A Regularization Framework for Learning from Graph Data

Dengyong Zhou DENGYONG.ZHOU@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics
Spemannstr. 38, 72076 Tuebingen, Germany

Bernhard Schölkopf BERNHARD.SCHOELKOPF@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics
Spemannstr. 38, 72076 Tuebingen, Germany

Abstract

The data in many real-world problems can be thought of as a graph, such as the web, co-author networks, and biological networks. We propose a general regularization framework on graphs, which is applicable to the classification, ranking, and link prediction problems. We also show that the method can be explained as lazy random walks. We evaluate the method on a number of experiments.

1. Introduction

In many real-world problems, the data can be represented as a graph. Each vertex of the graph corresponds to a datum, and the edges encode the pairwise relationships or similarities among the data. A typical example of graph data is the web. The vertices are just the web pages, and the edges denote the hyperlinks. In market basket analysis, the items also form a graph by connecting any two items which have appeared in the same shopping basket. More examples include co-author relationships, terrorists networks, biological networks and so on.

One problem addressed in this paper is classification. Specifically speaking, some of the vertices are labeled, and the task is to classify the remaining unlabeled vertices. A toy classification problem is shown in Figure 1. Two vertices of the graph are labeled as positive and negative respectively. A real-world example is to classify the web pages into different categories given some manually classified instances. Obviously, a good classifier should effectively exploit the relations among the data.

Appearing in *Workshop on Statistical Relational Learning at International Conference on Machine Learning*, Banff, Canada, 2004. Copyright 2004 by the first author.

The other problem investigated here is ranking. This problem generally can be understood as finding the vertices of most interest. For instance, given a terrorist network, detect the people who are the core of the criminal community. Another ranking problem is to find the vertices most relevant to some given vertices (Figure 3), which are often called quires. We call this problem relative ranking as distinct from the absolute ranking without quires. The example of relative ranking in terrorist networks is to discover the criminals who have strong connections to some given criminals.

A problem closely related to ranking is link prediction (Figure 3). For example, given a co-author network, predict which two scientists are most likely to collaborate in the future. Link prediction is essentially the relative ranking problem. In fact, we can compute the pairwise relevance between any two unconnected vertices, and then pick up the pair of vertices which are most relevant.

Recently there has been considerable research on these problems. Here we try to introduce a general regularization framework on graphs as a new approach to these problems. Our idea is very simple. First develop discrete calculus on graphs, and then naturally shift classical regularization from the continuous case to graph data.

2. Regularization Framework

A graph $\Gamma = (V, E)$ consists of a set V of vertices and a set of pairs of vertices $E \subseteq V \times V$ called edges. A graph is undirected if for each edge $(u, v) \in E$ we also have $(v, u) \in E$. Edge e is incident on vertex v if e contains v. Suppose that Γ is connected, i.e., there is a path from every vertex to every other vertex. Suppose further that Γ has no self-loops or multiple edges. A graph is weighted if it is associated with a function

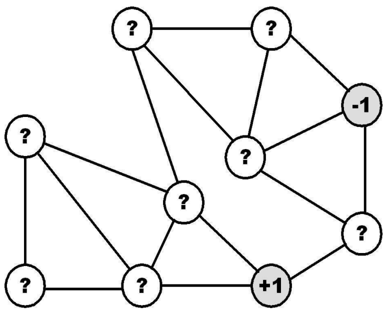

Figure 1. Classification problem on a graph. Two vertices are labeled as $+1$ and -1 respectively. The goal is to classify the remaining unlabeled vertices.

$w : V \times V \to \mathbb{R}$ satisfying

$$w(u,v) > 0, \text{ if } (u,v) \in E, \tag{2.1}$$

and

$$w(u,v) = w(v,u). \tag{2.2}$$

The degree function $d : V \to \mathbb{R}$ is defined to be

$$d(v) = \sum_{u \sim v} w(u,v), \tag{2.3}$$

where $u \sim v$ denotes all vertices u connected to v by the edges (u,v).

Let $L^2(V)$ denote the Hilbert space of real-valued function $f : V \to \mathbb{R}$ endowed with the usual inner product

$$\langle f, g \rangle = \sum_v f(v) g(v), \tag{2.4}$$

i.e., the functions are thought of as column vectors.

