Max–Planck–Institut für biologische Kybernetik

Max Planck Institute for Biological Cybernetics

Technical Report No. 144

A Combinatorial View of Graph Laplacians

Jiayuan Huang[1,2]

August 2005

[1] Department Schölkopf, email: huang@tuebingen.mpg.de
[2] University of Waterloo, email: j9huang@cs.uwaterloo.ca

This report is available in PDF–format via anonymous ftp at ftp://ftp.kyb.tuebingen.mpg.de/pub/mpi-memos/pdf/comview.pdf. The complete series of Technical Reports is documented at: http://www.kyb.tuebingen.mpg.de/techreports.html

A Combinatorial View of Graph Laplacians

Jiayuan Huang

Abstract. Discussions about different graph Laplacians—mainly the normalized and unnormalized versions of graph Laplacian—have been ardent with respect to various methods of clustering and graph based semi-supervised learning. Previous research in the graph Laplacians, from a continuous perspective, investigated the convergence properties of the Laplacian operators on Riemannian Manifolds. In this paper, we analyze different variants of graph Laplacians directly by solving the original NP hard graph partitioning problems that provides a combinatorial point of view of graph Laplacians. The spectral solutions provide evidence that the normalized Laplacian encodes more reasonable considerations for graph partitioning. We also explain the direct relationship between spectral clustering and graph-based semi-supervised learning. We provide experiments comparing the results of using different graph Laplacians.

1 Introduction

In recent decades, methods of spectral clustering and graph based semi-supervised learning involved with graph Laplacians have been attractive and prospective in the area of machine learning. Example include, ratio cut clustering [1], normalized cut spectral clustering [2], transductive inference with smoothing over graphs [3, 4, 5]. These methods implicitly take advantage of the connection between the Laplacian spectra and properties of the associated graphs for learning problems. It is known that graph Laplacians can be thought of as discrete analogue of Laplacian-Beltrami operators on Riemannian manifolds. One may refer to [6, 7, 8] about the relationship between graph Laplacians and Laplacian-Beltrami operators.

We propose analyzing graphing Laplacians from a combinatorial point of view, understand them from a discrete perspective. This is the opposite direction of analysis of manifolds and an alternative approach to explain the differences of graph Laplacians. We think it would be helpful to understand many current methods of clustering and graph-based semi-supervised learning that use different graph Laplacians. In this paper, we directly examine a discrete graph. What does the graph's spectrum of the associated partition really mean? In fact, the study of the connection between Laplacian spectra and associated graph connectivity dates back to Fielder's work in the 1970s [9]. But still we do not know what way of partitioning the graph will result in the best performance. As we know, the original graph partitioning problem is a combinatorial optimization problem. Here, we would like to explain difference in graph partitioning from a combinatorial point of view. Graph Laplacians are naturally recovered from the relaxed solutions to an original combinatorial problem. Actually, it is quite straightforward to go from clustering to graph based transductive learning. In this paper, we will also examine difference in graph based transductive methods with respect to spectral clustering methods.

2 Preliminaries

2.1 Definition

Let $G(V,E)$ be a finite, connected, undirected graph. G is connected if there is a path linking two of its vertices. The graph is associated with a symmetric weight function $w: E \rightarrow \mathbb{R}^+$. A degree function $d: V \rightarrow \mathbb{R}^+$ is defined to be,

$$d(u) = \sum_{v \sim u} w(u,v), \forall u \in V$$

where $v \sim u$ means v and u are adjacent. The volume of a graph is defined as

$$\operatorname{vol} G = \sum_{v \in V} d(v)$$

2.2 Graph Laplacian

This section introduces different versions of graph Laplacians which are commonly used.

The unnormalized Laplacian (also referred to as combinatorial Laplacian) L is defined as,

$$L(u,v) = \begin{cases} d(u) \text{ if } u = v \\ -w(u,v) \text{ if } u \sim v \\ 0 \text{ otherwise} \end{cases}$$

It would make more sense to take the Laplacian as a linear operator such that for any function $f: V \rightarrow \mathbb{R}$, we have,

$$Lf(u) = \sum_{v \sim u} w(u,v)(f(u) - f(v))$$

In matrix form, L is represented as

$$L = D - W$$

where D is a diagonal matrix where $D(u,u) = d(u)$, and W is the matrix for the weight function where $W(u,v) = w(u,v)$.

