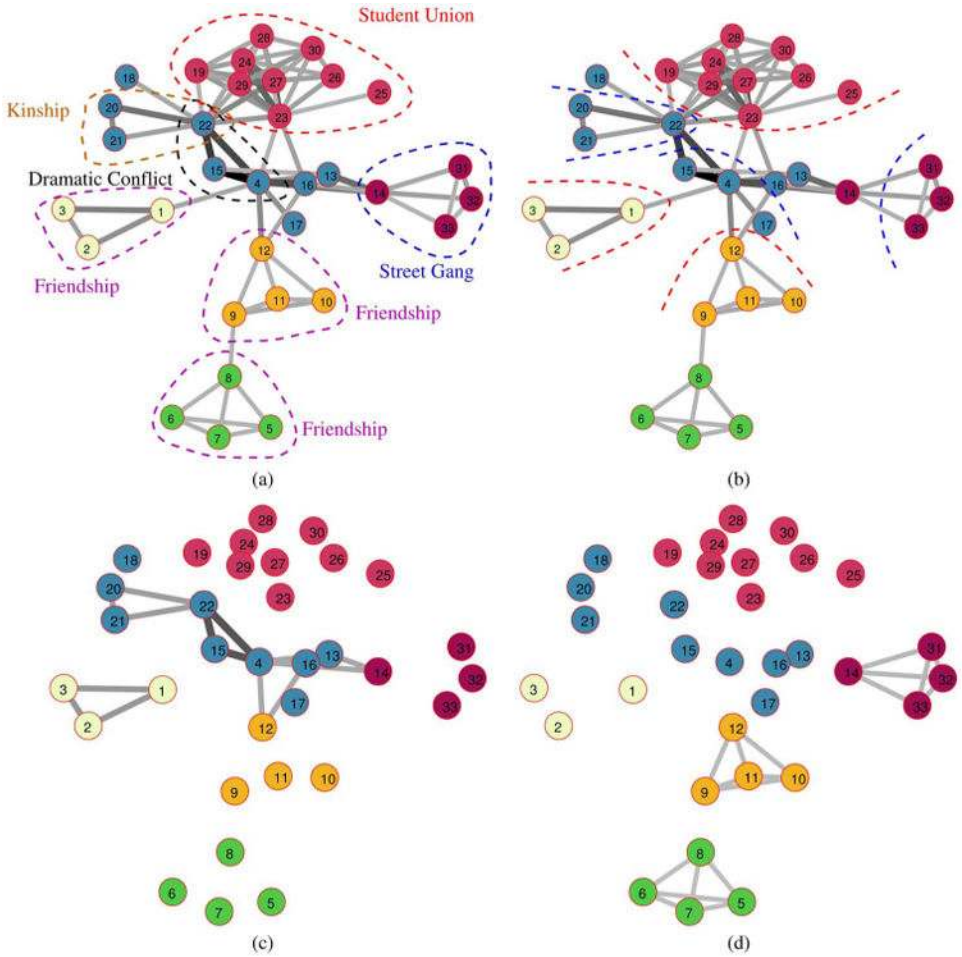

Fig. 3.

Decomposition of *Les Misérables* social network. (a) Social network of characters in *Les Misérables*. (b) Spectral clustering result. (c) The identified 3-cliques. (d) The identified 4-cliques.

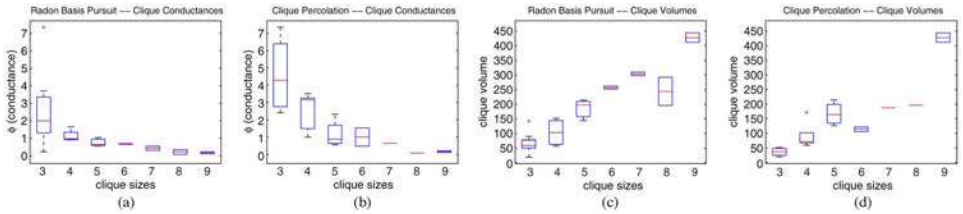

Fig. 4.

Les Misérables social network: Box plot of clique conductances and volumes for clique percolation method and our approach. Cliques identified by our approach have smaller conductances and larger volumes.

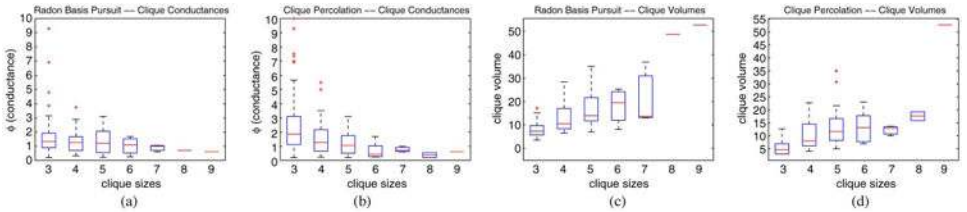

Fig. 5.

Coauthorship Network: Box plot of clique conductances and volumes for clique percolation method and our approach. Cliques identified by our approach have smaller conductances and larger volumes.

