PLANTED CLIQUE DETECTION BELOW THE NOISE FLOOR USING LOW-RANK SPARSE PCA

Alexis B. Cook[1] *and Benjamin A. Miller*[2]

[1]Department of Applied Mathematics, Brown University, Providence, RI 02906
alexis_cook@brown.edu

[2]MIT Lincoln Laboratory, Lexington, MA 02420
bamiller@ll.mit.edu

ABSTRACT

Detection of clusters and communities in graphs is useful in a wide range of applications. In this paper we investigate the problem of detecting a clique embedded in a random graph. Recent results have demonstrated a sharp detectability threshold for a simple algorithm based on principal component analysis (PCA). Sparse PCA of the graph's modularity matrix can successfully discover clique locations where PCA-based detection methods fail. In this paper, we demonstrate that applying sparse PCA to low-rank approximations of the modularity matrix is a viable solution to the planted clique problem that enables detection of small planted cliques in graphs where running the standard semidefinite program for sparse PCA is not possible.

***Index Terms*—** planted clique detection, graph analysis, community detection, semidefinite programming, sparse principal component analysis

1. INTRODUCTION

Real-world graphs are naturally inhomogeneous and exhibit nonuniform edge densities within local substructures. In this setting, it is often possible to break graphs into communities, or sets of vertices with a high number of within-group connections and fewer links between communities. The problem of community detection in networks has become increasingly prevalent in recent years and has important applications to fields such as computer science, biology, and sociology [1–3]. While community detection often considers partitioning a graph into multiple communities, a variant of the problem considers detection of a small subgraph with higher connectivity than the remainder of the graph [4], a special case of which is the planted clique problem [5–7].

A clique is a set of vertices such that every two vertices are connected by an edge. In the planted clique problem, one is given a graph containing a hidden clique, where each possible edge outside the clique occurs independently with some probability p. Detecting the location of this maximum-density embedding in a random background graph is a useful proxy for a variety of applications—such as computer network security or social network analysis—in which a subgraph with anomalous connectivity is to be detected.

This work is sponsored by the Assistant Secretary of Defense for Research & Engineering under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

We study approaches to solving the planted clique problem that are based on an analysis of the spectral properties of the graph's modularity matrix. Nadakuditi proved in [8] the presence of a phase transition between a regime in which the principal component of the modularity matrix can clearly identify the location of the clique vertices and a regime in which it cannot. Through understanding the breakdown point of this simple spectral method for clique detection in high dimensional settings, we hope to motivate new algorithms to detect cliques smaller than previously thought possible.

Sparse principal component analysis (SPCA) has recently been shown to enable detection of dense subgraphs that cannot be detected in the first principal component [7, 9]. The semidefinite programming formulation known as DSPCA has also been proposed as an approximation that nearly achieves the information-theoretic bound in [4], and as a complexity-theoretic bound for planted clique detection [10]. However, DSPCA has great computational burden. This is because several eigendecompositions are performed over the course of the procedure. For subgraphs that are relatively close to the detection threshold, however, the subgraph still manifests itself significantly in the largest eigenvectors. This suggests that these cliques would be detectable even if only a low-rank subspace of the original matrix were considered. In this paper, we investigate the use of DSPCA with a low-dimensional approximation of the modularity matrix. The running time of DSPCA is $O(n^4\sqrt{\log n}/\epsilon)$, so reducing the dimensionality of the problem can potentially improve the running time by multiple orders of magnitude.

The remainder of this paper is organized as follows. Section 2 outlines the problem model and defines notation. Section 3 briefly discusses recent relevant research in this area. In Section 4, we demonstrate analytically—using an approximation—that a large fraction of the magnitude of a vector indicating the planted clique will manifest itself in the eigenvectors of the graph's modularity matrix associated with the largest eigenvalues. Section 5 provides empirical results demonstrating improved performance using a small number of eigenvectors (rather than only one), and in Section 6 we summarize and outline possible future work.

2. DEFINITIONS AND NOTATION

In the standard (k, p, n) planted clique problem, one is given an Erdős-Rényi graph $G(n, p)$ with an embedded clique of size k, and the objective is to determine the hidden locations of the clique vertices. Formally, given the graph $G = (V, E)$, where V is the set of vertices and E is the set of edges (connections), with $|V| = n$, the desired outcome is the subset of vertices $V^* \subset V$, $|V^*| = k$ that

belong to the clique.

