ANOMALOUS SUBGRAPH DETECTION VIA
SPARSE PRINCIPAL COMPONENT ANALYSIS

Navraj Singh1,2, Benjamin A. Miller2, Nadya T. Bliss2, and Patrick J. Wolfe3

1Dept. of Electrical Engineering, Stanford University, Stanford, CA 94305
navraj@stanford.edu

2MIT Lincoln Laboratory, Lexington, MA 02420
{bamiller,nt}@ll.mit.edu

3Statistics and Information Sciences Laboratory, Harvard University, Cambridge, MA 02138
wolfe@stat.harvard.edu

ABSTRACT

Network datasets have become ubiquitous in many fields of study in recent years. In this paper we investigate a problem with applicability to a wide variety of domains - detecting small, anomalous subgraphs in a background graph. We characterize the anomaly in a subgraph via the well-known notion of network modularity, and we show that the optimization problem formulation resulting from our setup is very similar to a recently introduced technique in statistics called Sparse Principal Component Analysis (Sparse PCA), which is an extension of the classical PCA algorithm. The exact version of our problem formulation is a hard combinatorial optimization problem, so we consider a recently introduced semidefinite programming relaxation of the Sparse PCA problem. We show via results on simulated data that the technique is very promising.

Index Terms— Anomaly detection, community detection, graph analysis, semidefinite programming, sparse principal component analysis

1. INTRODUCTION

Network datasets have become increasingly prevalent in a variety of fields ranging from the social sciences to biology. As a result, network analysis has received much attention in recent years. In this paper we consider a basic problem that is closely related to community detection in networks: detecting one or more anomalous subgraphs embedded in a given 'background' graph. More precisely, we wish to detect the presence of and locate subgraphs that deviate significantly from their expected connectivity. This resembles the much-studied problem of community detection in graphs, where the goal is usually to partition the set of nodes into groups/modules/communities that have higher internal connectivity than external connectivity. We here consider the problem of detecting communities in a noisy background graph where not all nodes necessarily belong to a community. This setting is analogous to the classical signal processing setting of detecting signals in noise. In our case, however, the data (graphs) are non-Euclidean, and therefore traditional signal detection algorithms do not apply. The reader is referred to [1, 2] for a general overview of our problem setting and the challenges it presents (the problem setting is again briefly reviewed in Section 3). Prior work relevant to our problem setting generally falls under the category of graph anomaly detection, and is again discussed in [1, 2].

Our primary contribution is the revelation of a strong connection between the subgraph detection problem and a recently developed statistical technique called Sparse Principal Component Analysis (SPCA), which is an extension of PCA, a common tool in high-dimensional data analysis. In our problem setup, we use Newman's notion of network modularity [3] to characterize the anomalies that we wish to locate, and at a high level, our procedure can be summarized as the application of SPCA to a graph's modularity matrix (as opposed to a covariance matrix in the case of traditional data analysis). Various algorithms for SPCA have been introduced recently (we review SPCA in Section 5). For our results, we specifically use a relaxation based on semidefinite programming introduced in [4]. We show via simulations that the method proposed is able to detect very weak anomalous subgraph embeddings. It seems SPCA hasn’t been studied in a network analysis setting before, and it is our hope that the connection presented here will facilitate the direct application of recent developments (especially in algorithms) in the SPCA literature.

