# 1 Discrete Regularization

**Dengyong Zhou**
**Bernhard Schölkopf**

*Many real-world machine learning problems are situated on finite discrete sets, including dimensionality reduction, clustering, and transductive inference. A variety of approaches for learning from finite sets has been proposed from different motivations and for different problems. In most of those approaches, a finite set is modeled as a graph, in which the edges encode pairwise relationships among the objects in the set. Consequently many concepts and methods from graph theory are adopted. In particular, the graph Laplacian is widely used.*

*In this chapter we present a systemic framework for learning from a finite set represented as a graph. We develop discrete analogues of a number of differential operators, and then construct a discrete analogue of classical regularization theory based on those discrete differential operators. The graph Laplacian based approaches are special cases of this general discrete regularization framework. An important thing implied in this framework is that we have a wide choices of regularization on graph in addition to the widely-used graph Laplacian based one.*

## 1.1 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. Typically we may assume that there exist pairwise relationships among data. For examples, given a finite set of vectorial data, the pairwise relationships among data points may be described by a kernel [10]. A dataset endowed with pairwise relationships can be naturally modeled as a weighted graph. The vertices of the graph represent the objects, and the weighted edges encode the pairwise relationships. If the pairwise relationships are symmetric, the graph is undirected; otherwise, the graph is directed. A typical example for directed graphs is the World Wide Web (WWW), in which hyperlinks between web pages

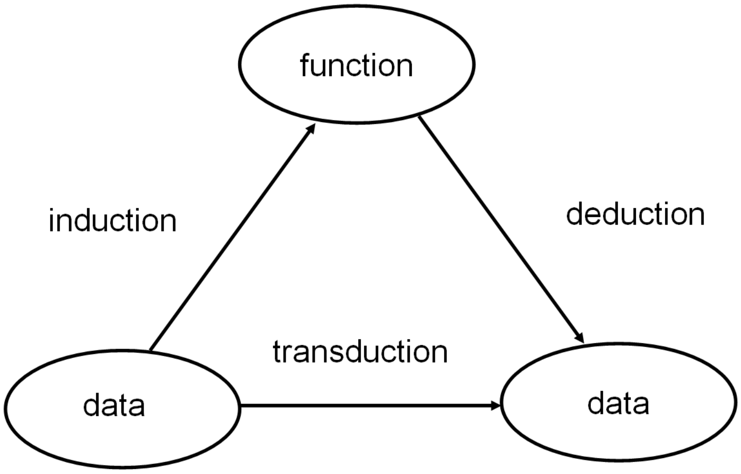

**Figure 1.1** The relations among induction, deduction and transduction.

may be thought of as directed edges.

Any supervised learning algorithm can be applied to the above inference problem, e.g., by training a classifier $f : \mathcal{X} \to \mathcal{Y}$ with 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 [13](see also Chap. 24), 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 a better approach is to directly predict the labels of the given unlabeled objects. Therefore here we consider estimating a discrete classification function which is defined on the given objects $X$ only. Such an estimation problem is called *transductive inference* [13]. In psychology, transductive reasoning means linking particular to particular with no consideration of the general principles. It is generally used by young children. In contrast, deductive reasoning, which is used by used by adults and older children, means the ability to come to a specific conclusion based on a general premise.

It is well known that many meaningful inductive methods such as Support Vector Machines (SVMs) can be derived from a regularization framework, which minimizes an empirical loss plus a regularization term. Inspired by this work, we define discrete analogues of a number of differential operators, and then construct a discrete analogue of classical regularization theory [12, 15] using the discrete operators. Much existing work including spectral clustering, transductive inference and dimensionality reduction can be understood in this framework. More importantly, a family of new approaches is derived.

## 1.2 Discrete Analysis and Differential Geometry

In this section, we first introduce some basic notions on graph theory, and then propose a family of discrete differential operators, which constitute the basis of the discrete regularization framework introduced in the next section.

### 1.2.1 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 $(v_1, v_2, \ldots, v_m)$ 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 $\mathcal{H}(E)$. In what follows, we will omit the subscript of inner products if we do not think it is necessary. Note that function $h \in \mathcal{H}(E)$ have not to be symmetric. In other words, we do not require $h([u, v]) = h([v, u])$.