In our procedure, instead of working directly with the edge set E, we will analyze the adjacency matrix corresponding to our model. The adjacency matrix A of an undirected, unweighted graph is a useful representation of the graph's topology. Each row and column is associated with a vertex in V. After applying an arbitrary ordering of the vertices with integers from 1 to n, we will denote the ith vertex as v_i. Then A_{ij} is 1 if v_i and v_j are connected by an edge and is 0 otherwise. The model for the adjacency matrix of the planted clique problem is:

$$A_{ij} = \begin{cases} 1 & \text{if } v_i, v_j \in V^* \\ 1 & \text{with prob } p \text{ if } v_i \notin V^* \text{ or } v_j \notin V^* \\ 0 & \text{with prob } 1-p \text{ if } v_i \notin V^* \text{ or } v_j \notin V^*. \end{cases}$$

Consistent with the methodology developed by Newman [1], we will study the modularity matrix B of our observed graph, defined as:

$$B_{ij} = A_{ij} - p,$$

where we have subtracted the expected value of the adjacency matrix for the Erdős-Rényi model that does not contain the planted clique.

3. RECENT METHODS AND RESULTS

It has been shown previously that thresholding the principal eigenvector of B can yield the locations of the clique vertices. The unit-normalized principal eigenvector u_1 of the modularity matrix associated with an Erdős-Rényi graph without an embedded clique is asymptotically distributed as $\sqrt{n}u_1 \sim N(0, I)$ [11]. Using this fact, Nadakuditi establishes an algorithm for detecting vertices $\hat{V}^*$ in the planted clique [8]:

$$\hat{V}^* = \left\{ v_i : |\sqrt{n}u_{i1}| > F^{-1}_{N(0,1)}\left(1 - \frac{\alpha}{2}\right) \right\}, \tag{1}$$

where u_{i1} denotes the ith entry in u_1 and the false-alarm probability of identifying a non-clique vertex as part of the clique is α.

PCA-based clique detection has been shown to have a well-defined breakdown point in high dimensions. Nadakuditi elucidates this breakdown point in the following theorem.

Theorem 3.1 (Nadakuditi [8]) *Consider a (k, p, n) planted clique problem where the clique vertices are identified using (1) for a significance level α. Then, for fixed p, as $k, n \to \infty$ such that $k/\sqrt{n} \to \beta \in (0, \infty)$ we have*

$$\mathbb{P}(\textit{clique discovered}) \xrightarrow{a.s.} \begin{cases} 1 & \textit{if } \beta > \beta_{crit.} := \sqrt{\frac{p}{1-p}} \\ \alpha & \textit{otherwise.} \end{cases}$$

This theorem implies that, for sufficiently large graphs, the algorithm for PCA-based clique detection described in (1) performs poorly when the clique size $k \leq \sqrt{\frac{np}{1-p}}$. Fortunately, there is another spectral technique that has proven capable of getting past this detection threshold.

We will use an approach relying on sparse PCA to attempt to find a vector that is "close to" the principal component, but that contains exactly k nonzero entries, in the hope that the entries will correspond to the clique vertices. Formally, we would like to solve:

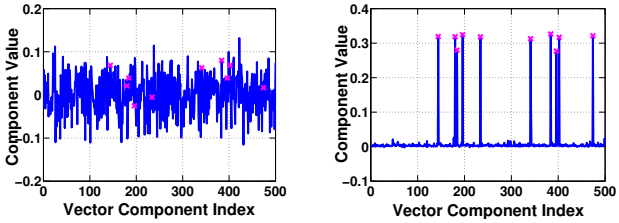
$$\begin{aligned} \hat{x} &= \arg\max_{\|x\|_2=1} x^T B x \\ &\text{subject to } \|x\|_0 = k, \end{aligned} \tag{2}$$

where $\|\cdot\|_0$ denotes the L_0 quasi-norm. We will apply a convex relaxation and use the lifting procedure for semidefinite programming (SDP), following the method of [12], to yield a semidefinite program:

$$\begin{aligned} \hat{X} &= \arg\max_{X \in S_n} \text{tr}(BX) - \rho \mathbf{1}^T |X| \mathbf{1} \\ &\text{subject to } \text{tr}(X) = 1, \end{aligned} \tag{3}$$

where $\text{tr}(\cdot)$ denotes the matrix trace, and S_n is the set of positive semidefinite matrices in $\mathbb{R}^{n \times n}$. Here, $\rho > 0$ is a tunable parameter that controls the sparsity of the solution. After applying DSPCA to a graph observation, we take the principal eigenvector of $\hat{X}$ and assign the vertices associated with the k largest entries in absolute value as belonging to the clique. We use the DSPCA toolbox provided by the authors of [12] for the results in this paper.

