Convex Optimization of Graph Laplacian Eigenvalues

Stephen Boyd*

Abstract. We consider the problem of choosing the edge weights of an undirected graph so as to maximize or minimize some function of the eigenvalues of the associated Laplacian matrix, subject to some constraints on the weights, such as nonnegativity, or a given total value. In many interesting cases this problem is convex, *i.e.*, it involves minimizing a convex function (or maximizing a concave function) over a convex set. This allows us to give simple necessary and sufficient optimality conditions, derive interesting dual problems, find analytical solutions in some cases, and efficiently compute numerical solutions in all cases.

In this overview we briefly describe some more specific cases of this general problem, which have been addressed in a series of recent papers.

- *Fastest mixing Markov chain.* Find edge transition probabilities that give the fastest mixing (symmetric, discrete-time) Markov chain on the graph.
- *Fastest mixing Markov process.* Find the edge transition rates that give the fastest mixing (symmetric, continuous-time) Markov process on the graph.
- *Absolute algebraic connectivity.* Find edge weights that maximize the algebraic connectivity of the graph (*i.e.*, the smallest positive eigenvalue of its Laplacian matrix). The optimal value is called the *absolute algebraic connectivity* by Fielder.
- *Minimum total effective resistance.* Find edge weights that minimize the total effective resistance of the graph. This is same as minimizing the average commute time from any node to any other, in the associated Markov chain.
- *Fastest linear averaging.* Find weights in a distributed averaging network that yield fastest convergence.
- *Least steady-state mean-square deviation.* Find weights in a distributed averaging network, driven by random noise, that minimizes the steady-state mean-square deviation of the node values.

Mathematics Subject Classification (2000). Primary 05C35; Secondary 90C25.

Keywords. Graph theory, Laplacian matrix, Convex optimization, Semidefinite programming, Markov chain, Distributed averaging, Effective resistance.

*Supported in part by the MARCO Focus Center for Circuit and System Solutions (C2S2, www.c2s2.org) under contract 2003-CT-888, by AFOSR grant AF F49620-01-1-0365, by NSF grant ECS-0423905, and by DARPA/MIT grant 5710001848. This paper reports work with coauthors Persi Diaconis, Arpita Ghosh, Seung-Jean Kim, Sanjay Lall, Pablo Parrilo, Amin Saberi. Jun Sun, and Lin Xiao.

1. Introduction

Let $G = (V, E)$ be an undirected graph with $n = |V|$ nodes and $m = |E|$ edges, with weights $w_1, \ldots, w_m \in \mathbf{R}$ on the edges. Suppose edge l connects vertices (or nodes) i and j. We define $a_l \in \mathbf{R}^n$ as $(a_l)_i = 1$, $(a_l)_j = -1$, with other entries 0. The *weighted Laplacian* (matrix) is the $n \times n$ matrix defined as

$$L = \sum_{l=1}^{m} w_l a_l a_l^T = A \, \mathbf{diag}(w) A^T,$$

where $\mathbf{diag}(w) \in \mathbf{R}^{m \times m}$ is the diagonal matrix formed from $w = (w_1, \ldots, w_m) \in \mathbf{R}^m$, and $A \in \mathbf{R}^{n \times m}$ is the *incidence matrix* of the graph, $A = [a_1 \ \cdots \ a_m]$.

We assume that the weights are such that L is positive semidefinite, which we write as $L \succeq 0$. This is always the case when the weights are nonnegative. Since $L\mathbf{1} = 0$, where $\mathbf{1}$ is the vector with all components one, L has smallest eigenvalue 0, corresponding to the eigenvector $\mathbf{1}$. We denote the eigenvalues of the Laplacian matrix L as

