Implementing regularization implicitly via approximate eigenvector computation

Michael W. Mahoney[a] and Lorenzo Orecchia[b]

(a) Mathematics Department, Stanford University
(b) Computer Science Division, UC Berkeley

Abstract

Regularization is a powerful technique for extracting useful information from noisy data. Typically, it is implemented by adding some sort of norm constraint to an objective function and then exactly optimizing the modified objective function. This procedure often leads to optimization problems that are computationally more expensive than the original problem, a fact that is clearly problematic if one is interested in large-scale applications. On the other hand, a large body of empirical work has demonstrated that heuristics, and in some cases approximation algorithms, developed to speed up computations sometimes have the side-effect of performing regularization implicitly. Thus, we consider the question: What is the regularized optimization objective that an approximation algorithm is exactly optimizing?

We address this question in the context of computing approximations to the smallest nontrivial eigenvector of a graph Laplacian; and we consider three random-walk-based procedures: one based on the heat kernel of the graph, one based on computing the the PageRank vector associated with the graph, and one based on a truncated lazy random walk. In each case, we provide a precise characterization of the manner in which the approximation method can be viewed as implicitly computing the exact solution to a regularized problem. Interestingly, the regularization is not on the usual vector form of the optimization problem, but instead it is on a related semidefinite program.

1 REGULARIZATION AND IMPLICIT REGULARIZATION

Regularization is a fundamental technique in the study of mathematical optimization. It allows us to take a generic optimization problem and convert it into a related regularized problem that enjoys many desirable properties, such as stability and uniqueness of the optimal solution.

Regularization has applications in statistics and learning, where it is used to improve the level of generalization in supervised learning, to prevent overfitting and to decrease the sensitivity to random noise. Recently, regularization methods have also found their way into the study of combinatorial optimization.

Formal Definition

Initial optimization program

$$\min_{x\in H} L(x)$$

Regularized program

$$\min_{x\in H} L(x) + \lambda \cdot F(x)$$

Regularizer F

Parameter $\lambda > 0$

F is chosen to have special properties (convexity, continuity) that yield the well-behaved features of the regularized problem. Usually, the regularized problem is explicitly stated and solved.

Implicit Regularization

Empirical Observation: Many heuristics and approximation techniques, designed to speed-up computations, seem to have regularizing effects. Important examples of this phenomenon are early stopping in gradient descent (e.g. in the training of neural networks) , and binning in image processing.

NB: This regularization is implicit, no optimization problem is explicitly solved and the regularizer is unknown.

Main Question:

Can such approximate computation procedures be seen as solving explicit regularized problems?

Specific setting: Computation of first non-trivial eigenvector of a graph.

2 GRAPH MATRICES

We consider an undirected weighted graph G=(V,E,w), where edge $\{i,j\} \in E$ has weight w_{ij}. In the study of Spectral Graph Theory, different matrices in $R^{V\times V}$ are associated with graph G. We denote by D the diagonal matrix of degrees of G and by A the adjacency matrix of G. The following are two fundamental graph matrices:

Natural Random Walk Matrix $W = A\,D^{-1}$

Laplacian Matrix $L = D - A$

The Natural Random Walk Matrix is the probability transition matrix of the natural random walk over G, i.e. the random walk that, in one step from vertex v, picks a neighbor u of v with probability proportionally to the weight of w_{uv} and moves to that vertex.

Connection between W and L

The Random Walk W has a stationary distribution $\pi \propto D1$, uniform over the edges.
The first (smallest) eigenvector of L is the constant eigenvector 1 with eigenvalue 0.
Given vector x such that $x^T D 1 = 0$

MIXING

$$Dx \longrightarrow W(Dx) \longrightarrow W^2(Dx) \longrightarrow \cdots \longrightarrow W^t(Dx) \longrightarrow 0$$

The rate of convergence or mixing is determined by the quadratic form $x^T Lx$

FAST MIXING — SLOW MIXING

$x^T Lx$ large — $x^T Lx$ small

The second eigenvalue λ_2 of the Laplacian of G and its eigenvector x^* describe the most slowly mixing unit vector, i.e. the unit vector that is the slowest to converge to 0 under the application of the random walk W.

