MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Application of Semidefinite Programming to Maximize the Spectral Gap Produced by Node Removal

Naoki MASUDA, Tetsuya FUJIE and Kazuo MUROTA

METR 2011–37 October 2011

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.keisu.t.u-tokyo.ac.jp/research/techrep/index.html

Application of Semidefinite Programming to Maximize the Spectral Gap Produced by Node Removal

Naoki MASUDA, Tetsuya FUJIE and Kazuo MUROTA

October 31, 2011

Abstract

The smallest positive eigenvalue of the Laplacian of a network is called the spectral gap and characterizes various dynamics on networks. We propose mathematical programming methods to maximize the spectral gap of a given network by removing a fixed number of nodes. We formulate relaxed versions of the original problem using semidefinite programming and apply them to example networks.

1 Introduction

An undirected and unweighted network (i.e., graph) on N nodes is equivalent to an $N \times N$ symmetric adjacency matrix $A = (A_{ij})$, where $A_{ij} = 1$ when nodes (also called vertices) i and j form a link (also called edge) such that they are adjacent, and $A_{ij} = 0$ otherwise. We define the Laplacian matrix of the network by $L \equiv D - A$, where D is the $N \times N$ diagonal matrix in which the ith diagonal element is equal to $\sum_{j=1}^{N} A_{ij}$, i.e., the degree of node i.

When the network is connected, the eigenvalues of L satisfy $\lambda_1 = 0 < \lambda_2 \leq \cdots \leq \lambda_N$. The eigenvalue λ_2 is called spectral gap or algebraic connectivity and characterizes various dynamics on networks including synchronizability [1, 2, 3], speed of synchronization [1], consensus dynamics [4], the speed of convergence of the Markov chain to the stationary density [3, 5], and the first-passage time of the random walk [3]. Because a large λ_2 is often considered to be desirable, e.g., for strong synchrony and high speed of convergence, maximization of λ_2 by changing networks under certain constraints is important in the context of practical applications.

In the present work, we consider the problem of maximizing the spectral gap by removing a specified number, $N_{\rm del}$, of nodes from a given network. We assume that an appropriate choice of $N_{\rm del}$ nodes keeps the network

connected. A heuristic algorithm for this task in which nodes are sequentially removed is proposed in [6]. In this study, we explore a mathematical programming approach. We propose two algorithms using semidefinite programming and numerically compare their performance with that of the sequential algorithm proposed in [6].

2 Methods

We start by introducing notations. First, the binary variable x_i $(1 \leq i \leq N)$ takes a value of 0 if node i is one of the $N_{\rm del}$ removed nodes and 1 if node i survives the removal. Our goal is to determine the x_i $(1 \leq i \leq N)$ that maximizes λ_2 under the constraint

$$\sum_{i=1}^{N} x_i = N - N_{\rm del}. \tag{1}$$

Second, we define $\tilde{L}_{ij}$ as the $N \times N$ Laplacian matrix generated by a single link $(i,j) \in E$, where E is the set of links. In other words, the (i,i) and (j,j) elements of $\tilde{L}_{ij}$ are equal to 1, the (i,j) and (j,i) elements of $\tilde{L}_{ij}$ are equal to -1, and all the other elements of $\tilde{L}_{ij}$ are equal to 0. It should be noted that

$$L = \sum_{1 \leq i < j \leq N; (i,j) \in E} \tilde{L}_{ij}. \tag{2}$$

Third, J denotes the $N \times N$ matrix in which all the N^2 elements are equal to unity. Fourth, E_i denotes the $N \times N$ diagonal matrix in which the (i,i) element is equal to unity and all the other $N^2 - 1$ elements are equal to 0.

After the removal of $N_{\rm del}$ nodes, we do not decrease the size of the Laplacian. Instead, we remove $\tilde{L}_{ij}$ $((i,j) \in E)$ from the summation on the RHS of Eq. (2) if node i or j has been removed from the network. The Laplacian of the remaining network, if connected, has $N_{\rm del} + 1$ zero eigenvalues. These zero eigenvectors are given by $\boldsymbol{u}^{(0)} \equiv (1 \cdots 1)^\top$ and $\boldsymbol{e}_i$, where $\top$ denotes the transpose, $\boldsymbol{e}_i$ is the unit column vector in which the ith element is equal to 1 and the other $N - 1$ elements are equal to 0, and i is the index of one of the $N_{\rm del}$ removed nodes.