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

Let ϕ be a symmetric closed convex function defined on a convex subset of $\mathbf{R}^{n-1}$. Then

$$\psi(w) = \phi(\lambda_2, \ldots, \lambda_n) \tag{1}$$

is a convex function of w [2, §5.2]. Thus, a symmetric convex function of the positive Laplacian eigenvalues yields a convex function of the edge weights. As a simple example, consider $\phi(u_1, \ldots, u_{n-1}) = \sum_{i=1}^{n-1} u_i$, *i.e.*, the sum. In this case we have

$$\psi(w) = \sum_{i=2}^{n} \lambda_i = \sum_{i=1}^{n} \lambda_i = \mathbf{Tr}\, L = 2\mathbf{1}^T w,$$

twice the sum of the edge weights, which is linear and therefore also convex. As another example, the function $\phi(u_1, \ldots, u_{n-1}) = \max_{i=1}^{n-1} u_i$ (which is convex and symmetric) yields the function $\psi(w) = \lambda_n$, the largest eigenvalue (or spectral radius) of the Laplacian matrix (and a convex function of the edge weights).

We consider optimization problems with the general form

$$\begin{array}{ll} \text{minimize} & \psi(w) \\ \text{subject to} & w \in \mathcal{W}, \end{array} \tag{2}$$

where $\mathcal{W}$ is a closed convex set, and the optimization variable here is $w \in \mathbf{R}^m$. The problem (2) is to choose edge weights on a graph, subject to some constraints, in order to minimize a convex function of the positive eigenvalues of the associated Laplacian matrix. We can also handle the case of maximizing a concave function ϕ of the positive Laplacian eigenvalues, by minimizing $-\psi$ over $w \in \mathcal{W}$.

The problem (2) is a *convex optimization problem*. Roughly speaking, this means that the analysis of the problem is fairly straightforward, and that the problem is easily solved numerically; see, *e.g.*, [6]. In the cases we will consider,

the problem (2) can be formulated even more specifically as a *semidefinite program* (SDP), which has the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \sum_{i=1}^n x_i A_i \preceq B. \end{array} \tag{3}$$

Here $x \in \mathbf{R}^n$ is the variable, and the problem data are $c \in \mathbf{R}^n$ and the symmetric matrices $A_1, \ldots, A_n, B \in \mathbf{R}^{k \times k}$. The inequality symbol $\preceq$ between symmetric matrices refers to inequality with respect to the cone of positive semidefinite matrices. The constraint $\sum_{i=1}^n x_i A_i \preceq B$ is called a *linear matrix inequality* (LMI). The SDP (3) can be thought of as a generalization of a linear program (LP),

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \sum_{i=1}^n x_i a_i \leq b, \end{array}$$

where here, $a_1, \ldots, a_n, b$ are vectors, and the inequality symbol between vectors means componentwise. Many results for LPs have analogs for SDPs; moreover, in the last 15 years or so, effective algorithms for numerically solving SDPs have been developed, and are now widely used in many application areas.

2. Fastest mixing Markov chain

In this section we briefly describe the problem of finding the fastest mixing symmetric Markov chain on a given graph. Many more details (and additional references) can be found in [4, 5].

We consider a symmetric Markov chain on the graph G, with transition matrix $P \in \mathbf{R}^{n \times n}$, where $P_{ij} = P_{ji}$ is the probability of a transition from vertex i to vertex j. Since P is symmetric, the uniform distribution $\pi_{\text{unif}} = (1/n)\mathbf{1}^T$ is an equilibrium distribution. The rate of convergence of the distribution $\pi(t)$ to uniform is governed by $\mu(P)$, the *second largest eigenvalue magnitude* (SLEM) of P, with smaller $\mu(P)$ meaning faster asymptotic convergence. To find the fastest mixing symmetric Markov chain on the graph, we must choose the transition matrix P to minimize $\mu(P)$, subject to the following conditions:

$$P = P^T, \quad P\mathbf{1} = \mathbf{1}, \quad P_{ij} \geq 0, \quad i, j = 1, \ldots, n, \quad P_{ij} = 0 \text{ for } (i, j) \notin E.$$

The first three conditions state that P is a symmetric stochastic matrix; the last states that transitions can only occur over the graph edges.