Let e denote the edge between vertices u and v. The *edge derivative* of function f along e at the vertex u is defined to be

$$\left.\frac{\partial f}{\partial e}\right|_u = \sqrt{\frac{w(u,v)}{d(u)}} f(u) - \sqrt{\frac{w(u,v)}{d(v)}} f(v). \tag{2.5}$$

See Appendix A for the reason of adopting such a definition. Clearly

$$\left.\frac{\partial f}{\partial e}\right|_u = -\left.\frac{\partial f}{\partial e}\right|_v. \tag{2.6}$$

The *local variation* of f at each vertex v is then defined to be

$$\|\nabla_v f\| = \sqrt{\sum_{e \vdash v} \left(\left.\frac{\partial f}{\partial e}\right|_v\right)^2}, \tag{2.7}$$

where $e \vdash v$ denotes the set of the edges incident with vertex v. The *smoothness* of f is then naturally measured by the sum of the local variations at each vertex:

$$\mathcal{S}(f) = \frac{1}{2} \sum_v \|\nabla_v f\|^2. \tag{2.8}$$

The learning problems on graphs generally can be thought of seeking for a function f, which is smooth and simultaneously close to another given function y. This view can be formalized as the following optimization problem:

$$\arg \min_{f \in L^2(V)} \left\{ \mathcal{S}(f) + \frac{\mu}{2} \|f - y\|^2 \right\}. \tag{2.9}$$

The first term in the bracket measures the smoothness of the function f, and the second term measures its *closeness* to the given function y. The trade-off between these two terms is captured by a nonnegative parameter μ. Later we will show how this general framework fits into the learning problems introduced in Section 1.

Before solving this optimization problem, we introduce the Laplace operator $\Delta : L^2(V) \to L^2(V)$ defined by

$$(\Delta f)(v) = \frac{1}{2} \sum_{e \vdash v} \frac{1}{\sqrt{d}} \left(\frac{\partial}{\partial e} \sqrt{d} \frac{\partial f}{\partial e}\right)\bigg|_v. \tag{2.10}$$

We can show that Δ is linear and symmetrical. In fact,

$$\begin{aligned}
(\Delta f)(v) &= \frac{1}{2\sqrt{d(v)}} \sum_{e \vdash v} \frac{\partial}{\partial e} \left(\sqrt{d} \frac{\partial f}{\partial e}\right)\bigg|_v \\
&= \frac{1}{2\sqrt{d(v)}} \sum_{e \vdash v} \left[\sqrt{\frac{w(u,v)}{d(u)}} \left(\sqrt{d} \frac{\partial f}{\partial e}\right)\bigg|_u \right. \\
&\quad \left. - \sqrt{\frac{w(u,v)}{d(v)}} \left(\sqrt{d} \frac{\partial f}{\partial e}\right)\bigg|_v \right] \\
&= \frac{1}{2\sqrt{d(v)}} \sum_{e \vdash v} \sqrt{w(u,v)} \left(\left.\frac{\partial f}{\partial e}\right|_u - \left.\frac{\partial f}{\partial e}\right|_v \right) \\
&= \frac{1}{\sqrt{d(v)}} \sum_{e \vdash v} \sqrt{w(u,v)} \left.\frac{\partial f}{\partial e}\right|_v \\
&= \frac{1}{\sqrt{d(v)}} \sum_{u \sim v} \left(\frac{w(u,v)}{\sqrt{d(v)}} f(v) - \frac{w(u,v)}{\sqrt{d(u)}} f(u) \right) \\
&= f(v) - \sum_{u \sim v} \frac{w(u,v)}{\sqrt{d(u) d(v)}} f(u).
\end{aligned}$$

The last equality is usually used as the definition of the graph Laplacian in many literatures, such as [2].

It is not hard to show

$$f^T \Delta f = \mathcal{S}(f). \tag{2.11}$$

Note that this also shows that Δ is positive semi-definite.

Theorem 1. *The solution of the optimization problem (2.9) satisfies* $\Delta f + \mu(f - y) = 0$.

Proof. By Equality (2.11), we have

$$(\Delta f)(v) = \left.\frac{\partial \mathcal{S}(f)}{\partial f}\right|_v .$$

Differentiating the cost function in the bracket of (2.9) with respect to f completes the proof. $\square$

Let us introduce another operator $S : L^2(V) \to L^2(V)$ by

$$(Sf)(v) = \sum_{u\sim v} \frac{w(u,v)}{\sqrt{d(u)d(v)}} f(u). \tag{2.12}$$

Then Δ can be rewritten into

$$\Delta = I - S, \tag{2.13}$$

where I is the identity operator.

Corollary 2. *If* $y \neq 0$ *and* $\mu > 0$, *the solution of the optimization problem (2.9) is*

$$f = (1-\alpha)(I - \alpha S)^{-1} y, \tag{2.14}$$

where $\alpha = 1/(1+\mu)$.