L is a positive semidefinite operator, so its eigenvalues are real and non-negative. Obviously, its first eigenvector is $e = (1, 1, ..., 1)$ with the eigenvalue of 0.

The normalized Laplacian Δ is defined as,

$$\Delta(u,v) = \begin{cases} 1 - \frac{w(v,v)}{d(v)} \text{ if } u = v \\ -\frac{w(u,v)}{\sqrt{d(u)d(v)}} \text{ if } u \sim v \\ 0 \text{ otherwise} \end{cases}$$

Also, as a linear operator we have for function f,

$$\Delta f(u) = \frac{1}{\sqrt{d(u)}} \sum_{v \sim u} w(u,v) \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)$$

Note that if $d(u)$ is uniformly distributed, then $\Delta f(u)$ degrades into $Lf(u)$ up to a constant factor $\frac{1}{d(u)}$.

We can write the normalized Laplacian as a matrix as,

$$\Delta = I - D^{-1/2} W D^{-1/2}$$

In general, the following relation holds between L and Δ,

$$\Delta = D^{-1/2} L D^{-1/2}$$

Δ is also semidefinite. However, its first eigenvector is not constant anymore.

3 Graph Partitioning: A Combinatorial Problem

The binary graph partitioning intends to separate a connected simple graph into two parts such that dissimilar vertices are in different clusters and similar vertices are grouped together in the same clusters. We want to obtain an integer assignment function $f : V \to \{-1, 1\}$ that satisfies the objective well. This is mostly related to an NP-hard problem that finds a way of partitioning vertices in two equal subsets with the minimum number of edges cutting across the partition [10]. The problem requires an exponential search time for finding the exact optimum. For a long time, people have been devoted to developing heuristics for the problem. Then, spectral partitioning methods emerged as effective approaches to solve the partitioning problem. The spectral graph partitioning methods relaxed the combinatorial problems into real valued problems. We introduce some typical spectral methods and analysis graph Laplacians with respect to their solutions.

3.1 Definitions

A vertex partition on a graph separates the graph into two disjoint vertex sets S and S^c, where S^c is the compliment of S. It does this by removing edges connecting the two sets. Let $\Pi(S, S^c)$ represent a partition of the vertices of a graph G into two sets S and S^c. The *edge cut*$\mathrm{cut}(S, S^c)$ between S and S^c in an undirected graph is defined as

$$\mathrm{cut}(S, S^c) = \sum_{u \in S, v \in S^c} w(u, v)$$

The quantity of $\mathrm{cut}(S, S^c)$ across $\Pi(S, S^c)$ is also a measure of *association*. Basically, we want the edge cut to be as small as possible to reduce the association between S and S^c.

The *out-boundary* ∂S of S is defined to be $\partial S = \{(u, v) | u \in S, v \in S^c\}$.

We define $\mathrm{vol}\, S$, the volume of S, to be the sum of the degree of the vertices in S:

$$\mathrm{vol}\, S = \sum_{u \in S} d(u)$$

Obviously, the volume of ∂S measures the same quantity of association and edge cut.

3.2 Spectral Methods

There are various partitioning objectives we want to optimize. The graph Laplacians are naturally recovered from the relaxed solutions of the combinatorial problems.

1. Minimum cut

   The objective is to minimize the weighted edge connections between S and S^c. Since $\mathrm{cut}(S, S^c) = \mathrm{cut}(S^c, S)$, we can only minimizing $\mathrm{cut}(S, S^c)$.

   $$\min \mathrm{cut}(S, S^c)$$

   Let f be an indicator function with$f(u) = 1 \forall u \in S$ and $f(u) = -1 \forall u \in S^c$. For $(u, v) \in E, u \in S, v \in S^c$, we have $(f(u) - f(v))^2 = 4$ and $(f(u) - f(v))^2 = 0$ if u, v are both in S or S^c. Therefore, the objective equals,

   $$\min \frac{1}{4} \sum_{u \in S, v \in S^c} w(u, v)(f(u) - f(v))^2 = \frac{1}{8} \sum_{u \sim v} w(u, v)(f(u) - f(v))^2$$