Identifying the graph edge weights with edge transition probabilities, we find that P can be expressed as $P = I - L$. The conditions above are equivalent to the conditions

$$w \geq 0, \qquad \mathbf{diag}(L) \leq \mathbf{1}$$

imposed on the edge weight vector w. (Here $\mathbf{diag}(L)$ is the vector consisting of the diagonal entries of L, and both inequalities above are vector inequalities, *i.e.*, componentwise.)

The eigenvalues of P are $1-\lambda_1, \ldots, 1-\lambda_n$. Since $1-\lambda_1 = 1$, and $|1-\lambda_i| \leq 1$ (since P is stochastic), its SLEM is given by

$$\mu(P) = \max\{|1-\lambda_2|, \ldots, |1-\lambda_n|\} = \max\{1-\lambda_2, \lambda_n - 1\}. \tag{4}$$

This has the general form (1), with $\phi(u_1, \ldots, u_{n-1}) = \max_{i=1}^{n-1} |1-u_i|$. In particular, the SLEM $\mu(P)$ is a convex function of the edge transition probabilities. Thus, the fastest mixing symmetric Markov chain problem can be expressed as our general problem (2), with $\mathcal{W} = \{w \mid w \geq 0, \ \mathbf{diag}(L) \leq \mathbf{1}\}$, a polyhedron.

The semidefinite programming formulation of the problem is

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & -\gamma I \preceq I - L - (1/n)\mathbf{1}\mathbf{1}^T \preceq \gamma I, \quad w \geq 0, \quad \mathbf{diag}(L) \leq \mathbf{1}, \end{array}$$

with variables $w \in \mathbf{R}^m$ and $\gamma \in \mathbf{R}$.

Since the fastest mixing symmetric Markov chain problem is convex, indeed, equivalent to an SDP, it can be solved effectively. Generic methods can be used for problems with only a few thousand edges; far larger problems, with millions of edges, can be solved using subgradient optimization techniques, exploiting Lanczos methods to efficiently compute a few extreme eigenvalues and eigenvectors of $I - L - (1/n)\mathbf{1}\mathbf{1}^T$; see [5].

The optimal transition probabilities can be quite interesting; for example, a graph can have many edges with optimal transition probability zero. This means (roughly) that those edges are not needed to achieve fastest mixing on the given graph. We also note that the optimal transition probabilities can yield a mixing rate that is unboundedly better than some simple standard schemes for assigning transition probabilities for fast mixing, such as the maximum-degree method, or the Metropolis-Hastings method [5].

Standard methods can be used to construct various dual problems for the fastest mixing symmetric Markov chain problem. One such dual is

$$\begin{array}{ll} \text{maximize} & \mathbf{1}^T z \\ \text{subject to} & Y\mathbf{1} = 0, \quad Y = Y^T, \quad \|Y\|_* \leq 1 \\ & (z_i + z_j)/2 \leq Y_{ij}, \quad (i,j) \in E, \end{array} \tag{5}$$

with variables $z \in \mathbf{R}^n$ and $Y \in \mathbf{R}^{n \times n}$. Here $\|Y\|_* = \sum_{i=1}^n |\lambda_i(Y)|$, the sum of the singular values of Y, which is the dual norm of the spectral norm. This dual problem is convex, since the objective, which is maximized, is linear, hence concave, and the constraints are all convex. We have the following:

- *Weak duality.* If Y, z are feasible for the dual problem (5), then we have $\mathbf{1}^T z \leq \mu^\star$, where $\mu^\star$ is the optimal value of the fastest mixing symmetric Markov chain problem.

- *Strong duality.* There exist $Y^\star$, $z^\star$ that are optimal for the dual problem, and satisfy $\mathbf{1}^T z^\star = \mu^\star$. This means that the optimal values of the primal and dual problems are the same, and that the dual problem yields a sharp lower bound on the optimal SLEM.