Computation of x^*

The eigenvector can be computed up to an arbitrary degree of precision by simulating the limit process

$$x^* = \lim_{t\to\infty} \frac{D^{-1}W^t y_0}{||W^t y_0||_{D^{-1}}}$$

for random y_0 such that $y_0^T D^{-1} 1 = 0$

3 MOTIVATION: COMMUNITY STRUCTURE IN NETWORKS

In many cases, a network can be modeled as an undirected weighted graph and significant communities can be found by optimizing some notion of community score over all cuts $S \subseteq V$. A common community score is conductance:

Conductance Score $\quad \phi(S) = \frac{w(S,\bar{S})}{\min\{\mathrm{vol}(S),\mathrm{vol}(\bar{S})\}}.$

In words, the conductance of a cut is the ratio of the weight of the edges that cross the cut over the total weight of the edges adjacent to the smaller side of the cut. Optimizing conductance is NP-hard, but the eigenvector x^* can be used to obtain a good approximation in nearly-linear time.

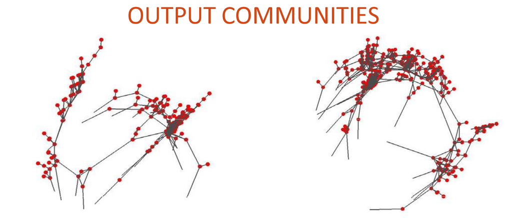
Implicit Regularization in Community Detection

Our study is motivated by an observation made by Lang, Leskovec and Mahoney, who evaluate the performance of different algorithms in identifying communities in information networks. In particular, they compared the use of eigenvector computation and other algorithms optimizing conductance against the use of random walks that are stopped before they converge to the first non-trivial eigenvector.

EIGENVECTOR COMPUTATION:

$$\lim_{t\to\infty} \frac{D^{-1}W^t y_0}{||W^t y_0||_{D^{-1}}}$$

OUTPUT COMMUNITIES

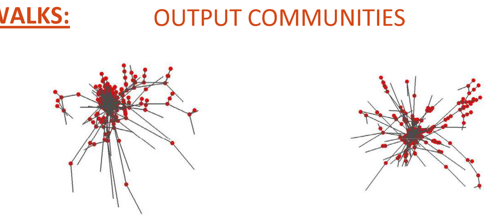


LOW CONDUCTANCE, BUT OFTEN DISJOINT, ELONGATED, SENSITIVE TO NOISE

APPROXIMATE EIGENVECTOR COMPUTATION BY RANDOM WALKS:

Finite, smaller t $\quad \frac{D^{-1}W^t y_0}{||W^t y_0||_{D^{-1}}}$

OUTPUT COMMUNITIES



HIGHER CONDUCTANCE, BUT SMOOTHER, STABLER CLUSTERS

RANDOM WALKS IMPLICITLY REGULARIZE EIGENVECTOR COMPUTATION

4 THREE FUNDAMENTAL RANDOM WALKS

Different random walks can be used to obtain different approximations of the eigenvector x^*, yielding different regularization properties. In this work, we focus on the three random walk processes that appear most prominently in the study of community detection and graph partitioning.

Random walks considered in our work

DISTRIBUTION OF NUMBER OF STEPS

- Heat Kernel random walk with parameter t

$$H_t = e^{-tL} = e^{-t}\sum_{i=1}^{\infty} \frac{t^i}{i!} W^i \qquad \text{Poisson } (t)$$

- Personalized PageRank random walk with teleportation α

$$R_\alpha = \alpha \sum_{i=0}^{\infty} (1-\alpha)^i W^i \qquad \text{Geometric } (\alpha)$$

- Truncated lazy random walk with staying probability p and number of steps t

$$T_{p,t} = (pI + (1-p)W)^t \qquad \text{Binomial } (t,\, p)$$

Our Result

In this work, we formulate a regularized version of the spectral optimization program defining the first non-trivial eigenvector and show that three common choices of regularizers yield the three random walks as optimal solutions.

5 REGULARIZED SPECTRAL OPTIMIZATION

In this presentation, we assume that the graph G is d-regular. Our arguments are easily extendable to general graphs. Under this condition, the eigenvector x^* can be characterized as the optimal solution to a simple quadratic optimization program. To obtain a regularized version of this program, we proceed in two steps. As a first step, we construct a semidefinite program (SDP) that is equivalent to the original program.