2. NOTATION

Before proceeding, here we specify the notation that will be used throughout the paper. We denote by \( \mathcal{G} = (\mathcal{V}_G, \mathcal{E}_G) \) an unweighted, undirected graph with vertex set \( \mathcal{V}_G \) and edge set \( \mathcal{E}_G \). Throughout the paper, all instances of graphs are unweighted and undirected for simplicity. By \( S(\mathcal{G}) \) we denote the set of all possible subgraphs of a given graph \( \mathcal{G} \). For a given \( S \in S(\mathcal{G}) \), we denote its vertex set by \( \mathcal{V}_S \) and its edge set by \( \mathcal{E}_S \). Further, for \( k \leq n \), let \( S^k(\mathcal{G}) = \{ S \in S(\mathcal{G}) : |\mathcal{V}_S| \leq k \} \). So, \( S^k(\mathcal{G}) \) is the set of all subgraphs of \( \mathcal{G} \) with size less than or equal to \( k \) (note that \( S^n(\mathcal{G}) = S(\mathcal{G}) \) by definition). For a generic graph \( \mathcal{G} \) with \( |\mathcal{V}_G| = n \), we assume \( \mathcal{V}_{G_0} = \{1, \ldots, n\} \), and we denote by \( k_i \) its degree vector (so, \( k \in \mathbb{R}^n \), with \( k_i \) being the degree of vertex \( i \)). Following the convention in some of the random graph literature, for a subgraph \( S \) of \( \mathcal{G} \), we denote...
its total degree (volume) by $\text{vol}(S)$. That is, $\text{vol}(S) = \sum_{i \in V_S} k_i$. Finally, $1$ denotes the vector of all $1$’s, $|A|$ denotes the matrix with entries equal to the absolute value of the corresponding entries in some matrix $A$, and $S^n$ denotes the set of all symmetric matrices in $\mathbb{R}^{n \times n}$. We next define the anomalous subgraph detection problem.

3. ANOMALOUS SUBGRAPH DETECTION PROBLEM

As described in [1, 2], we can view the anomalous subgraph detection problem as one of detecting a signal in noise, or equivalently as a binary hypothesis test. Let $G_B = (V_B, E_B)$ denote a background graph with no anomalies, generated via some underlying random process (i.e., it is an instance of some random graph ensemble, generated using a random graph generator, examples of which are outlined in [5]). Let $\{G_B\}$ denote the ensemble of graphs under the background random process model. We can view $G_B$ as “noise”. Further, let $G_S = (V_S, E_S)$ denote a smaller, anomalous graph, with $V_S \subset V_B$. By ‘anomalous’, we mean that $G_S$ is unlikely to occur under the random process that generated $G_B$ (i.e. $G_S \notin \{G_B\}$ with high probability). Given an observed graph $G$, under the null hypothesis $H_0$, $G$ is “noise”, i.e., $G \notin \{G_B\}$, and under the alternate hypothesis $H_1$, $G \notin \{G_B\}$ (i.e., $G = G_B \cup G_S$, where $G_B \in \{G_B\}$, $G_S \notin \{G_B\}$, and $G_B \cup G_S \equiv (V_B, E_B \cup E_S)$).

3.1. Modularity

As mentioned above, we view a subgraph as anomalous if it is unlikely to arise under an assumed random process model (a null model). In particular, it is anomalous if its total weight (number of edges in the subgraph) deviates substantially from its expected total weight under the assumed null model. Newman’s notion of modularity [3] captures just this notion. The specific null model assumed in this case is the so-called configuration model in physics, where the graph is generated randomly according to a fixed (or in some cases, expected) degree sequence (i.e. it is drawn uniformly at random from the set of all graphs that satisfy the given degree sequence). Computing the expected weight of a subgraph in such a random graph is straightforward. For an unweighted, undirected graph $G$, let $A$ denote its adjacency matrix (i.e., $A_{ij} = 1$ if there is an edge between vertices $i$ and $j$, and 0 otherwise). Then, the expected number of edges between vertices $i$ and $j$, under the configuration model, is $\frac{k_i k_j}{2m}$, where $m = |E_G|$. Let $B_{ij}$ denote the difference between the actual and expected number of edges between vertices $i$ and $j$ (so, $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$). The matrix $B$ with elements $B_{ij}$ is then the so-called modularity matrix. Note also that for a graph with $n$ vertices, $B \in S^n$, and thus has real eigenvalues. Finally, it is worth noting the similarity between the modularity matrix and a covariance matrix (this is discussed further in Section 5 in the context of PCA and SPCA).