Both of these conclusions follow from general results for convex optimization problems (see, *e.g.*, [10, 1, 6]). We can conclude strong duality using (a refined form of) Slater's condition (see, *e.g.*, [1, §3.3] and [6, §5.2]), since the constraints are all linear equalities and inequalities.

3. Fastest mixing Markov process

Here we briefly describe the problem of finding the fastest mixing continuous-time symmetric Markov process on a given graph [11].

Consider a continuous-time Markov process on the graph G, with transition rate (or intensity) w_l across edge l. The probability density $\pi(t) \in \mathbf{R}^{1\times n}$ at time $t \geq 0$ is given by $\pi(t) = \pi(0)e^{-tL}$. It follows that the asymptotic rate of convergence to the uniform distribution is governed by λ_2, the smallest positive eigenvalue of the Laplacian matrix. The deviation from uniform distribution decays, in the worst case, as $e^{-\lambda_2 t}$. We can express λ_2 as

$$\lambda_2 = \min\{\lambda_2, \ldots, \lambda_n\},$$

which has the standard form (1), with $\phi(u_1, \ldots, u_{n-1}) = \min_{i=1}^{n-1} u_i$. Since the minimum function is concave, we see that λ_2 is a concave function of the edge weights w. It is evidently homogeneous in w, so to get a sensible problem we must normalize the weights in some way, for example, as $\mathbf{1}^T w = 1$.

To find the transition rates that give fastest convergence (among weights that sum to one), we pose the problem

$$\begin{array}{ll} \text{maximize} & \lambda_2 \\ \text{subject to} & w \geq 0, \quad \mathbf{1}^T w = 1, \end{array}$$

with variable $w \in \mathbf{R}^m$. This is a convex optimization problem, which can be formulated as the SDP

$$\begin{array}{ll} \text{maximize} & \gamma \\ \text{subject to} & \gamma I \preceq L + \beta \mathbf{1}\mathbf{1}^T, \quad w \geq 0, \quad \mathbf{1}^T w = 1, \end{array}$$

with variables $\gamma, \beta \in \mathbf{R}$, $w \in \mathbf{R}^m$.

The same problem, allocating a fixed total edge weight across the graph edges so as to maximize the smallest positive Laplacian eigenvalue, arises in other areas. For example, λ_2 arises in graph theory, and is called the *algebraic connectivity* of the graph. Fiedler refers to the maximum value of λ_2 that can be obtained by allocating a fixed total weight to the edges of a graph, as its *absolute algebraic connectivity* [7].

The dual of the fastest mixing Markov process problem can be given a very interesting interpretation. It is equivalent to the following problem. We are given some distances $d_1, \ldots, d_m$ on the graph edges. The goal is find a configuration of points $x_1, \ldots, x_n \in \mathbf{R}^n$ that satisfy $\|x_i - x_j\|_2 \leq d_l$, whenever edge l connects vertices i and j, and in addition maximizes the total variance, given by $\sum_{i\neq j} \|x_i - x_j\|^2$. This problem was recently formulated in the machine learning literature as a method for identifying low dimensional structure in data; see, *e.g.*, [12].

4. Minimum total effective resistance

Here we describe the problem of choosing the edge weights to minimize the total effective resistance of a graph, subject to some given total weight [8]. We consider the graph as an electrical circuit or network, with the edge weight representing the conductance (inverse of resistance) of the associated electrical branch. We define R_{ij} as the resistance in the network seen between nodes i and j. The total effective resistance is defined as $R = \sum_{i<j} R_{ij}$.

The total effective resistance comes up in several applications beyond circuit theory. For example, it is proportional to the average commute time, over all pairs of vertices, in the random walk on the graph defined by the weights w_l [8]. (The probability of a transition from vertex i to vertex j is w_l, the associated edge weight, divided by the total weight of all edges adjacent to vertex i.)

