A Learning Framework using Green's Function and Kernel Regularization with Application to Recommender System

Chris Ding
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
chqding@lbl.gov

Rong Jin
Department of CSE
Michigan State University
East Lansing, MI 48824
rongjin@cse.msu.edu

Tao Li
School of Computer Science
Florida International University
Miami, FL 33199
taoli@cs.fiu.edu

Horst D Simon
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
hdsimon @lbl.gov

ABSTRACT

Green's function for the Laplace operator represents the propagation of influence of point sources and is the foundation for solving many physics problems. On a graph of pairwise similarities, the Green's function is the inverse of the combinatorial Laplacian; we resolve the zero-mode difficulty by showing its physical origin as the consequence of the Von Neumann boundary condition. We propose to use Green's function to propagate label information for both semi-supervised and unsupervised learning. We also derive this learning framework from the kernel regularization using Reproducing Kernel Hilbert Space theory at strong regularization limit. Green's function provides a well-defined distance metric on a generic weighted graph, either as the effective distance on the network of electric resistors, or the average commute time in random walks. We show that for unsupervised learning this approach is identical to Ratio Cut and Normalized Cut spectral clustering algorithms. Experiments on newsgroups and six UCI datasets illustrate the effectiveness of this approach. Finally, we propose a novel item-based recommender system using Green's function and show its effectiveness.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Algorithms, Experimentation, Performance

Copyright 2007 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
KDD'07, August 12–15, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-609-7/07/0008 ...$5.00

Keywords

Green's Function, Semi-supervised, Label Propagation

1. INTRODUCTION

In semi-supervised learning, we have a large amount of unlabeled data, but only a very small fraction of them are labeled. This situation is common due to ever increasing amount of data being accumulated, and most of them are unlabeled or partially labeled because labeling data requires human skills and extensive labor. The learning task is to classify the unlabeled data based on labeled data.

There are many different approaches to this problem [7, 43]. (a) The classification-based approach, in which a classifier is first trained on the small labeled data and is gradually improved by incorporating unlabeled data. Earlier methods mostly follow this approach [4]. (b) The clustering-based approach, in which a clustering algorithm is used on the whole data (labeled and unlabeled), with the labeled data serve as penalty or regularization or prior information. Recent methods mostly follow this approach, such as spectral clustering based methods [22, 8]. (c) Special mechanisms, such as Gaussian process [26], graph mincut [3], entropy minimization [16], etc.

In this paper, we focus on the label information propagation point of view. Given a dataset with pairwise similarities (W), the semi-supervised learning can be viewed as label propagation from labeled data to unlabeled data. In its simplest form, the label propagation is like a random walk on a similarity-graph W [38]. Using diffusion kernel [25, 37, 24], the semi-supervised learning is like a diffusive process of the labeled information. The harmonic function approach [44] emphasizes the harmonic nature of the diffusive function; and consistency labeling approach [42], emphasizes the spread of label information in an iterative way. Our work is inspired by these prior works, especially by the work of Zhou et al [42].

In physics, diffusion is a process of particle random walk driven by a heat gradient, which emphasizes the local and random nature of the process. We believe, however, label information propagation is more like the field response to the presence of point charges, which emphasizes the global and coherent nature of influence propagation. This response function is the Green's function of the Laplace operator.

In this paper, we formalize the above ideas into a concrete learning algorithm as outlined in §2. We introduce Green's function as the kernel for the Laplace operator in (§3). We resolve the zero-mode problem of the combinatorial Laplacian by showing its physics origin as the Von Neumann boundary condition (§4).

We further justify the Green's function approach in §5 by showing that Green's function is a well-defined similarity metric on a graph, utilizing a well-established (but not widely known) remarkable results on the effective resistance on an electric resistor network, which also can be derived from random walk perspective. In §6, we derive the Green's learning framework independently from the kernel regularization theory of reproducing kernel Hilbert space at strong regularization limit.

In §7, we discuss the unsupervised learning aspects of Green's function approach and show the Green's function approach is equivalent to Ratio Cut and Normalized Cut spectral clustering algorithms. In §8, we explore the relations of Green's function approach with the harmonic function approach. In §9, we present the experimental results on Internet news groups and six UCI datasets. In §10 we propose a novel item-based recommender system using Green's function approach. Finally §11 summarize our most important contribution.

2. GREEN'S FUNCTION LEARNING ALGORITHM

Combinatorial Laplacian

Given a mesh/graph with edge weights W, the *combinatorial Laplacian* is defined to be

$$L = D - W,$$

where the diagonal matrix contains row sums of W: $D = \mathrm{diag}(W\mathbf{e})$, $\mathbf{e} = (1 \cdots 1)^T$.

Green's Functions

We define Green's function for a generic graph as the inverse of $L = D - W$ with zero-mode discarded. (The complete discussion of zero-mode and its physical origin is one of the main contributions of this paper and is discussed in §4).

We construct the Green's function using eigenvectors of L:

$$L\mathbf{v}_k = \lambda_k \mathbf{v}_k, \;\; \mathbf{v}_p^T \mathbf{v}_q = \delta_{pq}. \qquad (1)$$

where $0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ are the eigenvalues. We assume the graph is connected (otherwise we deal with each connected component one at a time). The first eigenvector is a constant vector $\mathbf{v}_1 = \mathbf{e}/n^{1/2}$ with zero eigenvalue and multiplicity one. This zero-mode is discarded (see § 4). The Green's function is then the positive definite part of L

$$G^{(1)} = L_+^{-1} = \frac{1}{(D-W)_+} = \sum_{i=2}^{n} \frac{\mathbf{v}_i \mathbf{v}_i^T}{\lambda_i}. \qquad (2)$$

where $(D-W)_+$ indicates that zero eigen-mode is discarded. Green's function can also be defined on the generalized eigenvectors of the Laplacian matrix:

$$L\mathbf{u}_k = \zeta_k D \mathbf{u}_k, \;\; \mathbf{u}_p^T D \mathbf{u}_q = \mathbf{z}_p^T \mathbf{z}_q = \delta_{pq}. \qquad (3)$$

where $0 = \zeta_1 \le \zeta_2 \le \cdots \le \zeta_n$ are the eigenvalues and the zero-mode is $\mathbf{u}_1 = \mathbf{e}/n^{1/2}$. We have

$$G^{(2)} = \frac{1}{(D-W)_+} = \sum_{k=2}^{n} \frac{\mathbf{u}_k \mathbf{u}_k^T}{\zeta_k}. \qquad (4)$$

(see Eq.19 for derivation). In practice, we truncate the expansion at K terms and store the $K-1$ vectors. G is computed on the fly. So the storage requirement is $O(Kn)$.

