Regularization on Discrete Spaces

Dengyong Zhou and Bernhard Schölkopf

Max Planck Institute for Biological Cybernetics
Spemannstr. 38, 72076 Tuebingen, Germany
{dengyong.zhou, bernhard.schoelkopf}@tuebingen.mpg.de

Abstract. We consider the classification problem on a finite set of objects. Some of them are labeled, and the task is to predict the labels of the remaining unlabeled ones. Such an estimation problem is generally referred to as transductive inference. It is well-known that many meaningful inductive or supervised methods can be derived from a regularization framework, which minimizes a loss function plus a regularization term. In the same spirit, we propose a general discrete regularization framework defined on finite object sets, which can be thought of as discrete analogue of classical regularization theory. A family of transductive inference schemes is then systemically derived from the framework, including our earlier algorithm for transductive inference, with which we obtained encouraging results on many practical classification problems. The discrete regularization framework is built on discrete analysis and geometry on graphs developed by ourselves, in which a number of discrete differential operators of various orders are constructed, which can be thought of as discrete analogues of their counterparts in the continuous case.

1 Introduction

Many real-world machine learning problems can be described as follows: given a set of objects $X = \{x_1, x_2, \ldots, x_l, x_{l+1}, \ldots, x_n\}$ from a domain of $\mathcal{X}$ (e.g., $\mathbb{R}^d$) of which the first l objects are labeled as $y_1, \ldots, y_l \in \mathcal{Y} = \{1, -1\}$, the goal is to predict the labels of remaining unlabeled objects indexed from $l + 1$ to n. If the objects to classify are totally unrelated to each other, we cannot make any prediction statistically better than random guessing. Hence we generally assume that there are pairwise relationships among data. A dataset endowed with pairwise relationships can be naturally thought of as a graph. In particular, if the pairwise relationships are symmetric, then the graph is undirected. Thus we consider learning on graphs.

Any supervised learning algorithm can be applied to this problem, by training a classifier $f : \mathcal{X} \to \mathcal{Y}$ on the set of pairs $\{(x_1, y_1), \ldots, (x_l, y_l)\}$, and then using the trained classifier f to predict the labels of the unlabeled objects. Following this approach, one will have estimated a classification function defined on the whole domain $\mathcal{X}$ before predicting the labels of the unlabeled objects. According to [8], estimating a classification function defined on the whole domain $\mathcal{X}$ is more complex than the original problem which only requires predicting the labels of the given unlabeled objects, and it is simpler to directly predict the labels of the given unlabeled objects. Therefore we consider estimating a discrete classification function which is defined on the given objects X only. Such an estimation problem is called *transductive inference* [8].

Many meaningful inductive methods can be derived from a regularization framework, which minimizes an empirical loss plus a regularization term. Inspired by this work, we develop a general discrete regularization framework defined on graphs, and then derive a family of transductive algorithms from the discrete regularization framework. This framework can be considered as discrete analogues of variational problems [2, 3, 5] and classical regularization [7, 9]. The transductive inference algorithm which we proposed earlier [11] can be naturally derived from this framework, as can various new methods. Furthermore, to a certain extend, much existing work can be thought of in the framework of discrete regularization on graphs. The discrete regularization framework is built on discrete analysis and differential geometry on graphs developed by ourselves, in which a number of discrete differential operators of various orders are constructed. We follow the notation used in classical differential topology and geometry, which can be found in any standard textbook, e.g., see [6].

2 Discrete Analysis and Differential Geometry

2.1 Preliminaries

A graph $G = (V, E)$ consists of a finite set V, together with a subset $E \subseteq V \times V$. The elements of V are the *vertices* of the graph, and the elements of E are the *edges* of the graph. We say that an edge e is *incident* on vertex v if e starts from v. A *self-loop* is an edge which starts and ends at the same vertex. A *path* is a sequence of vertices $(v_1, v_2, \dots, v_m)$ such that $[v_{i-1}, v_i]$ is an edge for all $1 < i \leq m$. A graph is *connected* when there is a path between any two vertices. A graph is *undirected* when the set of edges is *symmetric*, i.e., for each edge $[u, v] \in E$ we also have $[v, u] \in E$. In the following, the graphs are always assumed to be connected, undirected, and have no self-loops or multiple edges.