Fig. 1 compares the results returned by PCA and DSPCA for a sample test case. We generated an Erdős-Rényi graph $G(500, 0.2)$ and embedded a clique of size 10. The PCA-based technique fails here, as the entries in the principal component corresponding to the clique vertices (marked by magenta dots) are not consistently high in absolute value. However, the vector returned by DSPCA, or the principal eigenvector of solution to the SDP relaxation described in (3), accurately identifies the clique vertices. Here, $\rho = 0.6$ was used. For this sparsity level, the DSPCA solution contains a very clear signal with high values at the clique vertices and is almost zero at background vertices.



(a) (b)

Fig. 1. Example PCA result (a) and DSPCA result (b).

DSPCA is an iterative algorithm that needs to compute a full eigendecomposition of a perturbation of the modularity matrix at each step. Thus, for large graphs, the algorithm can run quite slowly. However, while the clique does not stand out in the principal component, it may still be well-represented within a low-dimensional subspace corresponding to the eigenvectors with the largest eigenvalues. In this paper, we will show that a low-rank approximation of the modularity matrix can replace its full-rank counterpart in the DSPCA algorithm, to result in a faster method that still breaks the detection threshold go the PCA-based algorithm.

4. PERFORMANCE PREDICTION: APPROXIMATION AND LOWER BOUND

We can approximate the modularity matrix B as the sum of a real, symmetric matrix X and a rank-1 matrix $P = \theta u u^T$, where $X_{i,j} + p \sim \text{Bernoulli}(p)$, $\theta = k(1-p)$, and u is a vector whose ith entry u_i is $1/\sqrt{k}$ if $i \in V^*$ and is 0 otherwise. Note X is diagonalizable, and we can write $X = U_X^T D U_X$, where D is a diagonal matrix of eigenvalues, $\lambda_1 \geq \ldots \geq \lambda_n$, of X. The spectrum of $X + P$ is equal to the spectrum of $D + \theta \tilde{u} \tilde{u}^T$, where $\tilde{u} = U_X^T u$, and there is a unique mapping between their eigenvectors via U_X. Thus,

as in [13], we will analyze this matrix rather than the modularity matrix to determine the concentration of u among the eigenvectors.

Computing the principal m-dimensional eigenspace, $1 \leq m \leq n$, can be recast as maximizing the trace for a symmetric orthonormal projection of the original matrix, i.e.,

$$U^* = \arg\max_{U \in \mathcal{U}_m} \operatorname{tr}\left(U^T\left(D + \theta \tilde{u}\tilde{u}^T\right)U\right),$$

where $\mathcal{U}_m$ is the set of $n \times m$ matrices with orthonormal columns. Let:

$$V_m = \begin{bmatrix} I_{m-1} & 0_{m-1} \\ 0_{n-(m-1)\times m-1} & v \end{bmatrix},$$

where $v \in \mathbb{R}^{n-(m-1)}$. We will use $V_m \in \mathcal{U}_m$ to get a lower bound for the representation of $\tilde{u}$ in the principal eigenspace, since the first $m-1$ columns disregard the impact of $\tilde{u}$. We know that:

$$\operatorname{tr}\left(U^{*T}\left(D + \theta \tilde{u}\tilde{u}^T\right)U^*\right) \geq \max_v \operatorname{tr}\left(V_m^T(D + \theta \tilde{u}\tilde{u}^T)V_m\right).$$

We can rewrite the right-hand side as:

$$\max_v \sum_{i=1}^{m-1}\left(\lambda_i + \theta \tilde{u}_i^2\right) + \sum_{i=m}^{n} v_{i-m+1}^2 \lambda_i + \theta\left(\sum_{i=m}^{n} \tilde{u}_i v_{i-m+1}\right)^2. \quad (4)$$

We will approximate $\tilde{u}$ as a vector where each entry is $\pm 1/\sqrt{n}$ with equal probability. Through introducing a new vector $g \in \mathbb{R}^n$, where $g_i = \tilde{u}_i v_{i-m+1}$ for $i \geq m$ and $g_i = 0$ for $i < m$, we note that the quantity in (4) is maximized at the following value of g:

$$\arg\max_g \sum_{i=m}^{n} \frac{1}{n} g_i^2 \lambda_i + \theta\left(\sum_{i=m}^{n} \frac{1}{n} g_i\right)^2.$$