Allowing nonlinearity, we formulate the eigenvalue problem as follows; we name this problem EIGEN:

$$\text{maximize } t \quad \text{subject to Eq. (1) and}$$

$$-tI + \sum_{i<j;(i,j)\in E} x_i x_j \tilde{L}_{ij} + \alpha J + \beta \sum_{i=1}^{N} (1 - x_i) E_i \succeq 0, \tag{3}$$

and $x_i \in \{0,1\}$ $(1 \leq i \leq N)$, where $\succeq 0$ indicates that the LHS is a semidefinite matrix. The semidefinite constraint Eq. (3) is derived from a

standard prescription in semidefinite programming for optimization of an extreme eigenvalue of a matrix. Maximizing t is equivalent to maximizing the smallest eigenvalue of the matrix given by the summation of the second, third, and fourth terms on the LHS of Eq. (3).

Without the third and fourth terms on the LHS of Eq. (3), the optimal solution would be trivially equal to $t = 0$ because the Laplacian of any network has 0 as the smallest eigenvalue. Because $J = \boldsymbol{u}^{(0)}\boldsymbol{u}^{(0)\top}$, the third term transforms a zero eigenvalue to $\approx \alpha$. We should take a sufficiently large $\alpha > 0$ such that the zero eigenvalue is shifted to a value larger than the spectral gap of the remaining network, denoted by $\tilde{\lambda}_2$. This technique was introduced in [7] for solving the traveling salesman problem.

For each removed node i (i.e., $x_i = 0$), the matrix represented by the second term on the LHS of Eq. (3) has a zero eigenvalue associated with eigenvector $\boldsymbol{e}_i$. The fourth term shifts this zero eigenvalue to $\approx \beta$. Note that the fourth term disappears for the remaining $N - N_{\mathrm{del}}$ nodes because $x_i = 1$ for the remaining nodes. If the transformed eigenvalues are larger than $\tilde{\lambda}_2$, the solution to the problem stated above returns the N_{del} nodes whose removal maximizes $\tilde{\lambda}_2$.

The second term on the LHS of Eq. (3) represents a nonlinear constraint. To linearize the problem in terms of the variables, we follow a conventional prescription to introduce auxiliary variables $X_{ij} \equiv x_i x_j$, where $1 \le i \le j \le N$ [8, 9, 10] (also reviewed in [11]). If x_i is discrete, $x_i(1 - x_i) = 0$ holds true. Therefore, we require $X_{ii} = x_i^2 = x_i$. In the following discussion, we use x_i in place of X_{ii}.

We define the $(N + 1) \times (N + 1)$ matrix

$$Y \equiv \begin{bmatrix} 1 & \boldsymbol{x}^\top \\ \boldsymbol{x} & X \end{bmatrix}, \tag{4}$$

where $\boldsymbol{x} \equiv (x_1 \ \ldots \ x_N)^\top$, the (i, i) element of the $N \times N$ matrix X is equal to x_i, and the (i, j) element $(i \neq j)$ of X is equal to X_{ij}. By allowing x_i and X_{ij} $(1 \le i < j \le N)$ to take any continuous value between 0 and 1, we define the relaxed problem named SDP1 as follows:

$$\text{maximize } t \quad \text{subject to Eq. (1) and}$$

$$-tI + \sum_{i<j;(i,j)\in E} X_{ij}\tilde{L}_{ij} + \alpha J + \beta \sum_{i=1}^{N} (1 - x_i)E_i \succeq 0, \tag{5}$$

$$Y \succeq 0. \tag{6}$$

Note that Eq. (6) implies $0 \le x_i \le 1$ $(1 \le i \le N)$ and that SDP1 relaxes the original problem in that x_i and X_{ij} are allowed to take continuous values

while Eq. (6) is imposed. The method that we propose here for approximately maximizing the spectral gap is to remove the $N_{\rm del}$ nodes corresponding to the $N_{\rm del}$ smallest values among $x_1, \ldots, x_N$ in the optimal solution of SDP1.

SDP1 involves $N(N+1)/2+1$ variables (i.e., t, x_i, and X_{ij}, where $i<j$). In fact, X_{ij} for which $(i,j) \notin E$ is free unless Eq. (6) is violated; it does not appear in the main semidefinite constraint represented by Eq. (5). Because a given network is typically sparse, this implies that there are many redundant variables in SDP1. To exploit the sparsity and thus to save time and memory space, a technique based on matrix completion might be useful [12, 13]. In this paper, however, we propose another relaxation SDP2 for this purpose.