It can be shown that

$$R = \frac{1}{n} \sum_{i=2}^{n} 1/\lambda_i,$$

i.e., it is proportional to the sum of the inverses of the positive Laplacian eigenvalues. This follows our general form (1), with $\phi(u_1, \ldots, u_{n-1}) = \sum_{i=1}^{n-1} 1/u_i$, with domain $\mathbf{R}^{n-1}_{++}$. ($\mathbf{R}_{++}$ is the set of positive reals.) In particular, the total effective resistance is a convex function of the weight vector w. Minimizing total effective resistance, subject to $w \geq 0$ and $\mathbf{1}^T w = 1$, is thus a convex optimization problem.

The problem can be formulated as the SDP

$$\begin{array}{ll} \text{minimize} & n \, \mathbf{Tr}\, Y \\ \text{subject to} & \mathbf{1}^T w = 1, \quad w \geq 0, \\ & \left[\begin{array}{cc} L + (1/n)\mathbf{1}\mathbf{1}^T & I \\ I & Y \end{array} \right] \succeq 0, \end{array}$$

with variables $w \in \mathbf{R}^m$ and the (slack) matrix $Y = Y^T \in \mathbf{R}^{n \times n}$ (see [8]).

5. Fast averaging

Here we describe the problem of choosing edge weights that give fastest averaging, using a classical linear iteration [13]. The nodes start with value $x(0) \in \mathbf{R}^n$, and at each iteration we update the node values as $x(t+1) = (I - L)x(t)$. The goal is to choose the edge weights so that $x_i(t)$ converges, as rapidly as possible, to the average value, *i.e.*, $x(t) \to (1/n)\mathbf{1}\mathbf{1}^T x(0)$.

This iteration can be given a very simple interpretation. At each step, we replace each node value with a weighted average of its previous value and its neighbors' previous values. The weights used to form the average are taken from the graph edge weights, with the self-weight chosen so that the sum of the adjacent edge weights, plus the self-weight, equals one. The weights used to carry out this local averaging sum to one at each node, but can be negative.

When the weights are symmetric (which we assume here), the convergence rate of this averaging process is determined by the SLEM of $I - L$, *i.e.*, (4), exactly as in the Markov chain problem. The difference here is that the weights can be negative; in the Markov chain, of course, the weights (transition probabilities) must be nonnegative. The optimal weights can be found by solving the unconstrained problem

$$\text{minimize} \quad \max_{i=2}^{n} |1 - \lambda_i|,$$

which evidently is a convex optimization problem. It can be posed as the SDP

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & -\gamma I \preceq I - L - (1/n)\mathbf{1}\mathbf{1}^T \preceq \gamma I, \end{array}$$

with variables $\gamma \in \mathbf{R}$, $w \in \mathbf{R}^m$. Without loss of generality, we can assume that $L \succeq 0$. The problem is the same as the fastest mixing symmetric Markov chain problem, but without the nonnegativity requirement on w. It often happens that some of the optimal weights are negative [13].

6. Minimum RMS consensus error

Here we describe a variation on the fastest linear averaging problem described above, in which an additive random noise perturbs the node values [15]. The iteration is $x(t+1) = (I - L)x(t) + v(t)$, where $v(t) \in \mathbf{R}^n$ are uncorrelated zero mean unit variance random variables, *i.e.*,

$$\mathbf{E}\, v(t) = 0, \qquad \mathbf{E}\, v(t)v(t)^T = I, \qquad \mathbf{E}\, v(t)v(s)^T = 0, \quad t \neq s.$$

This iteration arises in noisy averaging, distributed data fusion, and load balancing applications; see the references in [15].

We can measure the effectiveness of the averaging iteration at countering the effects of the additive noises by the steady-state mean-square deviation, defined as

$$\delta_{\text{ss}} = \lim_{t\to\infty} \mathbf{E} \left(\frac{1}{n} \sum_{i<j} (x_i(t) - x_j(t))^2 \right).$$