4. OPTIMIZATION PROBLEM FORMULATION

As mentioned above, we view a subgraph as anomalous if it has more (or fewer) edges than expected under the assumed null model. Throughout the paper, we will only look at the case where the anomaly is denser than the expectation (as opposed to sparser, in which case the methods presented here easily carry over). Under this notion of an anomaly, a given subgraph $S$ is anomalous if $B(S)$ is large, where $B(S) = \sum_{i,j \in V_S} B_{ij}$. Suppose for the moment we know the size of the anomaly we wish to detect. To find the most anomalous subgraph of size at most $k$, we need to solve the optimization problem

$$\text{maximize } \sum_{g \in S^k} B(S),$$

(1)

A more mathematically tractable, but equivalent, way to write this would be as the optimization problem

$$\text{maximize } x^T B x \tag{2}$$

subject to $x \in \{0,1\}^n$, $\|x\|_0 \leq k$, with variable $x \in \mathbb{R}^n$. Here, $\|x\|_0$ (commonly called the $L_0$ “norm”) denotes the cardinality of $x$ (i.e., $\|x\|_0 = |\{i : x_i \neq 0\}|$). This is an NP-hard combinatorial problem. Consider instead the relaxation

$$\text{maximize } x^T B x \tag{3}$$

subject to $\|x\|_2 = 1$, $\|x\|_1 \leq t$, with variable $x \in \mathbb{R}^n$. Here we have replaced the constraint $x \in \{0,1\}^n$ with $\|x\|_2 = 1$ and replaced the cardinality constraint on $x$ with a constraint on its $L_1$ norm (a convex relaxation). A particular value of $t$ chosen by the user would correspond to a rough prior knowledge of an anomaly size instead of an exact value. Similar relaxations have recently been studied in various problem settings in the statistics and signal processing literature, in problems where some form of sparsity is assumed in a quantity that is to be estimated. We consider the equivalent penalized maximization problem

$$\text{maximize } x^T B x - \lambda \|x\|_1 \tag{4}$$

subject to $\|x\|_2 = 1$, with variable $x \in \mathbb{R}^n$. Here $\lambda > 0$ is a tunable regularization parameter that needs to be chosen by the user (a particular value of $\lambda$ here would correspond to a particular value of $t$ in (3)). Note, again, that in this final formulation we do not have an explicit constraint on the cardinality of $x$. A particular choice of $\lambda$ would express a rough prior knowledge of the size of an anomalous subgraph we are trying to detect. Indeed, the second term in the objective and a particular value of $\lambda$ can be thought of as imposing a prior distribution on $x$ in a Bayesian context. The $i^{th}$ component of the solution to (4) should roughly indicate the degree to which vertex $i$ belongs to the most anomalous subgraph. If there is a significantly anomalous subgraph present, then we should be able to get an idea of its size and location by analyzing the sparsity (after a thresholding of the components) of the solution to (4). On the other hand, a noisy solution showing a lack of any sparsity should indicate absence of an anomaly.

Note that $B$ is in general indefinite, so the problem in (4) is still a hard problem to solve as it involves maximization of an objective function that is neither convex nor concave, over a quadratic equality constraint. We consider a semidefinite programming relaxation of (4) in the next section to make the problem tractable without sacrificing a lot in accuracy.

5. CONNECTION WITH SPARSE PCA

The reader will recognize the similarity of (4) to the problem of finding the first principal component of a set of data points. Given $N$ data points $\{x^{(i)} \in \mathbb{R}^n\}_{i=1}^N$, the eigendecomposition of the sample covariance matrix $\Sigma \in S^n$ (estimated as $\Sigma = \frac{1}{N-1} XX^T$, where $X \in \mathbb{R}^{n \times N}$ is the data matrix), yields the principal components
used in the well-known PCA algorithm. The first principal component, which gives the direction that maximizes sample variance, is simply the leading eigenvector of $\Sigma$, obtained either via an eigendecomposition or by solving the optimization problem

$$\begin{align*}
\text{maximize} & \quad u^T \Sigma u \\
\text{subject to} & \quad \|u\|_2 = 1,
\end{align*}$$

(5)

with variable $u \in \mathbb{R}^n$. Note that if we remove the second term in the objective of (4) and replace $B$ with $\Sigma$, we are left with (5).