A graph is *weighted* when it is associated with a function $w : E \to \mathbb{R}_+$ which is symmetric, i.e. $w([u, v]) = w([v, u])$, for all $[u, v] \in E$. The *degree* function $d : V \to \mathbb{R}_+$ is defined to be $d(v) := \sum_{u \sim v} w([u, v])$,where $u \sim v$ denote the set of the vertices *adjacent with* v, i.e. $[u, v] \in E$. Let $\mathcal{H}(V)$ denote the Hilbert space of real-valued functions endowed with the usual inner product $\langle f, g \rangle_{\mathcal{H}(V)} := \sum_{v \in V} f(v)g(v)$, for all $f, g \in \mathcal{H}(V)$. Similarly define $\mathcal{H}(E)$. Note that function $h \in \mathcal{H}(E)$ have not to be symmetric. In other words, we do not require $h([u, v]) = h([v, u])$.

2.2 Gradient and Divergence Operators

In this section, we define the discrete gradient and divergence operators, which can be thought of as discrete analogues of their counterparts in the continuous case.

Definition 1. *The graph gradient is an operator* $\nabla : \mathcal{H}(V) \to \mathcal{H}(E)$ *defined by*

$$(\nabla\varphi)([u,v]) := \sqrt{\frac{w([u,v])}{g(v)}}\varphi(v) - \sqrt{\frac{w([u,v])}{g(u)}}\varphi(u), \text{ for all } [u,v] \in E. \tag{1}$$

The gradient measures the variation of a function on each edge. Clearly,

$$(\nabla\varphi)([u,v]) = -(\nabla\varphi)([v,u]), \tag{2}$$

i.e., $\nabla\varphi$ is skew-symmetric.

We may also define the graph gradient at each vertex. Given a function $\varphi \in \mathcal{H}(V)$ and a vertex v, the gradient of φ at v is defined by $\nabla\varphi(v) := \{(\nabla\varphi)([v,u]) | [v,u] \in E\}$. We also often denote $\nabla\varphi(v)$ by $\nabla_v\varphi$. Then the norm of the graph gradient $\nabla\varphi$ at vertex v is defined by

$$\|\nabla_v\varphi\| := \left(\sum_{u\sim v}(\nabla\varphi)^2([u,v])\right)^{\frac{1}{2}},$$

and the *p-Dirichlet form* of the function φ by

$$\mathcal{S}_p(\varphi) := \frac{1}{2}\sum_{v\in V}\|\nabla_v\varphi\|^p.$$

Intuitively, the norm of the graph gradient measures the roughness of a function around a vertex, and the p-Dirichlet form the roughness of a function over the graph. In addition, we define $\|\nabla\varphi([v,u])\| := \|\nabla_v\varphi\|$. Note that $\|\nabla\varphi\|$ is defined in the space $\mathcal{H}(E)$ as $\|\nabla\varphi\| = \langle\nabla\varphi, \nabla\varphi\rangle^{1/2}_{\mathcal{H}(E)}$.