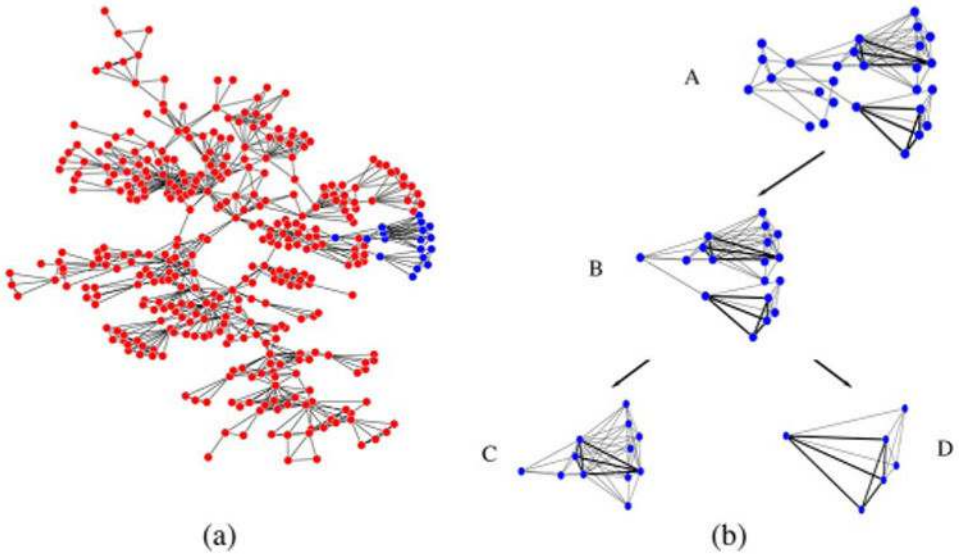

Fig. 6.

(a) Coauthorships in Network Science, only a part of the network is shown. (b) Important cliques identified within clusters behave in a persistent way. Clustering node B is exactly the blue part in (a).

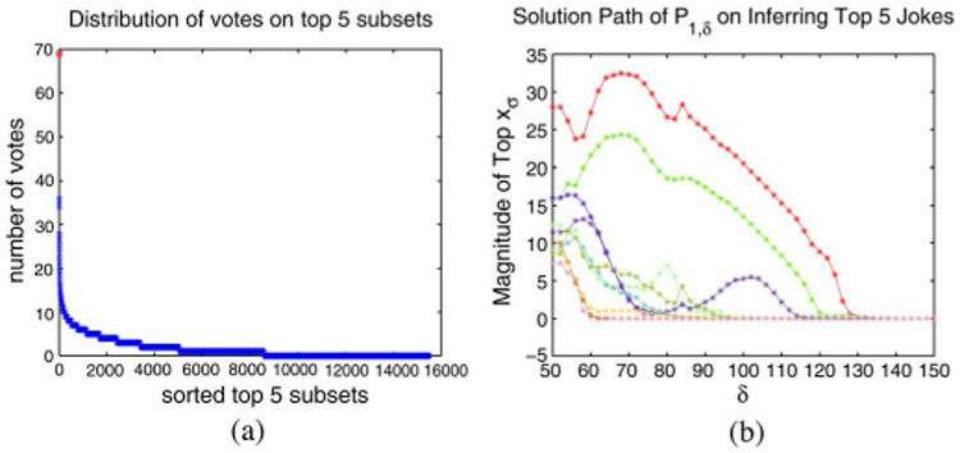

Fig. 7.

(a) There is a significant 5-joke tuple (in red) whose joke ID is {27, 29, 35, 36, 50}. (b) We try to infer this top 5-joke tuple from the scores of all 3-joke tuples. The regularization path is plotted. The red path, which consistently achieves the highest scores, coincides with the top 5-joke tuple.

Table I

Social Networks of Les Misérables

Cliques	Names of Characters	Relationships	Perco.	Radon	Spectral
{1,2,3}	{Myriel, Mile Baptistine, Mme Magloire}	Friendship	N	Y	Y
{4,13,14}	{Valjean, Mme Thenardier, Thenardier}	Dramatic Conflicts	N	Y	N
{4,15,22}	{Valjean, Cosette, Marius}	Dramatic Conflicts	N	Y	N
{20,21,22}	{Gillenormand, Mile Gillenormand, Marius}	Kinship	N	Y	Y
{5,6,7,8}	{Tholomyes, Listolier, Fameuil, Blacheville}	Friendship	Y	Y	Y
{9,10,11,12}	{Favourite, Dahlia, Zephine, Fantine}	Friendship	Y	Y	Y
{14,31,32,33}	{Thenardier, Gueulemer, Babet, Claquesous}	Street Gang	N	Y	N

IEEE Trans Automat Contr. Author manuscript; available in PMC 2015 January 21.

Table II

Running Time of Algorithms

Iterative	Stagewise	Our Algorithm	Spectral	Percolation
5.12 (s)	76.05 (s)	35.52 (s)	0.13 (s)	0.18 (s)