Semi-supervised Learning

Suppose we have labeled data $\{\mathbf{x}_i\}_{i=1}^{\ell}, \{y_i\}_{i=1}^{\ell}$. and unlabeled data $\{\mathbf{x}_i\}_{i=\ell+1}^{n}$. The algorithm is a simple influence propagation from labeled data points to unlabeled data points, and can be written as

$$y_j = \mathrm{sign} \sum_{i=1}^{\ell} G_{ji} y_i, \;\; \ell < j \le n. \qquad (5)$$

for 2-class problems ($y_i = \pm 1$), or

$$y_{jk} = \begin{cases} 1 & \text{if} \quad k = \arg\max_k \sum_{i=1}^{\ell} G_{ji} y_{ik}, \\ 0 & \text{otherwise} \end{cases} \quad \ell < j \le n. \qquad (6)$$

for K-class problems where $Y = (\mathbf{y}_1, \cdots, \mathbf{y}_K)$, $Y_{ik} = 1$ if the x_i is a labeled as class k, $Y_{ik} = 0$ otherwise. This algorithm is derived from Eq.(14) and more formally in §6.

Unsupervised Learning

In semi-supervised learning, the influence propagates only once, and propagates only from labeled data to unlabeled data. In unsupervised learning, we let influence propagate from any points to any other points; and repeat multiple times until convergence,

$$h_{jk}^{(t+1)} = \begin{cases} 1 & \text{if} \quad k = \arg\max_k \sum_{i=1}^{n} G_{ji} h_{ik}^{(t)} \\ 0 & \text{otherwise} \end{cases} \quad 1 \le j \le n. \qquad (7)$$

This ensures labels are consistent with influence propagation. Given an initial guess of the labeling for parts or all of the data, we run the above algorithm until convergence. This algorithm is derived in §7.

We often use vector/matrix notation and write Eq.(5) for 2-class semi-supervised learning as

$$\mathbf{y} = \mathrm{sign}\; G\mathbf{y}_0, \qquad (8)$$

Eq.(6) for multi-class semi-supervised learning as

$$Y = \arg\max\; GY_0, \qquad (9)$$

and Eq.(7) for multi-class unsupervised learning as

$$H^{(t+1)} = \arg\max\ GH^{(t)}, \tag{10}$$

where arg max is a row-by-row operation and interpreted as in Eq.(7). For example, $A = \arg\max B$ is done by going through all rows of B, and for each row of B, we select the largest element and set the corresponding element in A as 1 and 0 for the rest of the row.

3. GREEN'S FUNCTION OF THE LAPLACE OPERATOR

We give an introduction to Laplace operator and Green's function.

The Laplace operator

$$\mathcal{L}f(\mathbf{r}) = \nabla^2 f(x,y,z) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) f(x,y,z)$$

describes the most fundamental spatial variations in nature, which determines all major physical phenomenon: heat flow, wave propagation, quantum physics, etc. For example, given the electric charge distribution $\rho(\mathbf{r})$ (a source) and proper boundary condition, the equation $\nabla^2 f(\mathbf{r}) = -4\pi\rho(\mathbf{r})$ governs the scalar electric field $f(\mathbf{r})$, which determines the static and induced charge distributions.

Green's function plays essential role in solving partial differential equations by transforming them to integral equations. Given a linear differential operator $\mathcal{L}$ and *source* function $s(\mathbf{r})$, the differential equation

$$\mathcal{L}f(\mathbf{r}) = s(\mathbf{r}) \tag{11}$$

can be solved by

$$f(\mathbf{r}) = \mathcal{L}^{-1}s(\mathbf{r}) \equiv \int G(\mathbf{r},\mathbf{r}')s(\mathbf{r}')d\mathbf{r}' \tag{12}$$

The kernel $G(\mathbf{r},\mathbf{r}')$ of the integral operator is the Green's function, which captures the field response at $\mathbf{r}$ duo to a single source at $\mathbf{r}'$ represented by $\delta(\mathbf{r}-\mathbf{r}')$:

$$\mathcal{L}G(\mathbf{r},\mathbf{r}') = \delta(\mathbf{r}-\mathbf{r}'). \tag{13}$$

In 3D with open boundary condition, the Green's function for the Laplace operator is the well-known Columb's inverse law: $G(\mathbf{r},\mathbf{r}') = G(\mathbf{r}-\mathbf{r}') = 1/||\mathbf{r}-\mathbf{r}'||$.

Semi-supervised Learning via Green's Function

Suppose we have labeled data $\{\mathbf{x}_i\}_{i=1}^{\ell}, \{y_i\}_{i=1}^{\ell}$ and unlabeled data $\{\mathbf{x}_i\}_{i=\ell+1}^{n}$. Our assumption is that labeled data points are the "electric charges", $\rho(\mathbf{r}) = \sum_{i=1}^{\ell} y_i\delta(\mathbf{r}-\mathbf{x}_i)$. Their influence on unlabeled data point at $\mathbf{r}$ is given by Eq.(12), or,

$$q(\mathbf{r}) = \int G(\mathbf{r},\mathbf{r}')\rho(\mathbf{r}')d\mathbf{r}' = \sum_{i=1}^{\ell} y_i G(\mathbf{r},\mathbf{x}_i). \tag{14}$$

In nature, there are two types charges, the positive and negative charges. This correspond to 2-class problems, which gives Eq.(5) or Eq.(8) in vector-matrix form.

We can generalize this to K types of charges. Different type of charges propagate with the same Green's function. The final charge type of the destination point depends on the competition among different charge types, same as in the positive-negative charge case. This generalization corresponds to the K-class classification problems.

4. ZERO-MODE OF THE COMBINATORIAL LAPLACIAN

The purpose of the detailed discussion of the Laplacian operator is to show the physical origin of the zero-mode of the combinatorial Laplacian $L = D - W$ and why we should discard it in constructing the Green's function of L as in §2.

On a discretized space specified with the weights W of a graph, the Laplacian operator becomes $\nabla^2 f \to -cL\mathbf{f}$ where L is a matrix, $\mathbf{f}$ is a vector defined on the nodes of the graph, and c is a constant depending on the discretization. For a 1D regular grid,

$$\nabla^2 f(x) = \frac{1}{a}[\frac{f(x+a)-f(x)}{a} - \frac{f(x)-f(x-a)}{a}].$$

where $a = \Delta x$ is the spacing between gridpoints. Now the matrix L and c can be inferred from this equation. $c = -1/a^2$. Under discretization, Green's function $G(\mathbf{r},\mathbf{r}')$ becomes a matrix $G_{\mathbf{r},\mathbf{r}'}$. Eq.(13) implies

$$LG = I, \quad \text{or} \quad G = L^{-1}.$$

i.e., Green's function $G(\mathbf{r},\mathbf{r}')$ is the inverse of L. We will see below there are two specific forms of L, the *combinatorial* Laplacian and the *physical* Laplacian.