Proof. By Theorem 1 and Equality (2.13), we have

$$(I - S)f + \mu(f - y) = 0,$$

which can be transformed into

$$f - \frac{1}{1+\mu} Sf - \frac{\mu}{1+\mu} f = 0.$$

Note the definition of α. Then

$$(I - \alpha S)f = (1-\alpha)y,$$

It is not hard to see that the eigenvalues of S are at most equal to 1. Hence $I - \alpha S$ is invertible, and we have

$$f = (1-\alpha)(I - \alpha S)^{-1} y.$$

$\square$

(2.14) is in fact the algorithm proposed by [7], where it was applied to semi-supervised learning problems with vectorial data.

For large-scale datasets, we can consider using the following iteration to compute f:

$$f(v) \leftarrow \alpha(Sf)(v) + (1-\alpha)y, \ \forall v \tag{2.15}$$

with the initial value $f(v) = y(v)$. We can intuitively understand the iteration as the process of information diffusion on graphs. In each round, every vertex updates its value by linearly combining its neighbor's current values with the initial value of itself. The positive parameter $\alpha \in (0,1)$ specifies the relative amount. It is not hard to show that the iteration converges to (2.14)[7]. The iteration can be expected to converge quickly when the graph is sparse (for instance, the web graph).

Corollary 3. *If* $y = 0$ *or* $\mu = 0$, *the non-zero solution of the optimization problem (2.9) is the principle eigenfunction of* S.

Proof. In the case of $y = 0$, by Theorem 1, we have $\Delta f + \mu f = 0$. Substituting (2.13) into the equality, then we obtain $Sf = (1+\mu)f$, which means that f is one of the eigenfunctions of S. Note that the eigenvalues of S are at most equal to 1. This requires that $1 + \mu \leq 1$. Since $\mu \geq 0$, we have $\mu = 0$. Hence f is the eigenfunction corresponding to the largest eigenvalue 1, i.e. the principle eigenfunction. $\square$

3. Lazy Random Walks

The regularization framework can be interpreted as *lazy random walks*. Let D denote the diagonal matrix with the (u,u)-entry equal to $d(u)$, and let W denote the matrix with the (u,v)-entry equal to $w(u,v)$ if $(u,v) \in E$ and 0 otherwise. A lazy random walk on graph Γ is decided by the transition probability matrix $P = (1-\alpha)I + \alpha D^{-1}W$, where α is a parameter in $(0,1)$. This means, with the probability α, following one link incident with the vertex of the current position and is chosen with the probability proportional to the weight of the link, and with the probability $1-\alpha$, just staying at the current position.

Assume there are n vertices. There exists a unique stationary distribution $\pi = [\pi_1, \ldots, \pi_n]$ for the lazy random walk, i.e. a unique probability distribution satisfying the balance equation

$$\pi = \pi P. \tag{3.1}$$

Let $\mathbf{1}$ denote the $1 \times n$ vector with all entries equal to 1.

$$\begin{aligned}
\mathbf{1}DP &= \mathbf{1}D[(1-\alpha)I + \alpha D^{-1}W] \\
&= (1-\alpha)\mathbf{1}D + \alpha \mathbf{1}DD^{-1}W \\
&= (1-\alpha)\mathbf{1}D + \alpha\mathbf{1}W \\
&= (1-\alpha)\mathbf{1}D + \alpha\mathbf{1}D \\
&= \mathbf{1}D.
\end{aligned}$$

Let vol Γ denote the volume of the graph, which is defined to be the sum of vertex degrees. Then the stationary distribution can be written as

$$\pi = \mathbf{1}D/\mathrm{vol}\,\Gamma. \tag{3.2}$$

Note that π does not depend on α. Hence π is also the stationary distribution of the random walk with the transition probability matrix $M = D^{-1}W$. In addition, the matrix $S = D^{-1/2}WD^{-1/2}$ can be rewritten in terms of stationary distribution:

$$S(u,v) = \pi_u^{1/2} M(u,v) \pi_v^{-1/2}, \tag{3.3}$$

which also shows that S is similar to M.