Definition 2. *The graph divergence is an operator* $\operatorname{div} : \mathcal{H}(E) \to \mathcal{H}(V)$ *which satisfies*

$$\langle\nabla\varphi, \psi\rangle_{\mathcal{H}(E)} = \langle\varphi, -\operatorname{div}\psi\rangle_{\mathcal{H}(V)}, \text{ for all } \varphi \in \mathcal{H}(V), \psi \in \mathcal{H}(E). \tag{3}$$

In other words, $-\operatorname{div}$ is defined to be the adjoint of the graph gradient. Eq.(3) can be thought of as discrete analogue of the Stokes' theorem [1]. Note that the inner products in the left and right sides of (3) are respectively in the spaces $\mathcal{H}(E)$ and $\mathcal{H}(V)$. We can show that the graph divergence can be computed by

$$(\operatorname{div}\psi)(v) = \sum_{u\sim v}\sqrt{\frac{w([u,v])}{g(v)}}\Big(\psi([v,u]) - \psi([u,v])\Big). \tag{4}$$

Intuitively, the divergence measures the net outflow of function ψ at each vertex. Note that if ψ is symmetric, then $(\operatorname{div}\psi)(v) = 0$ for all $v \in V$.

[1] Given a compact Riemannian manifold (M, g) with a function $f \in C^\infty(M)$ and a vector field $X \in \mathcal{X}(M)$, it follows from the stokes' theorem that $\int_M\langle\nabla f, X\rangle = -\int_M(\operatorname{div} X)f$.

2.3 Laplace Operator

In this section, we define the graph Laplacian, which can be thought of as discrete analogue of the Laplace-Beltrami operator on Riemannian manifolds.

Definition 3. *The graph Laplacian is an operator $\Delta : \mathcal{H}(V) \to \mathcal{H}(V)$ defined by* [2]

$$\Delta\varphi := -\frac{1}{2}\operatorname{div}(\nabla\varphi). \tag{5}$$

Substituting (1) and (4) into (5), we have

$$(\Delta\varphi)(v) = \varphi(v) - \sum_{u\sim v} \frac{w([u,v])}{\sqrt{g(u)g(v)}}\varphi(u). \tag{6}$$

The graph Laplacian is a linear operator because both the gradient and divergence operators are linear. Furthermore, the graph Laplacian is self-adjoint:

$$\langle \Delta\varphi, \phi\rangle = \frac{1}{2}\langle -\operatorname{div}(\nabla\varphi), \phi\ \rangle = \frac{1}{2}\langle \nabla\varphi, \nabla\phi\ \rangle = \frac{1}{2}\langle \varphi, -\operatorname{div}(\nabla\phi)\rangle = \langle \varphi, \Delta\phi\rangle.$$

and positive semi-definite:

$$\langle \Delta\varphi, \varphi\rangle = \frac{1}{2}\langle -\operatorname{div}(\nabla\varphi), \varphi\ \rangle = \frac{1}{2}\langle \nabla\varphi, \nabla\varphi\rangle = \mathcal{S}_2(\varphi) \geq 0. \tag{7}$$

It immediate follows from (7) that

Theorem 1. $2\Delta\varphi = D_\varphi \mathcal{S}_2$.

Remark 1. Eq. (6) shows that our graph Laplacian defined by (5) is identical to the Laplace matrix in [1] defined to be $D^{-1/2}(D - W)D^{-1/2}$, where D is a diagonal matrix with $D(v, v) = g(v)$, and W is a matrix satisfying $W(u, v) = w([u, v])$ if $[u, v]$ is an edge and $W(u, v) = 0$ otherwise.

2.4 Curvature Operator

In this section, we define the graph curvature as discrete analogue of the mean curvature operator in the continuous case.

Definition 4. *The graph curvature is an operator $\kappa : \mathcal{H}(V) \to \mathcal{H}(V)$ defined by*

$$\kappa\varphi := -\frac{1}{2}\operatorname{div}\left(\frac{\nabla\varphi}{\|\nabla\varphi\|}\right). \tag{8}$$

Substituting (1) and (4) into (8), we obtain

$$(\kappa\varphi)(v) = \frac{1}{2}\sum_{u\sim v} \frac{w([u,v])}{\sqrt{g(v)}}\left(\frac{1}{\|\nabla_u\varphi\|} + \frac{1}{\|\nabla_v\varphi\|}\right)\left(\frac{\varphi(v)}{\sqrt{g(v)}} - \frac{\varphi(u)}{\sqrt{g(u)}}\right). \tag{9}$$

Unlike the graph Laplacian (5), the graph curvature is a non-linear operator.