4.1 Boundary Condition

Physical problems are determined by (1) the differential equation and (2) the boundary condition. The Laplacian matrix L extracted from the Laplacian operator depends on the boundary condition.

Consider a semi-supervised learning problem on a graph, which consists of the interior domain V and the boundary ∂V. Nodes on the boundary are labeled (say $y_i = \pm 1$, $i \in \partial V$). Nodes on the interior domain V are unlabeled. The problem is to determine the labels on unlabeled data V.

Two common boundary conditions are: (1) Dirichlet boundary condition: $f(\mathbf{r}_i) = 0, \forall\ \mathbf{r}_i \in \partial V$. (2) Von Neumann boundary condition: $\partial f(\mathbf{r})/\partial\mathbf{r} = 0, \forall\ \mathbf{r}_i \in \partial V$. The discretized Laplacian operator differs for different boundary conditions. Let weights of the graph can be decomposed as

$$W = \begin{pmatrix} W_{VV} & W_{V\partial V} \\ W_{\partial VV} & W_{\partial V\partial V} \end{pmatrix}, \text{ or } W = \begin{pmatrix} W_{uu} & W_{ul} \\ W_{lu} & W_{ll} \end{pmatrix}. \tag{15}$$

As a main contribution of this paper, we provide the following new results:

Theorem 1. (1) Under the Von Neumann boundary condition, the resulting matrix representing the Laplace operator is the *combinatorial* Laplacian

$$L_c = D_{uu} - W_{uu}, \tag{16}$$

where the diagonal matrix $D_{uu} = \text{diag}(W_{uu}\mathbf{e})$, i.e., $(D_{uu})_{ii} =$ the sum of i-th row in W_{uu}. (2) Under the Dirichlet boundary condition, the resulting matrix representing the Laplace operator is the *physical* Laplacian

$$L_p = D_{uu} + D_{ul} - W_{uu}, \tag{17}$$

where diagonal matrix $D_{ul} = \text{diag}(W_{ul}\mathbf{e})$, i.e., $(D_{ul})_{ii}$ = the sum of i-th row in W_{ul}.

The proof is skipped due to lack of space. An important consequence of Theorem 1(a) is the presence of the zero-mode in the combinatorial Laplacian $L = D - W$. When the derivatives are specified on the boundary, the function value could differ by an overall additive constant. More formally, the solution of the problem is not unique: for any solution $f(\mathbf{x})$, $f(\mathbf{x}) + const$, is also a solution. This gives rise to the zero-mode $\mathbf{e}$ of L : $L\mathbf{e} = 0$. Thus the zero-mode of L being a constant vector is not accidental:

Corollary 1. A consequence of using Von Neumann boundary condition in deriving the combinatorial Laplacian $L = D - W$ is that the zero-mode must be a constant vector.

We note that the operator $\widetilde{L} = I - D^{-1/2}WD^{-1/2}$ is not a physical operator and its zero mode $(d_1, \cdots, d_n)^T$ is not a constant vector. As a consequence, $(\widetilde{L})_+^{-1}$ is not a kernel (see footnote in §6.1), while L_+^{-1} is a kernel (see §5.3).

Due to this zero-mode, strictly speaking, the Green's function of the combinatorial Laplacian $L = D - W$ does not exist. To our knowledge, previous work using Green's function skipped this zero mode without giving a justification or even mentioning it.

By clarifying the situation, we see that the overall constant due to the zero-mode does not affect the final results in influence propagation and thus we can discard this zero-mode in computing L^{-1}.

Another consequence of Theorem 1 is that the physical Laplacian matrix of Theorem 1(b) show show up in the harmonic function approach [44] (see §8). There we do not use von Neumann boundary condition; instead we fixed the boundary to the known labels, which is equivalent to Dirichlet boundary condition. For this reason, by Theorem 1, we expect the physical Laplacian matrix. We summarize this as

Corollary 2. In semi-supervised learning setting, we view data points with known labels as boundary points. This is equivalent to Dirichlet boundary condition and the results of the Laplacian operator approach will involve the physical Laplacian, rather than the combinatorial Laplacian.

Green's Function using generalized Laplacian

The standard approach to Green's function is to use the eigenvectors of Eq.(1). However, we show here that the generalized eigenvectors defined in Eq.(3) is equally suitable to define the inverse of $(D - W)_+$. We rewrite the generalized eigenvalue problem of Eq.(3) as a standard eigenvalue problem

$$\widetilde{W}\mathbf{z} = D^{-1/2}WD^{-1/2}\mathbf{z} = \xi_k \mathbf{z}_k,\ \xi_k = 1 - \zeta_k. \tag{18}$$

From this, we have

$$\begin{aligned} \frac{1}{(D-W)_+} &= D^{-1/2}\frac{1}{(I - D^{-1/2}WD^{-1/2})_+}D^{-1/2} \\ &= D^{-1/2}\sum_{k=2}^{n}\frac{\mathbf{z}_k\mathbf{z}_k^T}{1-\xi_k}D^{-1/2} \end{aligned} \tag{19}$$

Since $D^{-1/2}\mathbf{z} = \mathbf{u}_k$, we obtain Eq.(4).

5. A GENERIC DISTANCE METRIC USING ELECTRIC RESISTOR NETWORK AND RANDOM WALKS

Green's function has a rich content. In this section, we point out that the Green's function relates closely to a well-established (but not widely known) distance metric on a generic weighted graph.[1]

There are two equivalent ways to define this distance: (1) the *effective resistance* distance of a network of electric resistors (2) average number of random walks between two nodes on a graph.

5.1 Electric resistor networks

We view a generic weighted graph as a networks of electric resistors, where the edge connecting nodes i, j is a resistor with resistance r_{ij}. The graph edge weight (the pairwise similarity) between nodes i, j is $w_{ij} = 1/r_{ij}$. (Two nodes not connected by a resistor are viewed as equivalently connected by a resistor with $r_{ij} = \infty$ or $w_{ij} = 0$).

The most common task on a resistor network is to calculate the effective resistance between different nodes. The effective resistance R_{ij} between nodes i, j is equal to 1/(total current between i and j) when i is connected to voltage 1 and j is connected to voltage 0.