To linearize the second term on the LHS of Eq. (3), we take advantage of four inequalities $x_i x_j \geq 0$, $x_i(1-x_j) \geq 0$, $(1-x_i)x_j \geq 0$, and $(1-x_i)(1-x_j) \geq 0$ that must be satisfied for any link $(i,j) \in E$. By defining $X_{ij} \equiv x_i x_j$, as in the case of SDP1, we obtain the following four linear constraints [14]:

$$X_{ij} \geq 0, \tag{7}$$

$$x_i - X_{ij} \geq 0, \tag{8}$$

$$x_j - X_{ij} \geq 0, \tag{9}$$

$$1 - x_i - x_j + X_{ij} \geq 0. \tag{10}$$

SDP2 is defined by replacing Eq. (6) by Eqs. (7)–(10), where only the pairs $(i,j) \in E$ are considered. Note that Eqs. (7)–(10) guarantee $0 \leq x_i \leq 1$ $(1 \leq i \leq N)$. We remove the $N_{\rm del}$ nodes corresponding to the $N_{\rm del}$ smallest values among $x_1, \ldots, x_N$ in the optimal solution of SDP2.

Numerically, SDP2 is much easier to solve than SDP1 for two reasons. First, the number of variables is smaller in SDP2. In SDP2, X_{ij} is defined only on the links, whereas in SDP 1 it is defined for all the pairs $1 \leq i < j \leq N$. In sparse networks, the number of variables is $O(N^2)$ for SDP1 and $O(N)$ for SDP2. Second, the semidefinite constraint, which is much more time consuming to solve than a linear constraint of a comparable size, is smaller in SDP2 than in SDP1. While SDP1 and SDP2 share the $N \times N$ semidefinite constraint (5), SDP1 involves an additional semidefinite constraint (6) of size $(N+1) \times (N+1)$.

To choose the parameter values α and β, we consider the matrix represented by the sum of the second, third, and fourth terms on the LHS of Eq. (3). A straightforward calculation shows that the eigenvalues of this matrix are given by the $N - N_{\rm del} - 1$ positive eigenvalues of the Laplacian of the remaining network, $(N_{\rm del}-1)$-fold β, and $\beta + \left[\alpha N - \beta \pm \sqrt{(\alpha N - \beta)^2 + 4 N_{\rm del} \alpha \beta}\right]/2$. For a fixed β, we should choose α to maximize $\beta + \left[\alpha N - \beta - \sqrt{(\alpha N - \beta)^2 + 4 N_{\rm del} \alpha \beta}\right]/2$, which is always smaller than eigenvalue β. We set $\alpha = \beta/N$ to simplify the

expression of this eigenvalue to $\beta(1-\sqrt{N_{\rm del}/N})$ while approximately maximizing this eigenvalue.

We have the following bounds for the optimal solution to the original problem. We denote by $\tilde{\lambda}_2^{\rm opt}$ the optimal solution, i.e., the maximum spectral gap with $N_{\rm del}$ nodes removed. We denote by $\tilde{\lambda}_2^{\rm SDP}$ the smallest positive eigenvalue of the network obtained by the proposed method; the proposed method removes the $N_{\rm del}$ nodes corresponding to the $N_{\rm del}$ smallest values of x_1, ..., x_N in the optimal solution of SDP1 or SDP2. Obviously, $\tilde{\lambda}_2^{\rm SDP}$ is a lower bound for $\tilde{\lambda}_2^{\rm opt}$. On the other hand, the optimal value, $\max t$, of SDP1 or SDP2 serves as an upper bound for $\tilde{\lambda}_2^{\rm opt}$, as long as the parameter β is chosen to satisfy $\tilde{\lambda}_2^{\rm opt} \leq \beta(1-\sqrt{N_{\rm del}/N})$. This follows from the facts that the optimal value of EIGEN with such a β value coincides with $\tilde{\lambda}_2^{\rm opt}$ and both SDP1 and SDP2 are a relaxation of EIGEN. We can summarize our observation as follows: $\tilde{\lambda}_2^{\rm SDP} \leq \tilde{\lambda}_2^{\rm opt} \leq \max t$.

3 Numerical results

We apply SDP1 and SDP2 to the real data used in our previous paper [6]. We implement SDP1 and SDP2 using the free software package SeDuMi that runs on MATLAB [15].

We compare the performance of SDP1 and SDP2 with that of the optimal sequential method, which is a heuristic method proposed in [6]. In the optimal sequential method, we numerically calculate the spectral gap (i.e., smallest positive eigenvalue of the Laplacian) obtained by the removal of one node; we perform this calculation for all possible choices of nodes. Subsequently, we remove the node whose removal yields the largest spectral gap. Then, for the remaining network composed of $N-1$ nodes, we implement the same procedure to determine the second node to be removed. We repeat this procedure until $N_{\rm del}$ nodes have been removed.