As $n \to \infty$, we can recast g as a continuous function, where we will maximize:

$$\int_a^1 g^2(x)\lambda(x)dx + \theta\left(\int_a^1 g(x)dx\right)^2,$$

where $m/n \to a$ as $n \to \infty$. Let $h(x) = -\int_x^1 g(t)dt$, so $h'(x) = g(x)$. Adding a Lagrange multiplier σ to ensure that g is unit length, we want to maximize:

$$\int_a^1 g^2(x)\lambda(x)dx + \theta\left(\int_a^1 g(x)dx\right)^2 + \sigma\left(\int_a^1 g^2(x)dx - 1\right).$$

Rearranging terms gives us an objective function J, where:

$$J = \int_a^1 L(x, h(x), g(x))dx - \sigma,$$

$$L(x, h(x), g(x)) = g^2(x)\lambda(x) - \theta g(x)h(a) + \sigma g^2(x).$$

Then by the Euler–Lagrange Theorem [14], J is maximized where:

$$\frac{d}{dx}\frac{\partial L}{\partial g} = 2\left[g(x)\left(\lambda(x) + \sigma\right)\right]' = 0,$$

meaning $g(x)\left(\lambda(x) + \sigma\right)$ must be a constant c. This yields a formula:

$$g(x) = \frac{c}{\lambda(x) + \sigma},$$

which can be substituted into the objective function, yielding:

$$J = c^2\left(\int_a^1 \frac{dx}{\lambda(x) + \sigma} + \theta\left(\int_a^1 \frac{dx}{\lambda(x) + \sigma}\right)^2\right) - \sigma.$$

Differentiating with respect to σ, we have:

$$\frac{\partial J}{\partial \sigma} = \int_a^1 \frac{-c^2 dx}{(\lambda(x) + \sigma)^2}\left(1 + 2\theta\left(\int_a^1 \frac{dx}{\lambda(x) + \sigma}\right)\right) - 1.$$

With the constraint that $g^2(x)$ integrates to 1, we can find the critical point at

$$\int_a^1 \frac{dx}{\lambda(x) + \sigma} = -\frac{1}{\theta}. \quad (5)$$

Differentiating with respect to c yields the same equation. We can use (5) to solve for σ at a given θ, then solve for c by normalizing the integral to 1. As $n \to \infty$, the distribution of eigenvalues will converge to a semicircle with radius $R = \sqrt{4np(1-p)}$ [15]. Therefore, we can change variables into a space where λ is the dependent variable and x is the cumulative density function of λ. We achieve this through the following substitution, where γ ranges from 0 to π:

$$\lambda(\gamma) = R\cos(\gamma)$$
$$x = f(\gamma) = \frac{1}{\pi}(\gamma - \sin(2\gamma)).$$

Substituting this into (5) gives us:

$$\frac{2}{\pi}\int_{f^{-1}(a)}^{\pi} \frac{\sin^2\gamma d\gamma}{R\cos(\gamma) + \sigma} = -\frac{1}{\theta}.$$

Solving this integral analytically is complicated and beyond the scope of this paper, but it expresses a relationship between R, a, and θ that can be computed numerically. Also, since the radius R of the semicircle determines the detection bound using the PCA technique, we can use this to characterize the norm of the projection of $\tilde{u}$ onto V_m independent of R by considering θ as a proportion of the detection threshold. After solving for c, we can return to the vector formulation:

$$\sum_{i=m}^{n} \frac{g_i/n}{c} = \sum_{i=m}^{n} \frac{1/n}{\lambda_i + \sigma} \approx -\frac{1}{\theta}.$$

The square of the L_2 norm of the projection of the signal vector $\tilde{u}$ onto the column space of V_m is given by:

$$\begin{aligned} \|V_m^T\tilde{u}\|_2^2 &= \sum_{i=1}^{m-1} \tilde{u}_i^2 + \left(\sum_{i=m}^{n} g_i\right)^2 \\ &= \frac{m-1}{n} + \left(\frac{nc}{\theta}\right)^2. \end{aligned} \quad (6)$$

Then, using (6) we can plot $\|V_m^T\tilde{u}\|$, a lower bound of the projection of $\tilde{u}$ onto U^*, as shown in Fig. 2. The percentages (relative to the threshold in Theorem 3.1) correspond to cliques of size $k = 10, 8, 6$, and 4 embedded into a graph where $N = 5000$ and $p = 0.01$. $\|V_m^T\tilde{u}\|$ is a lower bound for $\|U^{*T}\tilde{u}\|$, the projection into the top m eigenvectors. Even significantly below the PCA detection threshold, a large portion of the signal vector's magnitude lies in the space spanned by a relatively small percentage of eigenvectors.