Sparse PCA (SPCA) is a recent modified version of PCA, aimed at obtaining modified principal components that are easier to interpret. One issue with standard PCA is that the projection of the data onto the lower dimensional space yields new derived variables which are linear combinations of all the original variables, which makes the new variables difficult to interpret. SPCA instead attempts to find, as the name suggests, ‘sparse principal components’, i.e., cardinality constrained unit vectors pointing in the directions that capture maximum sample variance. The result is that after dimensionality reduction, the new variables are linear combinations of only a few variables. One of the main applications of SPCA therefore has been in feature selection (variable subset selection) problems, especially in an unsupervised setting.

The optimization problem in (3) (or, equivalently, in (4)) is exactly the SPCA problem. Before considering the direct application of SPCA (or even standard PCA) to our setting, it must noted that in our case the role of $\Sigma$ is being played by $B$ (here we will denote a generic modularity matrix by $B$ and a covariance matrix by $\Sigma$). It is useful to consider $B$ as capturing some notion of covariance. SPCA on $B$ can be thought of as performing ‘vertex subset selection’ instead of ‘variable subset selection’ (i.e., we may view vertices as variables). More precisely, the goal of finding a small subset of the graph vertices that maximize modularity can be viewed as analogous to the goal (in SPCA) of finding a small subset of variables that maximize sample variance. From now on, we use $B$ wherever a covariance appears in the PCA and SPCA algorithms.

After its first LASSO-inspired introduction in [6], SPCA has received considerable attention in the statistics literature, and several formulations and approximation algorithms of varying complexity have been proposed (see, for example, [7, 8, 4, 9]). However, to the best of our knowledge, it has not yet been studied in the context of graph-valued data as proposed here.

Given $B$, we use the problem in (4) as our starting point in solving for the first sparse principal component. This form is in fact the SCOTLASS form proposed in [6]. We then use the lifting procedure for semidefinite programming to relax (4) to the semidefinite program

$$\begin{align*}
\text{maximize} & \quad \text{Tr}(BX) - \lambda \left(1^T|X|1\right) \\
\text{subject to} & \quad \text{Tr}(X) = 1 \\
& \quad X \succeq 0,
\end{align*}$$

(6)

with variable $X \in \mathbb{S}^n$. The reader is referred to [4] for a derivation of the relaxation. The approximate solution to (4) is then given by the leading eigenvector of the solution of (6). If we add the additional (hard) constraint that $\text{Rank}(X) = 1$, then the formulations in (6) and (4) would be equivalent. The formulation in (6) has sufficiently nice structure that can be exploited to obtain a solution in polynomial time $O(n^4 \sqrt{\log(n)}/\epsilon)$, where $\epsilon$ determines solution accuracy. We use the DSPCA toolbox [10] provided by the authors of [4] for the results in this paper. The reader is also referred to [11] for some previous related work on mathematical programming formulations for the related problem of modularity maximization. See [12] for the use of semidefinite programming in a related dense subgraph detection problem (though the problem setting and resulting SDP formulation are quite different from ours).

6. EXPERIMENTAL SETUP AND RESULTS

In the simulated experiments, we consider the R-MAT generator [5] for the background random graph model (null model). This model uses an edge probability matrix generated by a Kronecker product on a starting $2 \times 2$ probability matrix. As described in [5], an appropriate selection of the model parameters can result in graphs that have properties similar to those observed in many real world graphs (e.g., power law degree distribution, and intrinsic hierarchical community structure). The performance of our method on simpler random graph models, such as the classic Erdős-Rényi (ER) model, is better than on R-MAT models, so the ER model results are omitted.