Let X_t denote the position of the random walk at time t. Write $T(u,v) = \min\{t \geq 0 | X_t = v, X_0 = u, u \neq v\}$ for the *first hitting time* to v with the initial position u, and write $T(v,v) = \min\{t > 0 | X_t = v, X_0 = v\}$, which is called the *first return time* to v [1]. Let $H(u,v)$ denote the expected number of steps required for a random walk to reach v with an initial position u, i.e. $H(u,v)$ is the expectation of $T(u,v)$. $H(u,v)$ is called the *hitting time* [1]. Let $C(u,v)$ denote the expected number of steps for a random walk starting at u to reach v and then return, i.e. $C(u,v) = H(u,v) + H(v,u)$. $C(u,v)$ is called the *commute time* between u and v. Clearly, $C(u,v)$ is symmetrical, but $H(u,v)$ may be not. Let G denote the inverse of the matrix $D - \alpha W$. For distinct vertices u and v, the commute time satisfies [3]:

$$C(u,v) \propto G(u,u) + G(v,v) - G(u,v) - G(v,u), \tag{3.4}$$

and [1]

$$C(u,u) = 1/\pi(u). \tag{3.5}$$

The relation between G and C is similar to the inner product and the norm in Euclidean space. In other words, we can think of G as a Gram matrix which specifies a kind of inner product on the dataset. The commute time is the corresponding metric derived from this inner product [3].

Note that $H(u,v)$ is quite small whenever v is a node with a large stationary probability $\pi(v)$. Thus we can consider normalizing $H(u,v)$ by

$$\bar{H}(u,v) = \sqrt{\pi(u)\pi(v)}H(u,v). \tag{3.6}$$

Accordingly, the normalized commute time is

$$\bar{C}(u,v) = \bar{H}(u,v) + \bar{H}(v,u). \tag{3.7}$$

Let $\bar{G}$ denote the inverse of the matrix $I - \alpha S$. Then the normalized commute time satisfies

$$\bar{C}(u,v) \propto \bar{G}(u,u) + \bar{G}(v,v) - \bar{G}(u,v) - \bar{G}(v,u). \tag{3.8}$$

Noting Equality (3.5), we have

$$\bar{G}(u,v) = \frac{G(u,v)}{\sqrt{C(u,u)C(v,v)}}, \tag{3.9}$$

which is similar to the normalized Euclidean product or cosine.

4. Bayesian Interpretation

There is a simple Bayesian interpretation for the regularization framework inspired by [6]. Let $p(f)$ denote the prior probability of f, and let $p(y|f)$ denote the conditional probability of y given f. Then the MAP estimation is given by

$$\arg\max_{f \in L^2(V)} \Big\{ \log p(y|f) + \log p(f) \Big\}. \tag{4.1}$$

A general model for the prior distribution $p(f)$ is given by

$$p(f) = \frac{1}{Z_r} \exp\left[- \frac{\mathcal{S}(f)}{\mu} \right], \tag{4.2}$$

where Z_r is a normalization constant. Further, the conditional probability is given by

$$p(y|f) = \frac{1}{Z_c} \exp\left(- \frac{\|f - y\|}{2\sigma^2} \right), \tag{4.3}$$

where Z_c is another normalizing constant. Thus the MAP estimator (4.1) yields

$$\arg\min_{f \in L^2(V)} \Big\{ \mathcal{S}(f) + \frac{\mu}{2}\|f - y\|^2 \Big\}. \tag{4.4}$$

5. Applications

In this section, we apply the general framework to the learning problems introduced in Section 1, including classification, relative ranking, and link prediction. In all experiments, the regularization parameter α is simply fixed at 0.90. In our future research, we will address how to choose a suitable α.

5.1. Classification

Assume partial vertices are labeled as positive or negative. We want to predict the labels of the other vertices. Define $y(v) = 1$ or -1 if u is labeled as positive or negative and 0 otherwise. Then compute $f = (I - \alpha S)^{-1}y$, and classify the unlabeled vertices v as $y(v) = \mathrm{sgn}(f(v))$.

We applied this classification method to the toy problem shown in Figure 1 and obtained the result shown

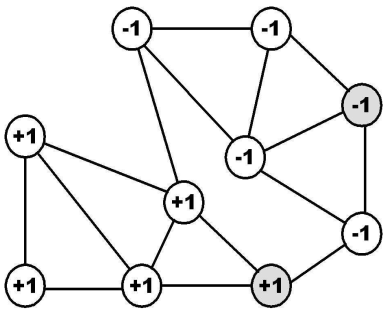

Figure 2. Classification on a toy graph. The two shaded circles are the initially labeled vertices. Note that the nodes can not be classified correctly if the classification only naively depends on the shortest paths to the labeled nodes.

in Figure 2. Apparently, the classification is consistent with our intuition. If we simply classify the unlabeled vertices by comparing the minimal distances from them to the labeled vertices, i.e., the length of the shortest paths, then some vertices can not be correctly classified.