Fig. 3 compares the prediction to empirical performance. The black dashed curve is predicted based on the red curve in Fig. 2, the blue curve uses V_m for a random instantiation, the green curve uses U^* when $\tilde{u}$ has entries that are $\pm 1/\sqrt{n}$, and the red curve uses U^* when $\tilde{u}$ each entry is drawn from a standard normal and the vector is unit-normalized. The approximation where $\tilde{u}$ has equal magnitude in each component is always greater than the lower bound, and we see similar performance when using a $\tilde{u}$ whose entries are drawn from a normal distribution.

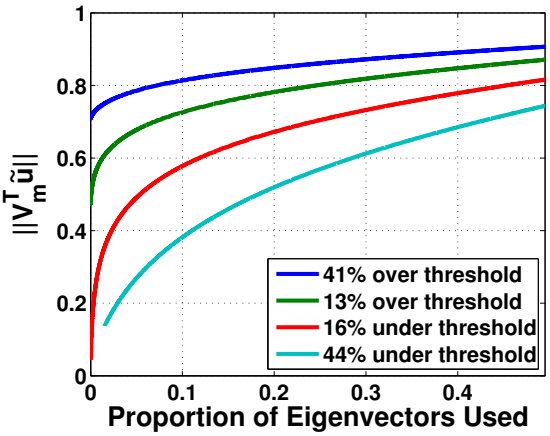


Fig. 2. L_2 norm of $V_m^T \tilde{u}$ for different clique sizes.

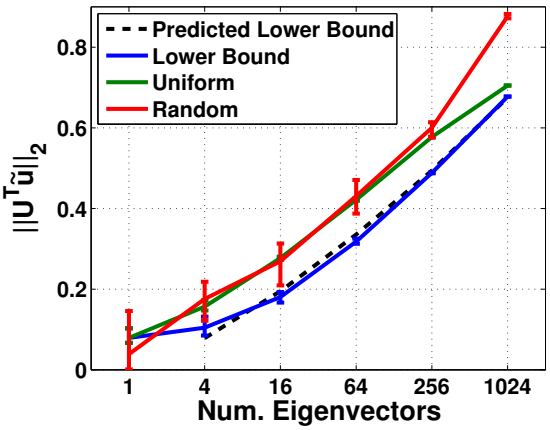
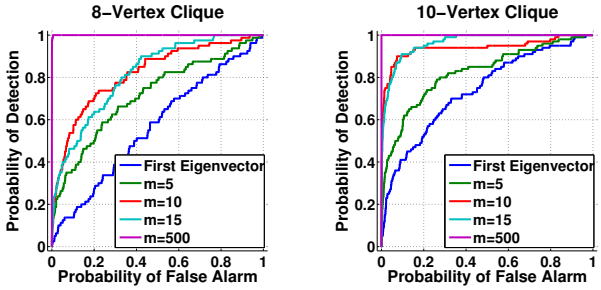


Fig. 3. Norm of $\tilde{u}$ when projected into the principal eigenspace, for $n = 5000$, $p = 0.01$, and $\theta = 6$. The plot includes the predicted lower bound, the average lower bound, the average using true eigenvectors for a uniform $\tilde{u}$, and the average using a random $\tilde{u}$ from the unit hypersphere. Error bars indicate the maximum and minimum over 10 random instantiations.



(a) (b)

Fig. 4. DSPCA detection performance for several values of m, for $G(500, 0.2)$ and $k = 8$ (a) or $k = 10$ (b).

5. SIMULATION RESULTS

When an approximation of the modularity matrix is used, we obtain B_m, a rank-m approximation of B, by keeping the first m terms of its eigendecomposition:

$$B_m = \sum_{i=1}^{m} \lambda_i u_i u_i^T,$$

where λ_i is the ith largest eigenvalue of B, and u_i is the corresponding eigenvector. We will evaluate how the performance of the DSPCA algorithm varies as a function of m through generating receiver operating characteristic (ROC) curves that use the entries of the vector returned by DSPCA as test statistics.