As Theorem 1, we can show that

Theorem 2. $\kappa\varphi = D_\varphi \mathcal{S}_1$.

[2] The Laplace-Beltrami operator $\Delta : C^\infty(M) \to C^\infty(M)$ is defined to be $\Delta f = -\operatorname{div}(\nabla f)$. The additional factor 1/2 in (5) is due to each edge being counted twice.

2.5 p-Laplace Operator

In this section, we generalize the graph Laplacian and curvature to an operator, which can be thought of as discrete analogue of the p-Laplacian in the continuous case [3, 4].

Definition 5. *The graph p-Laplacian is an operator $\Delta_p : \mathcal{H}(V) \to \mathcal{H}(V)$ defined by*

$$\Delta_p \varphi := -\frac{1}{2} \operatorname{div}(\|\nabla \varphi\|^{p-2} \nabla \varphi). \tag{10}$$

Clearly, $\Delta_1 = \kappa$, and $\Delta_2 = \Delta$. Substituting (1) and (4) into (10), we obtain

$$(\Delta_p \varphi)(v) = \frac{1}{2} \sum_{u \sim v} \frac{w([u,v])}{\sqrt{g(v)}} (\|\nabla_u \varphi\|^{p-2} + \|\nabla_v \varphi\|^{p-2}) \left(\frac{\varphi(v)}{\sqrt{g(v)}} - \frac{\varphi(u)}{\sqrt{g(u)}} \right), \tag{11}$$

which generalizes (6) and (9).

As before, it can be shown that

Theorem 3. $p \Delta_p \varphi = D_\varphi \mathcal{S}_p$.

Remark 2. There is much literature on the p-Laplacian in the continuous case. We refer to [4] for a comprehensive study. There is also some work on discrete analogue of the p-Laplacian, e.g., see [10], where it is defined as

$$\Delta_p \varphi(v) = \frac{1}{g_p(v)} \sum_{u \sim v} w^{p-1}([u,v]) |\varphi(u) - \varphi(v)|^{p-1} \operatorname{sign}(\varphi(u) - \varphi(v)),$$

where $g_p(v) = \sum_{u \sim v} w^{p-1}([u,v])$ and $p \in [2, \infty[$. Note that $p = 1$ is not allowed.

3 Discrete Regularization Framework

Given a graph $G = (V, E)$ and a label set $\mathcal{Y} = \{1, -1\}$, the vertices v in a subset $S \subset V$ are labeled as $y(v) \in \mathcal{Y}$. The problem is to label the remaining unlabeled vertices, i.e., the vertices in the complement of S. Assume a classification function $f \in \mathcal{H}(V)$, which assigns a label $\operatorname{sign} f(v)$ to each vertex $v \in V$. Obviously, a good classification function should vary as slowly as possible between closely related vertices while changing the initial label assignment as little as possible. Define a function $y \in \mathcal{H}(V)$ with $y(v) = 1$ or -1 if vertex v is labeled as positive or negative respectively, and 0 if it is unlabeled. Thus we may consider the optimization problem

$$f^* = \underset{f \in \mathcal{H}(V)}{\operatorname{argmin}} \{\mathcal{S}_p(f) + \mu \|f - y\|^2\}, \tag{12}$$

where $\mu \in]0, \infty[$ is a parameter specifying the trade-off between the two competing terms. It is not hard to see the objective function is strictly convex, and hence by standard arguments in convex analysis the optimization problem has a unique solution.