Original Program

$$\begin{aligned} \tfrac{1}{d}\ \min\ & x^T Lx \\ \text{s.t.}\ & ||x||_2 = 1 \\ & x^T 1 = 0 \end{aligned}$$

Equivalent by taking $X = x^*(x^*)^T$

SDP Formulation

$$\begin{aligned} \tfrac{1}{d}\ \min\ & L \bullet X \\ \text{s.t.}\ & I \bullet X = 1 \\ & J \bullet X = 0 \\ & X \succeq 0 \end{aligned}$$

Density Matrix

A feasible X for this SDP is a density matrix (i.e. a positive semidefinite matrix with trace 1). Hence, the eigenvector decomposition of X must satisfy the following constraints:

$$X = \sum p_i v_i v_i^T \quad \text{such that} \quad \begin{cases} \forall i, p_i \geq 0, \\ \sum p_i = 1, \\ \forall i, v_i^T 1 = 0. \end{cases}$$

Probability Distribution

Because of the equivalence above, for this SDP formulation the optimal solution is just $X = x^*(x^*)^T$ and the distribution over eigenvectors is trivial with all the weight concentrated on x^*. To obtain a regularized program, we introduce a regularizer term that will force the distribution of eigenvalues of X to be closer to uniform.

Regularized SDP

$$\begin{aligned} \tfrac{1}{d}\ \min\ & L \bullet X + \boxed{\eta F(X)} \\ \text{s.t.}\ & I \bullet X = 1;\quad J \bullet X = 0 \\ & X \succeq 0 \end{aligned}$$

6 THREE REGULARIZERS AND MAIN THEOREM

To constrain the distribution $\{p_i\}$ of eigenvalues of X to be less concentrated, we use convex regularizers that, as a function of X, are unitarily invariant (i.e. depend only on $\{p_i\}$) and are minimized when $\{p_i\}$ is uniform, i.e. when X $\propto$ I. Moreover, the regularizers that we consider are classical regularizers that have been used in many learning applications.

Regularizers considered in our work

- von Neumann Entropy

$$F_H(X) = -S(X) = \mathrm{Tr}(X \log X) = \sum p_i \log p_i$$

- Log Determinant

$$F_D(X) = -\log\det(X) = -\sum \log p_i$$

- p-Norm

$$F_p(X) = \tfrac{1}{p}||X||_p^p = \tfrac{1}{p}\mathrm{Tr}(X^p) = \tfrac{1}{p}\sum p_i^p$$

Main Theorem

REGULARIZER	OPTIMAL SOLUTION OF REGULARIZED PROGRAM	
$F = F_H$	$X^\star \propto H_t$	where t depends on parameter η
$F = F_D$	$X^\star \propto R_\alpha$	where α depends on parameter η
$F = F_p$	$X^\star \propto T_{q,\frac{1}{p-1}}$	where η depends on parameter η

7 DISCUSSION: REGULARIZATION AND LOCALIZATION

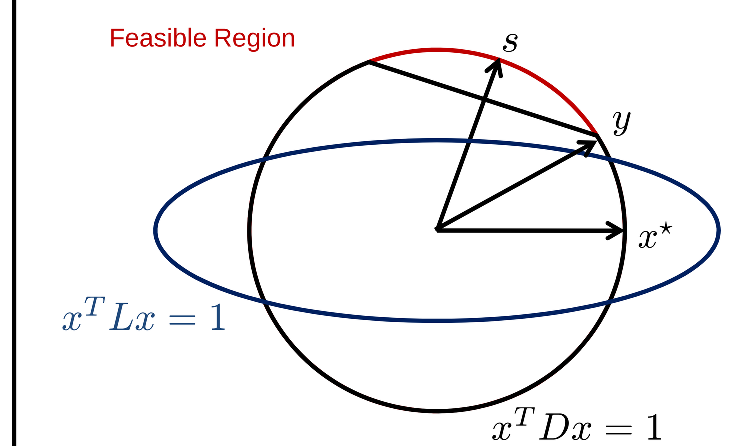
The main departure between our work and a standard regularization argument is the fact that our regularized program does not yield a vector, but a density matrix, which represents a probability distribution over vectors. Notice that it is possible to obtain a vector from a density matrix X by sampling its eigenvectors according to the probabilities given by the eigenvalues or, more simply, by multiplying the square root of X with a standard Gaussian random vector. In either cases, however, we do not have an optimization characterization of the resulting vector.