Let $G = (D - W)_+^{-1}$ be the Green's function on the graph. A remarkable result established in 1970s[13, 23] is

$$R_{ij} = (\mathbf{e}_i - \mathbf{e}_j)^T G(\mathbf{e}_i - \mathbf{e}_j) = G_{ii} + G_{jj} - 2G_{ij}, \tag{20}$$

where $\mathbf{e}_i$ is a vector of all 0's except an "1" at i-th entry [recall that the distance in a metric space is $d^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T M(\mathbf{x}_i - \mathbf{x}_j)$]. Clearly, we can view R_{ij} as a distance metric on a graph.

5.2 Random Walks

Random walks on a graph is a well studied subject [12, 13]. Given a graph with nonnegative edge weights W, one can do random walk on the graph, with transition probability $t_{ij} = p(i \to j) = w_{ij}/d_i$, or $T = D^{-1}W$.

Consider the average number of hops that a walker commutes from i to j and comes back to i. This is called *average commute time*. It is shown in [6, 32] that this quantity is proportional to R_{ij}. Thus R_{ij} is a distance metric from random walk point of view. This shows the critical role of the Green's function.

[1] In these graphs, the edge weight measures the similarity between the two end-nodes.

5.3 Green's Function as a Kernel

By definition, Green's function is the kernel of the integral operator as in Eq.(12). Here we list some properties. First, G is clearly a semi-positive definite function. Second, any function $\mathbf{f} \in \Re^n$ can be expanded in the basis of G, i.e, $(\mathbf{u}_2, \cdots, \mathbf{u}_n)$ plus a constant $\mathbf{e}/\sqrt{n} = \mathbf{u}_1$.

Third, for a kernel function $\mathcal{K}$, K_{ij} measures the similarity between two objects i, j. Indeed, G_{ij} can be viewed as an effective similarity between nodes i and j from the effective resistance in §5.1 or the average commute time in §5.2. We can see this in the following way. In statistics [28] given pairwise similarity $S = (s_{ij})$, a standard way to convert to distance is $d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$. ¿From Eq.(20), we have $s_{ij} = G_{ij} + const$. Ignore the additive constant, G is the similarity metric that underlie the effective resistor distance metric. We therefore conclude that Green's function is a well-defined similarity function among pairs of nodes. Thus the Green's function is a bona fide kernel.

6. REPRODUCING KERNEL HILBERT SPACE

In this section, we derive the Green's function learning algorithm of §2 from the theory of Reproducing Kernel Hilbert Space (RKHS) at strong regularization limit.

Suppose we have labeled data $\{\mathbf{x}_i\}_{i=1}^n$ with labels $\{y_i\}_{i=1}^n$. We wish to learn the mapping function $f(\mathbf{x})$ such as $\sum_i ||y_i - f(\mathbf{x}_i)||^2$ is minimized. In statistics, we often add a penalty (regularization) term to ensure smoothness of certain quantities such as derivatives. RKHS uses kernel as the regularization term. Let the kernel has the spectral expansion, $\mathcal{K} = \sum_j \gamma_j \mathbf{u}_j \mathbf{u}_j^T$, and define $c_j = <\mathbf{f}, \mathbf{u}_j> = \mathbf{f}^T\mathbf{u}_j$. RKHS find the function $f(\cdot)$ that minimizes

$$J[\mathbf{f}] = \sum_{i=1}^n [y_i - f(\mathbf{x}_i)]^2 + \beta \sum_j \frac{c_j^2}{\gamma_j} \qquad (21)$$

RKHS theory is equivalent to the uniform convergence theory of Vapnik [39]. When the loss function $[y_i - f(\mathbf{x}_i)]^2$ is replaced by the hinge function $[1 - y_i f(\mathbf{x}_i)]_+$, the dual space solution gives SVM.

Solution of RKHS for the quadratic loss of is well-known:

$$\mathbf{f} = \mathcal{K}(\mathcal{K} + \beta I)^{-1}\mathbf{y}. \qquad (22)$$

At the large β limit, we get the solution

$$\mathbf{f} = \frac{1}{\beta}(\mathcal{K} - \frac{1}{\beta}\mathcal{K}^2 + \frac{1}{\beta^2}\mathcal{K}^3 - \cdots)\mathbf{y}. \qquad (23)$$

Now, setting the kernel as the Green's function: $\mathcal{K} = G$, using only the leading order and ignoring the proportional constant, this gives the learning algorithm Eq.(5) or Eq.(8) of §2. Standard kernel machines are used for supervised learning with $y_i = \pm 1$. For semi-supervised learning we set $y_i = 0$ for those unlabeled data.

6.1 Kernel Regularization using Laplacian

The above generic results is valid for any type of kernels. Here we focus on the kernel being the Green's function of Laplacian operator. Regularization approach is generally used to control smoothness of certain derivatives. The Laplacian operator, a scalar quantity constructed from second order derivatives is sometimes used as the regularization term [?, 42]. Our focus here is to show that different Laplacian regularization's at strong regularization limit can be captured by Green's functions. This leads to a simple and efficient algorithm.

We write the second term in RKHS explicitly,

$$\sum_j \frac{c_j^2}{\gamma_j} = \mathbf{f}^T\mathcal{K}^{-1}\mathbf{f} = \mathbf{f}^T(D-W)_+\mathbf{f} = \mathbf{f}^T(D-W)\mathbf{f}$$

Since $\mathbf{f}^T(D-W)\mathbf{f} = \frac{1}{2}\sum_{i,j=1}^n (f_i - f_j)^2 w_{ij}$, we can write

$$J[\mathbf{f}] = \sum_{i=1}^n [y_i - f(\mathbf{x}_i)]^2 + \frac{\beta}{2}\sum_{i,j=1}^n (f_i - f_j)^2 w_{ij}. \qquad (24)$$

The solution to this variational problem can be easily derived. We write $J(\mathbf{f}) = ||\mathbf{y} - \mathbf{f}||^2 + \beta\mathbf{f}^T\mathcal{K}^{-1}\mathbf{f}$, where $\mathcal{K}^{-1} = (D-W)_+$. Setting $\partial J/\partial\mathbf{f} = 2\mathbf{f} - 2\mathbf{y} + 2\beta\mathcal{K}^{-1}\mathbf{f} = 0$, we obtain $\mathbf{f} = (I + \beta\mathcal{K}^{-1})^{-1}\mathbf{y} = (I + \beta(D-W)_+)^{-1}\mathbf{y}$. Note that $(I + \beta\mathcal{K}^{-1})^{-1} = \mathcal{K}(\mathcal{K} + \beta I)^{-1}$, this recovers Eq.(22).