The equation results from the fact that we count the cut twice for each edge. Expand the right-hand side,

$$\begin{aligned}
& \frac{1}{8}\sum_{u\sim v} w(u,v)(f(u)-f(v))^2 \\
= \quad & \frac{1}{8}\sum_{u\sim v} w(u,v)(f(u)^2+f(v)^2-2f(u)f(v)) \\
= \quad & \frac{1}{4}f(u)^2 d(u) - \frac{1}{2}\sum_{u\sim v} f(u)w(u,v)f(v) \\
= \quad & \frac{1}{4}f^T(D-W)f = \frac{1}{4}f^T Lf
\end{aligned}$$

Without any constraint, the solution for this can be very arbitrary. We can cut the vertex which has minimum sum of weight connections to all other vertices. Unfortunately, this causes an ill-conditioned problem: it produces a quite unbalanced partition that the result tends to group small number of isolated vertices. This is because the cut criterion does not add any constraint within each partition [2]. To overcome this problem, some proper constraints are proposed to balance the partition on graphs.

2. Ratio-cut [1]

   After the min-cut method was proposed, people tried to constraint partition size to obtain balanced partitions. However, this leads to a NP-complete problem. Many heuristics were proposed to solve this problem. The ratio-cut method is an important step that it incorporates the balance of partition sizes in the cut criterion rather than imposing constraints explicitly.

   This method attempts to minimize the cut cost while implicitly preserving the cardinality of partitions at the same time. The cut criterion is defined as,

$$\mathrm{Rcut}(S,S^c) = \frac{\mathrm{cut}(S,S^c)}{|S|} + \frac{\mathrm{cut}(S^c,S)}{|S^c|}$$

   where $|S|$ is the number of vertices in S. Let the total number of vertices in G be $|G|$ where $|G| = \sum_u f^2(u)$. Let α denote the ratio $|S^c|/|G|$. The Ratio Cut criterion can be further written as,

$$\mathrm{Rcut}(S,S^c) = \frac{\sum_{u\sim v} w(u,v)(f(u)-f(v))^2}{8\alpha(1-\alpha)\sum_u f^2(u)}$$

   Define another function g as,

$$g(u) = \begin{cases} 2(1-\alpha) > 0, u \in S \\ -2\alpha < 0, u \in S^c \end{cases}$$

   therefore, we have

$$\sum_u g^2(u) = 4\alpha(1-\alpha)\sum_u f^2(u)$$

   and

$$g(u) - g(v) = f(u) - f(v)$$

   thus,

$$\mathrm{Rcut}(S,S^c) = \frac{\sum_{u\sim v} w(u,v)(g(u)-g(v))^2}{2\sum_u g^2(u)} = \frac{g^T Lg}{g^T g}$$

If we drop the condition that f must be either 1 or -1, then by relaxing the combinatorial problem we obtain the optimum solution of the Rayleigh Quotient which leads to ,

$$\min g^T Lg \quad \text{s.t.} \quad \parallel f \parallel = 1, f^T \mathbf{1} = \mathbf{0}; \tag{1}$$

where $\mathbf{1}$ is a column vector with all elements equal 1. The solution is the second eigenvector that satisfies,

$$Lg = \lambda g$$

where $L = D - W$ which is the unnormalized Laplacian.

The unnormalized graph Laplacian is recovered by relaxing the solution for Ratio-Cut. Ratio-cut balances the two partitions by adding the implicitly constraint that the number of vertices should be approximately equal in cut criterion. This is a significant improvement from previous heuristics. However, we should note that this constraint is not reasonable if the weights on edges are not uniformly distributed. We will demostrate this pitfall in our later experiment comparisons.

3. Normalized min-cut [2]

In this method, the proposed cut cost function is

$$\text{Ncut}(S, S^c) = \frac{\text{cut}(S, S^c)}{\text{vol}\, S} + \frac{\text{cut}(S^c, S)}{\text{vol}\, S^c}$$

The intuition is that we want not only to minimize the edge cut between the two sets, but also to keep the size of the two sets as big as possible so that the cut is no trivial. By "size" we mean the total edge weight in a partition. This is different from the Ratio-Cut which only considers the total number of vertices.