The first network that we examine is the largest connected component of the undirected and unweighted version of a macaque cortical network [16]. The network has $N = 71$ nodes and 438 links. We set $\beta = 2$.

The spectral gap obtained by the different methods is shown in Fig. 1(a) as a function of $N_{\rm del}$. Up to $N_{\rm del} = 4$, the optimal sequential method yields the exact solution, as do SDP1 and SDP2. For $N_{\rm del} \geq 5$, we could not obtain the exact solution because of the combinatorial explosion. For $N_{\rm del} \geq 5$, SDP1 and SDP2 perform worse than the optimal sequential method. The final values of x_i ($1 \leq i \leq N$) are not bimodally distributed around 0 and 1. The distribution is rather unimodal except for the first three values of x_i that are close to 0. The ten values of x_i when $N_{\rm del} = 5$, in ascending order, are as follows: $x_{33} = 0.1086$, $x_{62} = 0.1531$, $x_{53} = 0.1589$, $x_1 = 0.4813$, $x_2 = 0.5246$, $x_8 = 0.5591$, $x_7 = 0.6449$, $x_{24} = 0.7866$, $x_{51} = 0.8749$, and $x_{63} = 0.8931$ in SDP1, and $x_{53} = 0.000$, $x_{33} = 0.145$, $x_{62} = 0.177$, $x_2 = 0.585$, $x_1 = 0.588$,

$x_8 = 0.610$, $x_7 = 0.668$, $x_{24} = 0.708$, $x_5 = 0.738$, and $x_4 = 0.937$ in SDP2. We consider that the nature of this distribution to be the reason why SDP1 and SDP2 perform poorly as compared to the optimal sequential method.

The second network is the largest connected component of the *C. elegans* neural network [17, 18]. Two nodes are regarded as being connected when they are connected by a chemical synapse or gap junction. We ignore the direction and weight of links. The network has $N = 279$ nodes and 2287 links. We set $\beta = 2.5$. SDP1 did not work for this network because N is too large. Therefore, we only tried to solve SDP2.

The results for SDP2 and the optimal sequential method are shown in Fig. 1(b). Although SDP2 gradually increases the spectral gap as $N_{\rm del}$ increases, which is nontrivial, it again performs poorly compared to the optimal sequential method. The 15 values of x_i when $N_{\rm del} = 10$, in ascending order, are as follows: $x_{274} = 0.0988$, $x_{148} = 0.1350$, $x_{149} = 0.1479$, $x_{36} = 0.2215$, $x_{93} = 0.2215$, $x_{107} = 0.8510$, $x_{186} = 0.8604$, $x_{137} = 0.8637$, $x_{229} = 0.8817$, $x_{230} = 0.8820$, $x_{123} = 0.8831$, $x_{275} = 0.8975$, $x_{127} = 0.8980$, $x_{122} = 0.8990$, and $x_{132} = 0.8990$.

4 Discussion

We proposed a method to maximize the spectral gap using semidefinite programming. Although the two algorithms have a firmer mathematical foundation as compared to the heuristic numerical methods (i.e., brute force method and optimal sequential method), they perform worse than the heuristic methods.

We should also be careful about the choice of β. If β is too large, SDP1 and SDP2 would result in $x_i \approx N_{\rm del}/N$ $(1 \le i \le N)$. This is because setting $x_i = N_{\rm del}/N$ $(1 \le i \le N)$ makes the fourth term on the LHS of Eq. (3) equal to $\beta \frac{N-N_{\rm del}}{N} I$, which increases all the eigenvalues, including the spectral gap of the remaining network, by $\beta \frac{N-N_{\rm del}}{N}$. In contrast, if β is smaller than $\tilde{\lambda}_2$, SDP1 and SDP2 would maximize the false eigenvalue originating from the fourth term on the LHS of Eq. (3).

To enhance the performance of SDP1 and SDP2, it may be useful to neglect the convexity of the problem. For example, we could replace $(1 - x_i)$ in the fourth term by $(1 - x_i)^p$ and gradually increase p from unity. When $p > 1$, the problem is no longer convex. Accordingly, the existence of the unique solution or the convergence of a proposed algorithm is not guaranteed. Nevertheless, we may be able to track the optimal solution $\boldsymbol{x}$ by the Newton method while we gradually increase p (see p.5 and p.63 in [19]). An alternative extension is to add $-p \sum_{i=1}^{N} x_i(1 - x_i)$ to the objective function to be maximized (i.e., t). When $p > 0$, the convexity is violated. However, we may be able to adopt a procedure similar to the method explained above, i.e., start with $p = 0$ and gradually increase p to track the solution by the

Newton method.