3.1 Regularization with $p = 2$

When $p = 2$, it follows from Theorem 1 that

Theorem 4. *The solution of (12) satisfies that $\Delta f^* + \mu(f^* - y) = 0$.*

The equation in the theorem can be thought of as discrete analogue of the Euler-Lagrange equation. It is easy to see that we can obtain a closed form solution $f^* = \mu(\Delta + \mu I)^{-1} y$, where I denotes the identity operator. Define the function $c : E \to \mathbb{R}_+$ by

$$c([u,v]) = \frac{1}{1+\mu} \frac{w([u,v])}{\sqrt{g(u)g(v)}}, \text{ if } u \neq v; \text{ and } c([v,v]) = \frac{\mu}{1+\mu}. \tag{13}$$

We can show that the iteration

$$f^{(t+1)}(v) = \sum_{u \sim v} c([u,v]) f^{(t)}(v) + c([v,v]) y(v), \text{ for all } v \in V, \tag{14}$$

where t indicates the iteration step, converges to a closed form solution [11]. Note that the iterative result is independent of the setting of the initial value. The iteration can be thought of as discrete analogue of heat diffusion on Riemannian manifolds [2]. At every step, each node receives the values from its neighbors, which are weighed by the normalized pairwise relationships. At the same time, they also retain some fraction of their values. The relative amount by which these updates occur is specified by the coefficients defined in (13).

3.2 Regularization with $p = 1$

When $p = 1$, it follows from Theorem 2 that

Theorem 5. *The solution of (12) satisfies that $\kappa f^* + 2\mu(f^* - y) = 0$.*

As we have mentioned before, the curvature κ is a non-linear operator, and we are not aware of any closed form solution for this equation. However, we can construct an iterative algorithm to obtain the solution. Substituting (9) into the equation in the theorem, we have

$$\sum_{u \sim v} \frac{w([u,v])}{\sqrt{g(v)}} \left(\frac{1}{\|\nabla_u f^*\|} + \frac{1}{\|\nabla_v f^*\|} \right) \left(\frac{f^*(v)}{\sqrt{g(v)}} - \frac{f^*(u)}{\sqrt{g(u)}} \right) + 2\mu(f^*(v) - y(v)) = 0. \tag{15}$$

Define the function $m : E \to \mathbb{R}_+$ by

$$m([u,v]) = w([u,v]) \left(\frac{1}{\|\nabla_u f^*\|} + \frac{1}{\|\nabla_v f^*\|} \right). \tag{16}$$

Then

$$\sum_{u \sim v} \frac{m([u,v])}{\sqrt{g(v)}} \left(\frac{f^*(v)}{\sqrt{g(v)}} - \frac{f^*(u)}{\sqrt{g(u)}} \right) + 2\mu(f^*(v) - y(v)) = 0,$$

which can be transformed into

$$\left(\sum_{u\sim v}\frac{m([u,v])}{g(v)}+2\mu\right)f^*(v)=\sum_{u\sim v}\frac{m([u,v])}{\sqrt{g(u)g(v)}}f^*(u)+2\mu y(v).$$

Define the function $c: E\to\mathbb{R}_+$ by

$$c([u,v])=\frac{\dfrac{m([u,v])}{\sqrt{g(u)g(v)}}}{\sum\limits_{u\sim v}\dfrac{m([u,v])}{g(v)}+2\mu},\ \text{if } u\neq v;\ \text{and } c([v,v])=\frac{2\mu}{\sum\limits_{u\sim v}\dfrac{m([u,v])}{g(v)}+2\mu}. \tag{17}$$

Then

$$f^*(v)=\sum_{u\sim v}c([u,v])f^*(v)+c([v,v])y(v). \tag{18}$$

Thus we can use the iteration

$$f^{(t+1)}(v)=\sum_{u\sim v}c^{(t)}([u,v])f^{(t)}(v)+c^{(t)}([v,v])y(v),\ \text{for all } v\in V \tag{19}$$

to obtain the solution, in which the coefficients $c^{(t)}$ are updated according to (17) and (16). This iterative result is independent of the setting of the initial value. Compared with the iterative algorithm (14) in the case of $p=2$, the coefficients in the present method are adaptively updated at each iteration, in addition to the function being updated.