The problem is a combinatorial optimization problem. It can be solved by relaxing the integer constraints. Define the same indicator function $h \in V$. Let γ denote the ratio $\text{vol}\, S / \text{vol}\, V$. Since for undirected graphs $\text{cut}(S, S^c) = \text{cut}(S^c, S)$, the cut criterion is written as,

$$\text{Ncut}(S, S^c) = \frac{\sum_{(u,v)\in E} w(u,v) \left(h(u) - h(v)\right)^2}{8\gamma(1-\gamma) \sum_{v \in V} h^2(v) d(v)}$$

Define an indicator function g with $g(v) = 2(1-\gamma)$ if $v \in S$ and -2γ if $v \in S^c$. Clearly for all $u, v \in V$, sign $g(v)$=sign $h(v)$ and $g(u) - g(v) = h(u) - h(v)$. It's easy to see that $\sum_{v \in V} g(v) d(v) = 0$ and $\sum_{v \in V} g^2(v) d(v) = 0$. Therefore,

$$\begin{aligned} \text{Ncut}(S, S^c) &= \frac{\sum_{(u,v)\in E} w(u,v) \left(g(u) - g(v)\right)^2}{2 \sum_{v \in V} g^2(v) d(v)} \\ &= \frac{\sum_{(u,v)\in E} w(u,v) \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(u)}} \right)^2}{2 \sum_{v \in V} f^2(v)} \\ &= \frac{f^T \Delta f}{f^T f} \end{aligned}$$

where $f = \sqrt{d} g$.

If we allow to relax the value of f to be real values, then the problem becomes,

$$\min f^T \Delta f \quad \text{s.t.} \quad \parallel f \parallel = 1, < f, \sqrt{d} >= 0 \tag{2}$$

Clearly, the solution f is the second smallest eigenvector of the normalized Laplacian.

Here we recover the normalized graph Laplacian by solving the relaxed optimization problem with a normalized cut criterion which implicitly constrains the volume of partition.

4 Why We Prefer the Normalized Graph Laplacian

Having understood the spectral methods in graph partitioning, a key observation is that we obtain different graph Laplacians by defining different cut criterions. It is not hard to see that the normalized cut criterion is more preferable in the sense that it balances the sum of edge weights of partitions, not the sum of vertices. The former considers a more general graph in which the graph edge weights are not uniformly distributed. Therefore, simply balancing the vertex number in the two partitions is generally not reasonable. The solution involving the unnormalized graph Laplacian is derived from Ratio-cut which only take balance on number of vertices. The solution involving the normalized graph Laplacian is naturally derived from Normalized cut which more reasonably balances the volume for each class. This explains why the clutering method using normalized graph Laplacian performs better than the one using unnormalized graph Laplacian. The normalized graph Laplacian implicitly takes the weights information into account while the unnormalized one simply ignores this information.

5 From Clustering to Transduction

This understanding of the graph Laplacian based on the original combinatorial problem also helps explain the efficiency of different graph based semi-supervised learning methods in [3, 4, 5]. It is quite straightforward to go from clustering to transductive learning. A general objective for these methods can be written as,

$$\min \phi(f(x), y) + Reg(f) \tag{3}$$

where $\phi(\cdot)$ is a loss function measuring the difference between the estimated function value over labeled vertices and the original label value. For simplicity, ϕ is a square loss function. $Reg(f)$ is a regularization term which involves a graph Laplacian. It has the form of $f^T L_G f$ where L_G is a graph Laplacian. This is exactly the objective in (1) and (2). Further, we can rewrite the objective as,

$$\min Reg(f) \quad \text{s.t.} \quad \phi(f(x_i), y_i) \leq \xi$$

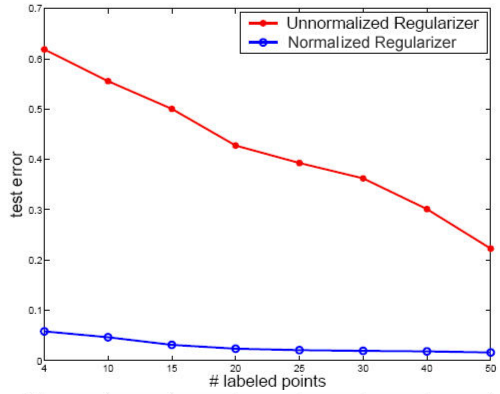
where ξ is a small positive real number.