Acknowledgments

Naoki Masuda acknowledges the financial support of the Grants-in-Aid for Scientific Research (no. 23681033) from MEXT, Japan. This research is also partially supported by the Aihara Innovative Mathematical Modelling Project, the Japan Society for the Promotion of Science (JSPS) through the "Funding Program for World-Leading Innovative R&D on Science and Technology (FIRST Program)", initiated by the Council for Science and Technology Policy (CSTP).

References

[1] J. A. Almendral and A. Díaz-Guilera. Dynamical and spectral properties of complex networks. *New J. Phys.*, 9:187, 2007.

[2] A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, and C. Zhou. Synchronization in complex networks. *Phys. Rep.*, 469:93–153, 2008.

[3] L. Donetti, F. Neri, and M. A. Munoz. Optimal network topologies: expanders, cages, Ramanujan graphs, entangled networks and all that. *J. Stat. Mech.*, page P08007, 2006.

[4] R Olfati-Saber, J Fax, and R Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95:215–233, 2007.

[5] D. Cvetković, P. Rowlinson, and S. Simić. *An introduction to the theory of graph spectra*. CMU, 2010.

[6] T. Watanabe and N. Masuda. Enhancing the spectral gap of networks by node removal. *Phys. Rev. E*, 82:046102, 2010.

[7] D. Cvetković, C. Čangalović, and V. Kovačević-Vujčić. Semidefinite programming methods for the symmetric traveling salesman problem. In *Integer Programming and Combinatorial Optimization (LNCS)*, volume 1610, pages 126–136, 1999.

[8] L. Lovász. On the shannon capacity of a graph. *IEEE Trans. on Info. Th.*, 25:1–7, 1979.

[9] L Lovász and A Schrijver. Cones of matrices and set-functions and 0–1 optimization. *Siam J. Optimiz.*, 1:166–190, 1991.

[10] M. Grötschel, L. Lovász, and A. Schrijver. Relaxations of vertex packing. *J. Comb. Theory B*, 40:330–343, 1986.

[11] M. X. Goemans. Semidefinite programming in combinatorial optimization. *Math. Programming*, 79:143–161, 1997.

[12] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion I: general framework. *SIAM J. Optim.*, 11:647–674, 2000.

[13] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota. Exploiting sparsity in semidefinite programming via matrix completion II: implementation and numerical results. *Math. Program. Ser. B*, 95:305–327, 2003.

[14] M Padberg. The Boolean quadric polytope – some characteristics, facets and relatives. *Mathematical Programming*, 45(1):139–172, 1989.

[15] http://sedumi.ie.lehigh.edu.

[16] O. Sporns and J. D. Zwi. The small world of the cerebral cortex. *Neuroinformatics*, 4:145–162, 2004.

[17] B. L. Chen, D. H. Hall, and D. B. Chklovskii. Wiring optimization can relate neuronal structure and function. *Proc. Natl. Acad. Sci. USA*, 103:4723–4728, 2006.

[18] http://www.wormatlas.org.

[19] M. P. Bendsøe and O. Sigmund. *Topology optimization.* Springer, 2003.

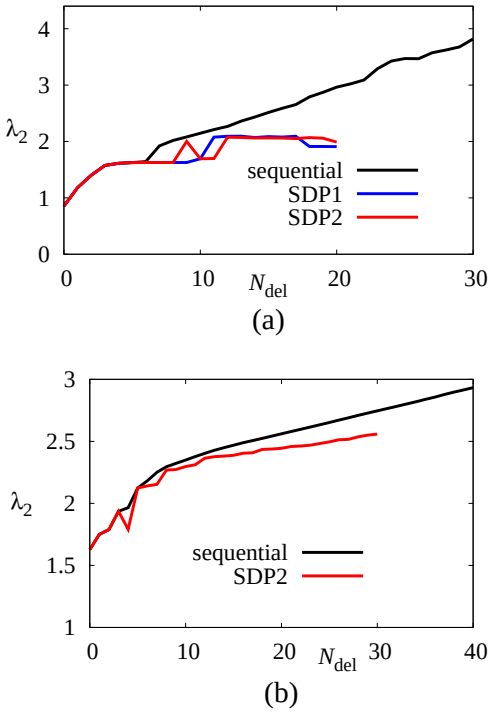

Figure 1: Spectral gap as a function of the number of removed nodes. (a) Results for the macaque cortical network with $N = 71$ nodes. We set $\beta = 2$. (b) Results for the *C. elegans* neural network with $N = 279$ nodes. We set $\beta = 2.5$.