Zhou et al [42] propose the consistency framework that minimizes the functional

$$J[\mathbf{f}] = \mu\sum_{i=1}^n [y_i - f(\mathbf{x}_i)]^2 + \frac{1}{2}\sum_{i,j=1}^n (\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_i}})^2 w_{ij} \qquad (25)$$

and obtain

$$\mathbf{f} = \frac{\mu}{1+\mu}(I - \frac{1}{1+\mu}\widetilde{W})^{-1}\mathbf{y}. \qquad (26)$$

where $\widetilde{W}$ is defined in Eq.(18). They set $(1+\mu)^{-1} = 0.99$ (or $\mu = 0.01$) in practice.

n the strong regularization limit, $\mu \to 0$, Eq.26 becomes (ignoring a proportional constant) $\mathbf{f} = (I - \widetilde{W})^{-1}\mathbf{y}$. Discarding the zero mode (§4), $\mathbf{f} = (I - \widetilde{W})_+^{-1}\mathbf{y}$; this is a Green's function formulation of $(I - \widetilde{W})_+^{-1}$, which, strictly speaking however, is not a kernel[2].

7. UNSUPERVISED LEARNING

In this section, we show how the semi-supervised learning in §6 can be extend to unsupervised learning. This is based on the following three observations/perspectives.

(A) A key feature in the RKHS based formalism for the semi-supervised learning in §6 is that the Green's function learning is carried out in the parameter region where the regularization term is dominant i.e., the regression term is only a small perturbation to the regularization term which determines the unsupervised learning.

(B) In the semi-supervised learning using Green's function, on the labeled data points, the class labels are re-calculated using Eq.(6) or Eq.(8). It is important to note that the re-calculated labels may differ from the original ones. When

[2]Strictly speaking $(I - \widetilde{W})_+^{-1}$ is not a kernel, because not all functions in $\Re^n$ can be expanded by the eigenvectors of $(I - \widetilde{W})_+$ plus a constant, due to the fact that the excluded zero-mode of $(I - \widetilde{W})$ is $\mathbf{z}_1 = (d_1, \cdots, d_n)^T$ which is not a constant vector (see Eq.18).

this happens, we consider the newly computed labels as the correct label, because they are consistently computed in the same way as the labels on the unlabeled data points. Thus this mechanism allows us to correct mistakes on partially observed labels.
(C) Because of (A) and (B), we may consider the partially observed labels as temporary information, and the entire labels should be determined in a self-consistent manner. In this way, we can do the unsupervised learning by (1) temporarily assign label information to some or all data points and (2) iterate the label propagation a number of times until they are self-consistent. This is the rational for the unsupervised learning using Green's function, as explained in Eqs.(7,10).

7.1 Unsupervised Learning as a Maximization Problem

In this section, we capture the essence of the unsupervised learning by showing they are identical to the ratio cut and normalized cut spectral clustering. This shows the inherent consistency of the Green's function learning framework.

For simplicity, we write the influence propagates as

$$H^{(t+1)} = GH^{(t)}. \tag{27}$$

Proposition 3. In the unsupervised learning using Eq.(27), the converged solution is the optimal solution to the optimization problem $\max_H \mathrm{Tr}[H^T GH]$, with proper orthogonality condition.

Proof. It is a well-known in matrix computation theory that the solution for $\max \mathrm{Tr}[H^T GH]$ can be obtained by the subspace iteration algorithm[15] using Eq.(27) subject to the appropriate orthogonality condition. □

7.2 Equivalence to Spectral Clustering

A main results of this paper is to show that Green's function frame is equivalent to spectral clustering, the Ratio Cut[18] and the Normalized Cut[36]. It is known that Ratio Cut can be formulated as the optimization problem

$$H = \arg\max_{H^T H = I} \mathrm{Tr}[H^T G^{(1)} H]. \tag{28}$$

Similarly the Normalized Cut can be formulated as

$$H = \arg\max_{H^T DH = I} \mathrm{Tr}[H^T G^{(2)} H]. \tag{29}$$

Both optimization problems can be solved using the Green's function influence propagation approach with proper orthogonality. Therefore, our Green's function learning is equivalent to spectral clustering.

In recent years spectral clustering using the Laplacian of the graph emerges as solid approach for data clustering. Here we focus on the Ratio Cut [18] and the Normalized Cut [36] clustering objective functions. We are interested in the multi-way clustering objective functions,

$$J = \sum_{1 \le p < q \le K} \frac{s(C_p, C_q)}{\rho(C_p)} + \frac{s(C_p, C_q)}{\rho(C_q)} \tag{30}$$

where $\rho(C_k) = |C_k|$ for Ratio Cut, and $\rho(C_k) = \sum_{i \in C_k} d_i$ for Normalized Cut, and $s(C_p, C_q) = \sum_{i \in C_p} \sum_{j \in C_p} w_{ij}$. The multi-way clustering relaxation is studied in [17]. Using cluster indicators $H = (\mathbf{h}_1 \cdots \mathbf{h}_k)$, the Ratio-Cut becomes the minimization problem

$$\text{Ratio-Cut:} \quad \min_{H^T H = I} \mathrm{Tr}(H^T (D - W)H) \tag{31}$$

The Normalized Cut becomes the minimization problem

$$\text{Norm-Cut:} \quad \min_{H^T DH = I} \mathrm{Tr}(H^T (D - W)H) \tag{32}$$

We now show that the Ratio Cut optimization problem of Eq.(31) is equivalent to the optimization problem of Eq.(28) and the Normalized Cut optimization problem of Eq.(32) is equivalent to the optimization problem of Eq.(29).

Theorem 4. The iteration the algorithm of Eq.(27) with orthogonality $H^T H = I$ converges to an optimal solution of the multi-way Ratio Cut. The iteration the algorithm of Eq.(27) with orthogonality $H^T DH = I$ converges to an optimal solution of the multi-way Normalized Cut.

Proof. Clearly, the null space of $(D - W)$ does not contribute to $H^T (D - W)H$. This can be seen by doing eigen expansion of $L = D - W$ and the zero-eigenvalue modes drop out. Only the positive definite part of $(D - W)$ contribute. Thus we have

$$\begin{aligned} H &= \arg\min_{H^T H = I} \mathrm{Tr}[H^T (D - W)_+ H] \\ &= \arg\max_{H^T H = I} \mathrm{Tr}[H^T [(D - W)_+]^{-1} H]. \end{aligned}$$

This is $\mathrm{Tr}[H^T {G^{(1)}}^{-1} H]$. The orthogonality of the eigenvectors contained in $G^{(1)}$ is consistent with the $H^T H = I$ orthogonality. This proves the case for Ratio Cut. For Normalized Cut, the proof is identical. The D-orthogonality of the eigenvectors contained in $G^{(2)}$ is consistent with the $H^T DH = I$ orthogonality. This proves the case for normalized Cut. □

8. HARMONIC FUNCTION APPROACH

There exists a very large amount work on semi-supervised learning and we have mentioned several of them in §1. Among them, the closest to our Green's function approach are (1) the consistency framework of Zhou et al [42], which is discussed in §6 and (2) the harmonic function approach of Zhu et al. [44], which is discussed below. In §9, we do extensive experiments on seven datasets and compare our approach to these two methods.