The steady-state mean-square deviation can be expressed as

$$\delta_{\text{ss}} = \sum_{i=2}^{n} \frac{1}{\lambda_i(2 - \lambda_i)},$$

provided $0 < \lambda_i < 2$ for $i = 2, \ldots, n$, and is infinite otherwise. (The condition $0 < \lambda_i < 2$ for $i = 2, \ldots, n$ is the same as $\max\{1 - \lambda_2, \lambda_n - 1\} < 1$, which is the condition that the linear iteration for averaging, without the additive noise, converges.) Once again, this has the standard form (1), with $\phi(u_1, \ldots, u_{n-1}) = \sum_{i=1}^{n-1} 1/(u_i(2 - u_i))$, with domain $(0, 2)^{n-1}$. In particular, we see that δ_{ss} is a convex function of the edge weights. To find the weights that yield the smallest steady-state mean-square deviation, we simply minimize the convex function δ_{ss} over $w \in \mathbf{R}^m$.

7. Methods

All the problems described above can be effectively solved numerically, by a variety of standard methods for convex optimization, including interior-point methods for modest sized problems (with a few thousand weights) and subgradient-based methods for larger problems. We can exploit structure in the problems (such as sparsity of the underlying graph) to increase the efficiency of these methods.

We can also exploit symmetry in solving the problems. Two edges are symmetric if there exists an automorphism of the graph that maps one edge to the other. Whenever two edges are symmetric, we can assume without loss of generality that the corresponding edge weights are equal. (This follows from a basic result in convex optimization: there is always a solution that is invariant under the group of permutations that leave the objective function and constraint set fixed.) If the symmetry group of the graph is large, this can considerably reduce the size of the optimization problem that needs to be solved. As an extreme example, consider an edge-transitive graph, *i.e.*, one in which any two edges are symmetric. For such a graph, we can assume that all edge weights are equal, *i.e.*, there is only one common edge weight to be determined. This reduces the problem to one with at most one scalar variable (the common edge weight); if there is an equality constraint, such as $\mathbf{1}^T w = 1$, we conclude that an optimal solution is given by uniform edge weights, $w = (1/m)\mathbf{1}$ [3]. This idea is used in [9] to reduce some specific weight optimization problems to ones with a handful of variables, which can be solved analytically.

References

[1] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

[2] J. Borwein and A. Lewis. *Convex Analysis and Nonlinear Optimization, Theory and Examples*. Canadian Mathematical Society Books in Mathematics. Springer-Verlag, New York, 2000.

[3] S. Boyd, P. Diaconis, P. Parrilo, and L. Xiao. Symmetry analysis of reversible Markov chains. *Internet Mathematics*, 2(1):31–71, 2005.

[4] S. Boyd, P. Diaconis, J. Sun, and L. Xiao. Fastest mixing Markov chain on a path. *American Mathematical Monthly*, January 2006. To appear.

[5] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov chain on a graph. *SIAM Review*, 46(4):667–689, 2004.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[7] M. Fiedler. Absolute algebraic connectivity of trees. *Linear and Multilinear Algebra*, 26:85–106, 1990.

[8] A. Ghosh and S. Boyd. Minimizing effective resistance of a graph. *SIAM Review*, 2006. To appear. Available at `www.stanford.edu/~boyd/eff_res`.

[9] A. Ghosh and S. Boyd. Upper bounds on algebraic connectivity via convex optimization. *Linear Algebra Appl.*, 2006. To appear. Available at `www.stanford.edu/~boyd/eigbounds_laplacian`.

[10] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[11] J. Sun, S. Boyd, L. Xiao, and P. Diaconis. The fastest mixing Markov process on a graph and a connection to a maximum variance unfolding problem. *SIAM Review*, 2006. To appear. Available at `www/~boyd/fmmp`.

[12] K. Weinberger and L. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04)*, Washington D.C., 2004.

[13] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53:65–78, 2004.

[14] L. Xiao and S. Boyd. Optimal scaling of a gradient method for distributed resource allocation. *Journal of Optimization Theory and Applications*, 129(3), 2006. To appear.

[15] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 2006. To appear. Available at `www/~boyd/lms_consensus`.

Electrical Engineering Department
Stanford University
E-mail: boyd@stanford.edu