To generate an alternate model, once we generate a background graph $G_B$ via the R-MAT generator, we generate an anomalous graph $G_S$ of specified size, and embed it into $G_B$ by randomly selecting $V_S$ from a candidate subset of $V_B$. The embedding is done by simply updating the edge set to be $E_B \cup E_S$. The anomalous embeddings are varied in terms of their internal density and external connectivity (external connectivity is controlled by ensuring that the embedding vertex set $V_B$ is chosen from among only those vertices in $V_B$ that have degree at most some predefined value). Intuitively, higher density and low external connectivity should make an embedding easier to detect. This setup is similar to that used in [2].

Fig. 1(a) shows the performance of SPCA on a sample test case. Here, $|V_B| = 1024$, $|V_S| = 7$, R-MAT background is generated with seed probability matrix $[0.5, 0.125; 0.125, 0.25]$ and average degree of about 12. The embedded subgraph is 90% dense and the set $V_S$ is randomly selected from among the vertices in $V_B$ with degree at most 6 (as mentioned above, this controls the external connectivity of the embedding). The plot shows the sparse principal component returned by SPCA. Specifically, this is the leading eigenvector of the solution of the SDP relaxation described above, solved via the DSPCA toolbox. Here, $\lambda = 0.4$ was used. The plot clearly shows that the solution via DSPCA is sparse, with high values at the embedding vertex indices (marked by red crosses) and almost zero at the background indices. With an appropriately chosen value for $\lambda$, the result returned by DSPCA would be sparser (as in the plot shown here) for a graph with an embedding than for one without (indeed, for this specific example, if we omit the embedding, the DSPCA result is a noisy vector; we omit the plot).

We may therefore use the $L_1$ norm of the DSPCA result as a test statistic for detection. That is, if the $L_1$ norm is below some threshold, we declare the presence of an embedding (and in that case we may threshold the component values of the Sparse PC to identify the embedding vertices, expecting the values at those locations to be high), and declare no presence otherwise. This is a typical set of parameters where the eigenvector $L_1$ norm method of [2] and the chi-squared test approach of [1] both fail to detect the embedding.

Using the $L_1$ norm of the DSPCA result as a test statistic, and comparing it against a threshold, allows us to evaluate performance via receiver operating characteristic (ROC) curves. All ROC curves shown here were generated with a 1000-sample Monte Carlo simulation. Fig. 1(b) shows the performance as $\lambda$ is varied for detecting a 6-vertex embedding (90% dense, with external degree restricted to about 10) in a 512-vertex R-MAT graph. The performance starts out poor with a low value of $\lambda$, becomes almost perfect for $\lambda = 0.42$, and then decreases again for higher values of $\lambda$, as expected. In fact, $\lambda = 0.4$ seems to work well for most problem parameters. Detailed
analysis of behavior vs. $\lambda$ remains for future work.

Figs. 2(a) and 2(b) show ROC curves for the $L_1$ method of [2] and DSPCA, respectively, for 6-vertex embeddings of varying densities, with external degree at most 8, in 512-vertex R-MAT backgrounds (generated with same parameters as the single example presented earlier). For DSPCA, we get perfect performance for embedding densities higher than a particular value, indicating a phase-transition type phenomenon that remains to be studied in more depth. The $L_1$ method here does not perform much better than chance. Figs. 2(c) and 2(d) show similar results for 7-vertex embeddings of varying densities (with no external degree restriction) in 1024-vertex R-MAT backgrounds.

We note that all ROC curves shown here are for detection only. We omit the ROC curves for localization/identification of the subgraphs (which requires a different method, as we are measuring the performance w.r.t. classification of the individual vertices). The results are similar for localization (e.g., perfect detection performance usually also results in perfect localization performance).

In this paper we have demonstrated the application of Sparse PCA to anomalous subgraph detection and localization, a widely-applicable problem in graph analysis, and the results show that the technique is very promising. Future directions include tests on real network data, more analysis of sensitivity to $\lambda$, development of a deflation technique for finding multiple anomalous subgraphs/communities, development of faster approximation algorithms (e.g. the generalized power method of [8]), and theoretical studies to determine possible performance limits and guarantees (e.g., in relation to the recently studied resolution-limit problem in community detection [13]).

8. REFERENCES