"Harmonic function" refers to that fact the solution to $\nabla^2 f(\mathbf{r}) = 0$ has the property that $f(\mathbf{r}_i)$ equals to the average of $f(\mathbf{r})$ at $\mathbf{r}_i$'s neighbors. In their approach for semi-supervised learning, f_i is fixed to the label for labeled point i; Separating labeled and unlabeled nodes, the Laplace equation $Lf = 0$ becomes

$$\begin{pmatrix} D_{ll} + D_{lu} - W_{ll} & W_{lu} \\ W_{ul} & D_{ul} + D_{uu} - W_{uu} \end{pmatrix} \begin{pmatrix} f_l \\ f_u \end{pmatrix} = 0. \tag{33}$$

Focus on f_u part, we obtain

$$f_u^{\mathrm{HF}} = (D_{ul} + D_{uu} - W_{uu})^{-1} W_{ul} f_l. \tag{34}$$

Note the presence of the physical Laplacian

$$L_p = D_{ul} + D_{uu} - W_{uu},$$

as discussed in Corollary 2 (§4.1). Zhu et al. [44] obtain

$$f_u = (D_{uu} - W_{uu})^{-1} W_{ul} f_l,$$

which differs slightly from Eq.(34): D_{ul} is missing.

9. EXPERIMENTS

We apply Green's function (GF) approach to 7 datasets. We compare to prior methods which are closest to GF approach: (a) the consistent framework (CF) approach [42] and (b) the harmonic function (HF) approach [44].

Datasets. The seven datasets include Internet newsgroups and 6 datasets from UCI repository [11]. We use a 5-newsgroup subset of the standard 20-newsgroup dataset consisting of the following 5 categories: comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, and talk.politics.mideast. These 7 datasets are representative of real applications. Their data size, data dimension and number of classes are summarized in Table 1.

	#samples	# Dimensions	# Classes
newsgroup	497	1000	5
soybean	47	35	4
housing	506	13	3
protein	116	20	6
wine	178	13	3
balance	625	3	3
iris	150	4	3

Table 1: Dataset Descriptions

Semi-supervised Learning

We first study semi-supervised learning. We randomly selection 10% of data points and treat them as labeled data. The rest in the dataset are unlabeled data. We run the three methods. To get good statistics, we rerun these test 10 times so the labeled datasets are different from each run. Final results are the averages over these 10 runs. They are listed in Table 2. On one dataset, *housing*, all three methods are compatible. On other 6 datasets, our Green's function (GF) approach generally outperforms HF and CF methods, sometime very significantly.

	GF	HF	CF
newsgroup	91.2	80.6	32.6
soybean	89.6	52.3	36.2
housing	51.3	54.6	54.4
protein	46.6	27.4	27.6
wine	92.1	39.6	43.0
balance	53.2	46.6	47.5
iris	78.5	33.3	40.0

Table 2: Classification accuracy (in percentage) with 10% of data labeled. CF: Consistent Framework.

Semi-supervised Learning with Noisy Labels

For each dataset, 10% data points are randomly selected and are given correct labels. Another 5% data points are randomly selected and are given incorrect labels, to emulate the noises. All 3 learning methods are applied to the 7 datasets. Results for the average of 10 runs of random samples are listed in Table 2. In general, accuracies in Table 3 are slightly worse than those in Table 1, as expected. In the Line with "GF-MP", we present the results of doing multiple propagations. The results are generally improved from the single propagation.

	GF	GF-MP	HF	CF
newsgroup	90.8	87.8	27.2	30.9
soybean	76.1	85.7	54.8	36.2
housing	50.8	52.0	49.9	54.4
protein	44.5	48.3	23.8	27.6
wine	88.7	90.8	40.1	43.0
balance	52.3	53.2	64.4	47.2
iris	74.6	75.1	50.0	38.0

Table 3: Classification accuracy (in percentage) for 7 datasets with 10% correctly labeled and another 5% incorrectly labeled.

Effects of Dimension Reduction

In the learning formula Eq.(26) for CF, we apply the dimension reduction technique by expressing the function $(I - \frac{1}{1+\mu}\widetilde{W})^{-1}$ restricted in the first K principal eigenspace of $\widetilde{W}$ (see [10]). We applied this dimension reduction (DR) version of CF and the results are shown in Table 4. The first 2 columns are for the 10% labeled case as in Table 2. The second 2 columns are for the noise label case as in Table 3. Dimension reduction significantly and consistently improves the performance.

	CF	CF-DR	CF	CF-DR
% Labeled data	10%	10%	15%	15%
% Incorrectly Labeled	0%	0%	5%	5%
newsgroup	32.6	91.0	30.9	90.7
soybean	36.2	89.3	36.2	76.0
housing	54.4	51.5	54.4	51.0
protein	27.6	44.5	27.6	44.2
wine	43.0	91.1	43.0	89.5
balance	47.5	51.1	47.2	56.5
iris	40.0	80.0	38.0	77.8

Table 4: Results on Dimension Reduction

Unsupervised Learning

We study the unsupervised learning using Green's function approach. We use the results of K-means as initial starts, and run the influence propagation of Eq.(27). The results are given in Table 5. For 4 out of 7 datasets, the accuracy values are improved.

	Kmeans	Green Function
newsgroup	80.1	87.5
soybean	76.8	81.9
housing	60.5	52.6
protein	47.2	49.1
wine	92.7	91.5
balance	50.9	55.6
iris	78.8	73.2

Table 5: Results on unsupervised learning

10. ITEM-BASED RECOMMENDATION VIA LABEL PROPAGATION USING GREEN'S FUNCTION

Recommendation systems, as the *personalized* information navigation and filtering techniques used to identify a set of items that will be of interest to certain users, have been active and enjoying a growing amount of attentions with the explosive growth of world-wide-web and the emergence of e-commerce [1, 35]. They have been widely used to recommend products such as books, movies and musics to customers [27, 29] and to filter news stories [31, 2]. Various approaches for recommender systems have been developed by utilizing demographic, content, or historical information [5, 35, 9, 14]. Among these methods, *memory-based collaborative filtering* has been widely used in practice [41, 27, 20].