The results in this section demonstrate the outcomes of 100-trial Monte Carlo simulations. In each simulation, an Erdős-Rényi graph $G(500, 0.2)$ is generated, and a clique of size $k = 8$ or $k = 10$ is embedded on a subset of its vertices. Fig. 4 demonstrates that, even within the regime below the threshold elucidated in Theorem 3.1 (in this case approximately 11.18), most of the power is in the upper eigenvectors. The DSPCA solution to the planted clique problem represents a significant improvement over PCA methods, even in the case of small m.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have demonstrated that applying DSPCA to a low-rank approximation of the graph's modularity matrix is a viable algorithm to solve the planted clique problem. When the size of a planted clique is reduced below the threshold where it is detectable via PCA, it may still be well-represented in relatively few principal eigenvectors, which enables higher detection performance in a low-dimensional space. One future direction involves determining the detection threshold for SPCA algorithms. We are currently working on quantifying the breakdown point for the DSPCA approach, using either the modularity matrix or its rank-m approximation, to the planted clique problem. Since SPCA is NP-hard and thus computationally intractable, all existing algorithmic approaches motivated by SPCA are relaxations of the initial formulation described in (2). Alternative SPCA algorithms could prove to have higher detection power than DSPCA.

Other future work involves studying the relationship between the parameter ρ in DSPCA and the associated false-alarm probability of the algorithm. This will involve analyzing the noise characteristics as more eigenvectors are added. Combined with some more extensive Monte Carlo simulations over the various parameters of the model, this could reveal a method to directly compute a detectability bound based on the background parameters, foreground parameters, and number of eigenvectors used.

7. REFERENCES

[1] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, no. 3, 2006.

[2] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, 2004.

[3] Jianhua Ruan and Weixiong Zhang, "An efficient spectral algorithm for network community discovery and its applications

to biological and social networks," in *Proc. IEEE Int. Conf. Data Mining*, 2007, pp. 643–648.

[4] Ery Arias-Castro and Nicolas Verzelen, "Community detection in random networks," Preprint: arXiv.org:1302.7099, 2013.

[5] Noga Alon, Michael Krivelevich, and Benny Sudakov, "Finding a large hidden clique in a random graph," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, 1998, pp. 594–598.

[6] Yash Deshpande and Andrea Montanari, "Finding hidden cliques of size $\sqrt{N/e}$ in nearly linear time," preprint: http://www.stanford.edu/∼montanar/RESEARCH/FILEPAP/clique.pdf, 2013.

[7] Benjamin A. Miller, Nadya T. Bliss, Patrick J. Wolfe, and Michelle S. Beard, "Detection theory for graphs," *Lincoln Laboratory J.*, vol. 20, no. 1, 2013.

[8] Raj Rao Nadakuditi, "On hard limits of eigen-analysis based planted clique detection," in *Proc. IEEE Statistical Signal Process. Workshop*, 2012, pp. 129–132.

[9] Navraj Singh, Benjamin A. Miller, Nadya T. Bliss, and Patrick J. Wolfe, "Anomalous subgraph detection via sparse principal component analysis," in *Proc. IEEE Statistical Signal Process. Workshop*, 2011, pp. 485–488.

[10] Quentin Berthet and Philippe Rigollet, "Complexity theoretic lower bounds for sparse principal component detection," in *Conf. Learning Theory*, Shai Shalev-Shwartz and Ingo Steinwart, Eds., vol. 30 of *JMLR W&CP*, pp. 1046–1066. 2013.

[11] Florent Benaych-Georges, "Eigenvectors of Wigner matrices: Universality of global fluctuations," preprint: http://arxiv.org/abs/1104.1219, 2011.

[12] Alexandre D'Aspremont, Laurent El Ghaoui, Michael I. Jordan, and Gert R. G. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," *SIAM Review*, vol. 49, no. 3, pp. 434–448, 2007.

[13] Florent Benaych-Georges and Raj Rao Nadakuditi, "The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices," *Advances in Math.*, vol. 227, no. 1, pp. 494–521, May 2011.

[14] I. M. Gelfand and S. V. Fomin, *Calculus of Variations*, Dover, 1963.

[15] Zhidong Bai and Jack W. Sliverstein, *Spectral Analysis of Large Dimensional Random Matrices*, Springer, second edition, 2010.