5.2. Relative Ranking

Given a vertex (query) of interest in a graph, rank the remaining vertices with respect to their relevances to the given vertex. This can be viewed as an extreme case of classification problem, in which only positive examples are available. In other words, in some sense, relative ranking can be regarded as one-class classification problem [5]. Hence, similarly, define y with $y(v) = 1$ if v is the query and 0 otherwise and compute $f = (I - \alpha S)^{-1} y$. Then rank the vertices v according with $f(v)$ (largest ranked first) [8].

We address a toy ranking problem shown in the top panel of Figure 3. This toy network was first suggested by [4] for the problem measuring betweenness centrality in social networks. Assume that A is the query. The task is to rank the remaining vertices with respect to A. According with our intuition, D should be most similar to A because they are in the same *group*. Next should be B. Let us imagine that the vertices respectively in the left and right groups communicate with each other by passing their messages through the links. Then both A and B will be quite busy exchanging messages from the two opposite groups. Next C, which can be regarded as a redundant vertex in the context of communication. Hence the idea ranking list should be $A \to D \to B \to C \to E$. If we simply rank the vertices according with shortest paths, then the vertices D, B and C are similar to A at the same level. The ranking cores given by our method is as follows:

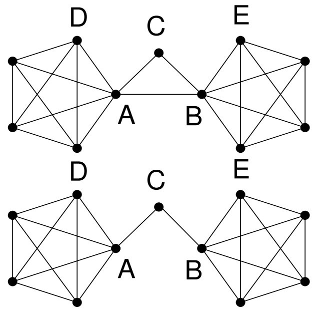

Figure 3. Ranking on a toy network. Top panel: ranking the vertices according to their relevances to the vertex A. Bottom panel: link prediction problem, in which the edge between the vertices A and B is removed.

	B	C	D	E
A	0.99	0.87	1.33	0.56

This ranking list is consistent with our intuitive analysis.

5.3. Link Prediction

This problem is essentially as same as ranking: choosing each vertex as the query and ranking the other vertices with respect to it, then adding a link between two most relevant vertices. This is equivalent to compute the matrix $(I - \alpha S)^{-1}$ and choose the entry which corresponds to the maximal value.

We investigate the toy network shown in the bottom panel of Figure 3, in which the link between A and B is removed with respect to the network shown in the top panel. The goal is to predict this removed link.

The relevance scores among unconnected vertices are the following:

(A, B)	(A, E)	(D, B)	(D, E)	(C, E)
0.48	0.29	0.29	0.18	0.52

This means the most possible link in the future is between C and E (and symmetrically the link between C and D as well). Unfortunately, this is not consistent with our setting. However, if we choose $\alpha = 0.95$, then

(A, B)	(A, E)	(D, B)	(D, E)	(C, E)
1.27	0.93	0.93	0.69	1.16

Now the next link predicted by the scores is between A and B. Different from the previous experiments, the result is sensitive to the choice of α. This shows how to choose a suitable α is a very important problem.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments.

A. Laplacian in Euclidean Space

Let f denote a differentiable function defined on $\mathbb{R}^m$. Then the gradient of function f at point $x = [x_1, \ldots x_m]^T$ is the vector

$$\nabla f(x) = \left[\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_m}\right]^T,$$

and

$$\|\nabla f(x)\| = \sqrt{\sum_{i=1}^{m} \left(\frac{\partial f}{\partial x_i}\right)^2}.$$

The smoothness of f is generally measured by

$$\mathcal{S}(f) = \frac{1}{2} \int \|\nabla f\|^2 dx,$$

which is usually called the *Dirichlet form* or *energy function.*

The Laplacian in the continuous case is a second-order differential operator defined by

$$\Delta f = \sum_{i=1}^{m} \frac{\partial^2 f}{\partial^2 x_i}.$$

The connection between the Laplacian and the gradient is expressed by

$$\int f(\Delta f) dx = \int \|\nabla f\|^2 dx,$$

which is exactly parallel to (2.11). In this sense, it is reasonable to think of (2.5) as the derivative on graphs and further use (2.8) to measure the smoothness of functions.

References

[1] D. Aldous and J. Fill. *Reversible Markov Chains and Random Walks on Graphs.* In Preparation, http://stat-www.berkeley.edu/users/aldous/RWG/book.html.

[2] F. Chung. *Spectral Graph Theory.* Number 92 in Regional Conference Series in Mathematics. American Mathematical Society, 1997.

[3] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[4] M. Newman. A measure of betweenness centrality based on random walks. Preprint cond-mat/0309045, 2003.

[5] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[6] G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* SIAM, 1990.

[7] 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*, Cambridge, MA, 2004. MIT press.

[8] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT press.