Memory-based methods for collaborative filtering aim at predicting user ratings for given items based on a collection of rating examples by averaging ratings between pairs of similar users or items. These methods can be further divided into user-based collaborative filtering [21, 20, 31] and item-based collaborative filtering [9, 40, 34]. To estimate an unknown rating of a given item by a test user, user-based methods first measure the similarities between the test user and other users, then the unknown rating is approximated by averaging the weighted known ratings of the given item by similar users. Similarly, item-based methods first measure the similarities between the test item and other items, the unknown rating is then calculated by averaging the known ratings of similar items by the test user.

Despite their success, however, there are still some major limitations with these recommendation methods such as data sparsity, recommendation reliability and scalability [33, 35]. In this section, we propose a novel item-based recommendation scheme via label propagation using Green's function. Many researchers have shown that item-based recommendation algorithms can recommend a set of items more quickly, with the recommendation results comparable to user-based methods [40]. We take a novel view by treating the item-based recommendation as the label information propagation. Given item similarity matrix W, the item recommendation can be viewed as label propagation from labeled data (i.e., items with ratings) to unlabeled data.

10.1 Notations and Framework

In a typical recommendation system, there is a set of users $\mathcal{U} = \{u_1, u_2, \cdots, u_M\}$ and a set of items $\mathcal{I} = \{i_1, i_2, \cdots, i_N\}$. And we can construct an $M \times N$ *user-item* matrix $\mathbf{R}$, with its (p, q)-th entry R_{pq} equal to the rating of the user u_p to the item i_q. If user u_p has not rated for item i_q, then $R_{pq} = 0$. Note that these ratings may either ordinal (as in *Movielens*[30]) or continuous (as in *Jester*[14]). We use $\mathbf{u}_p$ to denote the p-th row of $\mathbf{R}$, which is called the *user vector* of u_p, and $\mathbf{i}_q$ to denote the q-th column of $\mathbf{R}$, which is called the *item vector* of i_q.

Definition 1 (Item graph). *An item graph is an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where*
(1) $\mathcal{V} = \mathcal{I}$ is the node set ($\mathcal{I}$ is the item set, which means that each item is regarded as a node on the graph $\mathcal{G}$);
(2) $\mathcal{E}$ is the edge set. Associated with each edge $e_{pq} \in \mathcal{E}$ is a weight w_{pq} subject to $w_{pq} \geq 0, w_{pq} = w_{qp}$.

Typical similarity calculation methods include the *cosine similarity*, *conditional probability*, and *exponential cosine similarity* [9, 40]. In our study, we use the *cosine similarity* to build the item graph.

10.2 Item-based Recommendation

The recommendation on item graph can be viewed as a label propagation problem. Let $\mathbf{y}^T = (y_1, \cdots, y_n)$ be the rating for a user. Given an incomplete rating

$$\mathbf{y}_0^T = (5, ?, ?, 3, 1, ?, ?, ?, 8),$$

the question is to predict those missing values. Usually, the number of missing values are far greater than the number of known values. In the Green's function learning framework, we set

$$\mathbf{y}_0^T = (5, 0, 0, 3, 1, 0, 0, 0, 8),$$

and compute the complete rating as the linear influence propagation of Eq.(12)

$$\mathbf{y} = G\mathbf{y}_0, \tag{35}$$

where G is the Green's function built from the item graph.

Given a user-item matrix $\mathbf{R}$, with its (p, q)-th entry R_{pq} equal to the rating of the user u_p to the item i_q. Let R_0 contains the incomplete rating. Item-based recommendation is

$$R^T = GR_0^T, \tag{36}$$

One can see this is an extremely simple algorithm.

10.3 Experiments

In this section we experimentally evaluate the performance of our recommendation algorithm using Green's function and compare it with that of the traditional recommendation algorithms.

Dataset We use the Movielens [30] in our experiments. The *movielens* dataset is collected from a web-based research recommender system *MovieLens*, which is debut in Fall 1997. The dataset now contains the records of over 43000 users who have rated over 3500 different movies, with the ratings being integer values from 1 to 5. In our experiments, we use a subset of the 1 million movielens dataset, which contains 10,000 records including the ratings of 943 users to 1682 movies. We randomly divided this dataset into a training set (including 90,570 records) and a testing set (including 9,430 records), and the two sets have no intersections.

Evaluation Measures. We use the following three different measures in our experiments and we expect these measures would give us enough insights:

Mean Absolute Error (MAE): Given the target ratings R and the predicted ratings R', MAE is the average deviation of the prediction from the target, i.e.,

$$MAE(R') = \frac{1}{n}\sum_{i=1}^{n}|R'(i) - R(i)|$$

where n is the number of predicted ratings.

Mean Zero-one Error (MZOE): MZOE is defined by

$$MZOE(R') = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{R'(i)\neq R(i)},$$

i.e., it calculates the fraction of incorrect predictions.

Order Consistency (OC): Sometimes we do not need to compute very accurate ratings, but just want the predicted preferences (or, equivalently, the order of the unrated items) to be accurate. OC is used to measure how identical the predicted order is to the true order [40]. Assuming there are d items, $\mathbf{a}$ is the vector that these d items are sorted in an decreasing order according to their predicted ranking scores, $\mathbf{b}$ is the vector that these d items are sorted in an decreasing order according to their true ratings. For these d items, we have $C_d^2 = d!/(2!(d-2)!)$ ways to randomly select a pair of different items. $\mathcal{A}$ is the set of item pair whose order in $\mathbf{a}$ are the same as in $\mathbf{b}$, then order consistency (OC) is defined as:

$$OC = |\mathcal{A}|/C_d^2, \tag{37}$$

where $|\mathcal{A}|$ *represents the cardinality of* $\mathcal{A}$.

Result Analysis. We compare the performance of our Green's function approach with the traditional item-based methods and user-based methods using different similarity measures, namely the *cosine* (Cos), *conditional probability* (CP) and *exponential cosine* (ExCos) similarity measures. We also compare with Item Rating Smoothness Maximization (ISRM) recently developed in [40]. ISRM method explores the geometric information of the item data and computes the rating smoothness over the whole item graph via *combinatorial graph Laplacian*. It then predicts the ratings of a user to his unrated items by minimizing the rating smoothness.

Table 6 shows the performance comparisons of various methods. We observe that our Green's function method has the lowest MAE and MOE errors among all the recommendation methods. The comparison also shows that item-based recommendation methods perform better than the user-based methods. Figure 1 presents the order consistency (OC) values of Green's function method, item-based methods and the ISRM method and it shows that our Green's function method also have the best OC values. The experimental results illustrate the effectiveness of our Green's function approach.