OPEN QUESTION: Given the transition matrix P of one of the three random walk processes under consideration and a seed vector s such that $s^T D^{-1} 1 = 0$, can we characterize the vector

$$y = D^{-1}Ps$$

as the solution of a regularized version of the spectral optimization problem?

PARTIAL SOLUTION: It is possible for Personalized PageRank. In recent work with Nisheeth Vishnoi, we modify the original spectral program by adding a localization constraints forcing vectors to belong to a spherical cap centered at s. We show that the optimal solutions of the resulting program are Personalized PageRanks of s.



Localized Program

$$\begin{aligned} \min\ & x^T Lx \\ \text{s.t.}\ & x^T Dx = 1 \\ & x^T D1 = 0 \\ & x^T Ds \geq \kappa \end{aligned}$$

Global spectral problem

Additional local constraint

THEOREM:

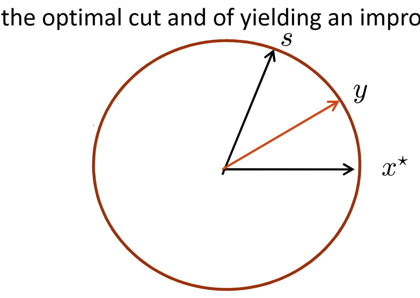
For every $\alpha \in (0,1)$, there exists a κ such that the optimal solution to the Localized Program is a scaling of R_α.

8 DISCUSSION: APPLICATIONS TO GRAPH PARTITIONING

Recently, the regularization of the eigenvector computation by using random walks has found application in combinatorial optimization in the design of improved algorithms for different graph partitioning problems, such as finding the cut of minimum conductance and finding the balanced cut of minimum conductance.

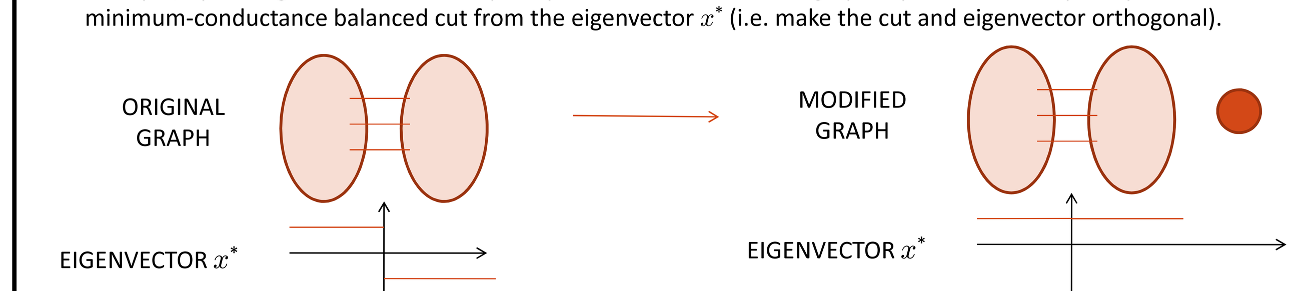
Minimum Conductance

The eigenvector x^* may be poorly correlated with the optimal cut v and may only yield a weak approximation. Replacing the eigenvector with a vector y obtained from a random walk process is helpful in this case, as y still displays slow mixing (i.e. it is correlated with some low conductance cut), but has non-zero probability of having better correlation with the optimal cut and of yielding an improved approximation.



Balanced Cut of Minimum Conductance

Regularization by random walk computation is particularly useful in the study of balanced cuts because the sensitivity of the first non-trivial eigenvector makes it a poor tool to detect low-conductance balanced cuts. For example, by adding a small number of poorly connected vertices to a graph is possible to completely hide the minimum-conductance balanced cut from the eigenvector x^* (i.e. make the cut and eigenvector orthogonal).



Because of its regularization properties, the random walk approach is more stable and more successful at eliminating the noise introduced by unbalanced cuts of low conductance.