3.3 Regularization with Arbitrary p

For arbitrary p, it follows from Theorem 3 that

Theorem 6. *The solution of (12) satisfies that* $p\Delta_p f^*+2\mu(f^*-y)=0$.

We can construct a similar iterative algorithm to obtain the solution. Specifically,

$$f^{(t+1)}(v)=\sum_{u\sim v}c^{(t)}([u,v])f^{(t)}(v)+c^{(t)}([v,v])y(v),\ \text{for all } v\in V, \tag{20}$$

where

$$c^{(t)}([u,v])=\frac{\dfrac{m^{(t)}([u,v])}{\sqrt{g(u)g(v)}}}{\sum\limits_{u\sim v}\dfrac{m^{(t)}([u,v])}{g(v)}+\dfrac{2\mu}{p}},\ \text{if } u\neq v;\ \text{and } c^{(t)}([v,v])=\frac{\dfrac{2\mu}{p}}{\sum\limits_{u\sim v}\dfrac{m^{(t)}([u,v])}{g(v)}+\dfrac{2\mu}{p}}, \tag{21}$$

and

$$m^{(t)}([u,v])=\frac{w([u,v])}{p}\left(\|\nabla_u f^{(t)}\|^{p-2}+\|\nabla_v f^{(t)}\|^{p-2}\right). \tag{22}$$

It is easy to see that the iterative algorithms in Sections 3.1 and 3.2 are the special cases of this algorithm with $p=2$ and $p=1$ respectively. Moreover, it is worth noticing that $p=2$ is a critical point.

4 Conclusions and Future Work

We have developed discrete analysis and geometry on graphs, and have constructed a general discrete regularization framework. A family of transductive inference algorithms was derived from the framework, including the algorithm we proposed earlier [11], which can substantially benefit from large amounts of available unlabeled data in many practical problems. There are many possible extensions to this work. One may consider defining discrete high-order differential operators, and then building a regularization framework that can penalize high-order derivatives. One may also develop a parallel framework on directed graphs [12], which model many real-world data structures, such as the World Wide Web. Finally, it is of interest to explore the properties of the graph p-Laplacian as the nonlinear extension of the usual graph Laplacian, since the latter has been intensively studied, and has many nice properties [1].

References

1. F. Chung. *Spectral Graph Theory.* Number 92 in CBMS-NSF Regional Conference Series in Mathematics. SIAM, 1997.
2. J. Eells and J.H. Sampson. Harmonic mappings of Riemannian manifolds. *American Journal of Mathematics*, 86:109–160, 1964.
3. R. Hardt and F.H. Lin. Mappings minimizing the L^p norm of the gradient. *Communications on Pure and Applied Mathematics*, 40:556–588, 1987.
4. J. Heinonen, T. Kilpeläinen, and O. Martio. *Nonlinear Potential Theory of Degenerate Elliptic Equations.* Oxford University Press, Oxford, 1993.
5. R. Jensen. Uniqueness of Lipschitz extensions: minimizing the sup-norm of the gradient. *Arch. Rat. Mech. Anal.*, 123(1):51–74, 1993.
6. J. Jost. *Riemannian Geometry and Geometric Analysis.* Springer-Verlag, Berlin-Heidelberg, third edition, 2002.
7. A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill-posed Problems.* W. H. Winston, Washington, DC, 1977.
8. V.N. Vapnik. *Statistical Learning Theory.* Wiley, NY, 1998.
9. G. Wahba. *Spline Models for Observational Data.* Number 59 in CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, 1990.
10. M. Yamasaki. Ideal boundary limit of discrete Dirichlet functions. *Hiroshima Math. J.*, 16(2):353–360, 1986.
11. 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.
12. D. Zhou, B. Schölkopf, and T. Hofmann. Semi-supervised learning on directed graphs. In *Advances in Neural Information Processing Systems 17.* MIT Press, Cambridge, MA, 2005.