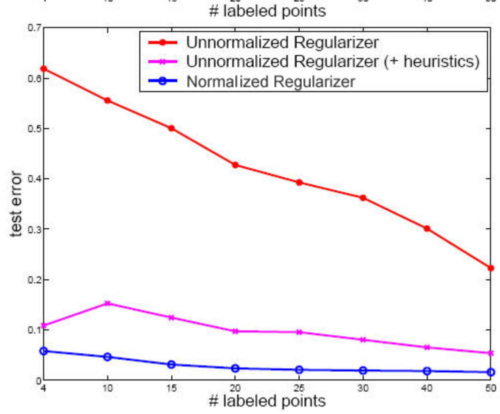
Generally, the difference in methods lies in using different graph Laplacians and whether to fix the labels for the labeled vertices. If we fix the labels, then the constraint becomes $\phi(f(x_i), y_i) = 0$ which equals $f(x_i) = y_i$.

Based on the observations above, the methods used in [4] and [5] amount to constrained clustering with basic minimum cut criterion that explain, in our later experiments, why they require additional heuristic to maintain the class proportions to achieve better performance. The method used in [3] is a constrained clustering with normalized cut criterion. In fact, in semi-supervised learning, since we only have a small number of labeled data, the most important part is how well we can do the clustering. The efficiency relies heavily on the clustering part when applying the framework (3) for semi-supervised learning.

6 Experiment Comparisons

We compare two semi-supervised methods in the framework (3) by adopting the unnormalized regularizer and normalized regularizer. The unnormalized regularizer and the normalized regularizer can be written as $\langle f, Lf \rangle$ and $\langle f, \Delta f \rangle$ separately. We consider a classification task using the USPS handwritten 16×16 digits dataset. We use digits $1, 2, 3$ and 4 in the experiment as the four classes. There are 1269, 929, 824 and 852 examples for each class, for a total of 3874. We construct a fully connected graph by using a RBF kernel





where the width is set to 1.25. We sequentially increase the number of labeled points. The test errors averaged over 100 trails are summerized in the following first figure. It is clear that the method with the normalized regularizer significantly improved the accuracy. The second figure shows the performance of three methods. The heuristic one approximates the class proportions as a prior knowledge that is used in [4]. However, the method with normalized regularizer still has the best performance.

7 Conclusion

We have discussed about different methods for a combinatorial graph partitioning problem. It is easier and clearer to recognize the difference in graph Laplacians by recovering them directly from the relaxed solutions of the partition problem. The combinatorial view serves as a proof which is lacking in the manifold approach. This paper also clarifies some debate on various methods in graph-based semi-supervised learning. Since, so far, the semi-supervised learning methods still heavily rely on clustering, it would be interesting to explore a more proper approach to use the labeled data.

References

[1] L. Hagen and A.B.Kahng. New spectral methods for ratio cut partitioning and clustering. In *IEEE Trans. Comput. Aided Des.*, volume 11, pages 1074–1085, 1992.

[2] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[3] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[4] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proc. 20th Intl. Conf. on Machine Learning*, 2003.

[5] M. Belkin and P. Niyogi. Semi-supervised learning on manifolds. *Machine Learning*, 56(Special Issue on Clustering):209–239, 2004.

[6] J. Audibert M. Hein and U. von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph laplacians. In *Proceedings of the 18th Conference on Learning Theory (COLT)*, pages 470–485, 2005.

[7] U. von Luxburg, O. Bousquet, and M. Belkin. Limits of spectral clustering. In *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

[8] R.R.Coifman B. Nadler, S. Lafon and I. G. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *Advances in Neural Information Processing Systems 18*, 2005.

[9] M. Fiedler. Algebraic connectivity of graphs. In *Czechoslovak Math. J.*, pages 298–305, 1973.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.