Methods \ Measures	MAE	MOE
Green's Function	**0.6907**	**1.0702**
Item-Cos	0.6980	1.0769
Item-ExCos	0.6978	1.116
Item-CP	0.7054	1.1381
User-Cos	0.7649	1.3302
User-ExCos	0.7631	1.322
User-CP	0.7270	1.2346

Table 6: Performance Comparisons of Various Recommendation Methods.

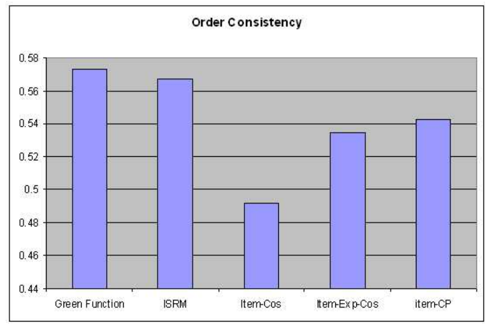

Figure 1: Order Consistency Performance Comparisons.

11. SUMMARY

In this paper, we propose to use Green's function as a mechanism of label information propagation. Theoretically, (1) we show that the zero-mode of the combinatorial Laplace matrix is originated from the von Neumann boundary condition, and thus its zero-mode must be a constant vector, which therefore should be discarded. (2) We derive the Green's function learning framework from the kernel regularization using Reproducing Kernel Hilbert Space theory at strong regularization limit. (3) We clarify that in semi-supervised learning, setting data points with known labels as boundary point is equivalent to using Dirichlet boundary condition and the results of the Laplacian operator approach will involve the physical Laplacian, rather than the combinatorial Laplacian. Overall, our results clarify the exact mechanisms of the often *vague* concept of label propagation.

We also show that the Green's function approach is closely related to the well-established distance metric on a graph, i.e., the effective resistor distance (via an analogy to a network of electric resistors) and the average commute time via random walks. These give more concrete understanding of Green's function approach and will help to derive more efficient approximations and effective variants.

We also performed extensive experiments on 7 datasets and the experimental results indicate the Green's function approach outperform other approaches. Finally, we propose a novel item-based recommender system using Green's function.

Acknowledgments

We thank Marco Saerens and Hongyuan Zha for pointing out the relationship to random walk and resistor distance. C. Ding and H.D. Simon are supported by the US Dept of Energy, Office of Science, the LBNL LDRD funding, under Contract No. DE-AC02-05CH11231. T. Li is partially supported by a IBM Faculty Research Award, NSF CAREER Award IIS-0546280 and NIH/NIGMS S06 GM008205.

12. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] D. Billsus and M. J. Pazzani. A personal news agent that talks, learns and explains. In *AGENTS '99*, pages 268–275, 1999.

[3] A. Blum, J. Lafferty, M. Rwebangira, and R. Reddy. Semi-supervised using randomized mincuts. *Proc. ICML 2004. pp.19-26.*, 2004.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *Proc. Comp. Learning Theo. (CLT1998) pp.92-100*, 1998.

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[6] A.K. Chandra, P. Raghavan, W. L. Ruzzo, R. Smolensky, and P. Tiwari. Electrical resistance of a graph captures its commute and cover times. *Proc. ACM Symposium on Theory of Computing*, 1989.

[7] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[8] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. *NIPS 2002*, 2002.

[9] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.

[10] C. Ding, X. He, H. Zha, and H. Simon. Unsupervised learning: self-aggregation in scaled principal component space. *PKDD'02*, pages 112–124, 2002.

[11] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[12] P. G. Doyle and L. Snell. *Random Walks and Electric Networks*. Mathematical Assn of America, 1984.

[13] F. Gobel and A. A. Jagers. Random walks on graphs. *Stochastic processes and their applications*, 2:311–336, 1974.

[14] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

[15] G. Golub and C. Van Loan. *Matrix Computations, 3rd edition*. Johns Hopkins, Baltimore, 1996.

[16] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. *NIPS 2004*, 2004.

[17] M. Gu, H. Zha, C. Ding, X. He, and H. Simon. Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering. *Penn State Univ Tech Report CSE-01-007*, 2001.

[18] L. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE. Trans. on Computed Aided Design*, 11:1074–1085, 1992.

[19] T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning*. Springer Verlag, 2001.

[20] J.L. Herlocker, J.A. Konstan, A.l. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99*, pages 230–237, 1999.

[21] R. Jin, J. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *SIGIR '04*, pages 337–344, 2004.

[22] T. Joachims. Transductive learning via spectral graph partitioning. *Proc. ICML 2003.*, 2003.

[23] D. J. Klein and M. Randic. Resistance distance. *J. Math. Chemistry*, 12:81–95, 1993.

[24] R. Kondor and J.-P. Vert. Diffusion kernels. *Kernel Methods in Computational Biology, ed. B. Scholkopf, K. Tsuda and J.-P. Vert, MIT Press, p.209-230*, 2002.

[25] R.I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. *Proc. Int'l Conf. Machine Learning*, 2002.

[26] N.D. Lawrence and M.I. Jordan. Semi-supervised learning via gaussian process. *Neural Info. Processing Systems (NIPS 2004)*, 2004.

[27] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[28] K.V. Mardia, J.T. Kent, and J.B. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[29] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl. Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of ACM 2003 Conference on Intelligent User Interfaces (IUI'03)*, 2003.

[30] GroupLens Research. http://movielens.umn.edu/, 2006.

[31] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, 1994.

[32] M. Saerens, F. Fouss, L. Yen, and P. Dupont. Principal components analysis of a graph, and its relationships to spectral clustering. *ECML*, 2004.

[33] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender systems–a case study. In *ACM WebKDD Workshop*, 2000.

[34] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01*, pages 285–295, 2001.

[35] J. Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, 1999.

[36] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[37] A.J. Smola and R.I. Kondor. Kernels and regularization on graphs. *Conference on Learning Theory and 7th Kernel Workshop*, pages 144–158, 2003.

[38] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. *NIPS*, 2001.

[39] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[40] F. Wang, S. Ma, L. Yang, and T. Li. Recommendation on item graphs. In *ICDM'06*, pages 1119–1123, 2006.

[41] J. Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06*, pages 501–508, 2006.

[42] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *Proc. Neural Info. Processing Systems*, 2003.

[43] X. Zhu. Semi-supervised learning literature survey. *University of Wisconsin CS TR-1530*, 2006.

[44] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *Proc. Int'l Conf. Machine Learning*, 2003.