### 1.2.2 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 1.1** *The graph gradient is an operator* $\nabla : \mathcal{H}(V) \to \mathcal{H}(E)$ *defined by*

$$(\nabla \varphi)([u, v]) := \sqrt{\frac{w([u, v])}{g(v)}} \varphi(v) - \sqrt{\frac{w([u, v])}{g(u)}} \varphi(u), \text{ for all } [u, v] \in E. \quad (1.1)$$

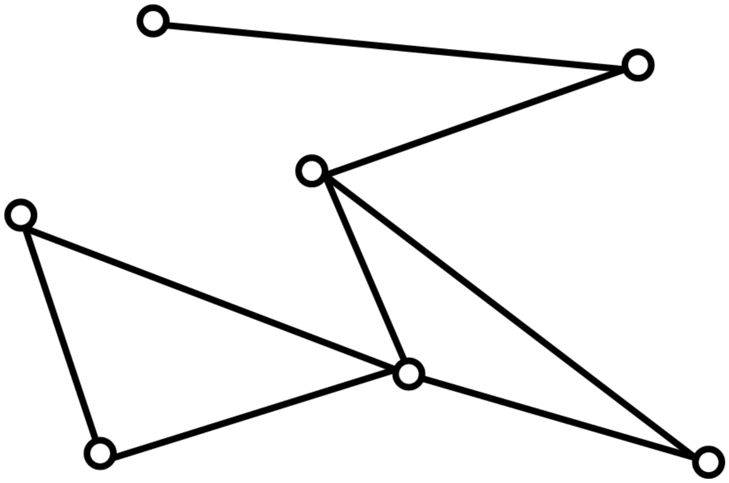

**Figure 1.2** An undirected graph.

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

$$(\nabla\varphi)([u,v]) = -(\nabla\varphi)([v,u]), \tag{1.2}$$

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

**Remark 1.2** *A obvious problem is why we define the graph gradient as equation 1.1. In the uniform 2-dimensional lattice case, one may naturally define the discrete gradient as*

$$(\nabla\varphi)([i,i+1]) = f(i+1) - f(i),$$

*where $i$ denotes the index of a node of the lattice. The problem that we have to deal with here is the irregularity of a general graph. Intuitively, in our definition, before computing the variation of a function between two adjacent vertices, we split the function value at each vertex along its adjacent edges before according to a certain proportion based on the weights. Formally, this definition is the only choice for recovering the well-known graph Laplacian in the way shown in the following sections. 1.2.3*

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 $\varphi$ 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 $\varphi$ 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 $\|\nabla\varphi\| = \langle\nabla\varphi,\nabla\varphi\rangle_{\mathcal{H}(E)}^{1/2}$.

**Definition 1.3** *The graph divergence is an operator* div : $\mathcal{H}(E) \to \mathcal{H}(V)$ *which satisfies*

$$\langle\nabla\varphi,\psi\rangle_{\mathcal{H}(E)} = \langle\varphi, -\operatorname{div}\psi\rangle_{\mathcal{H}(V)},\ \textit{for all}\ \varphi\in\mathcal{H}(V), \psi\in\mathcal{H}(E). \tag{1.3}$$

In other words, $-\operatorname{div}$ is defined to be the adjoint of the graph gradient. Eq.(1.3) can be thought of as discrete analogue of the Stokes' theorem [1]. Note that the inner products in the left and right sides of (1.3) are respectively in the spaces $\mathcal{H}(E)$ and $\mathcal{H}(V)$.

**Proposition 1.4** *The graph divergence can be computed as*

$$(\operatorname{div}\psi)(v) = \sum_{u\sim v}\sqrt{\frac{w([u,v])}{g(v)}}\Big(\psi([v,u]) - \psi([u,v])\Big), \tag{1.4}$$

***Proof***

$$\begin{aligned}
\langle\nabla\varphi,\psi\rangle &= \sum_{[u,v]\in E}\nabla\varphi([u,v])\psi([u,v])\\
&= \sum_{[u,v]\in E}\Bigg(\sqrt{\frac{w([u,v])}{g(v)}}\varphi(v) - \sqrt{\frac{w([u,v])}{g(u)}}\varphi(u)\Bigg)\psi([u,v])\\
&= \sum_{[u,v]\in E}\sqrt{\frac{w([u,v])}{g(v)}}\varphi(v)\psi([u,v]) - \sum_{[u,v]\in E}\sqrt{\frac{w([u,v])}{g(u)}}\varphi(u)\psi([u,v])\\
&= \sum_{r\in V}\sum_{u\sim r}\sqrt{\frac{w([u,r])}{g(r)}}\varphi(r)\psi([u,r]) - \sum_{r\in V}\sum_{v\sim r}\sqrt{\frac{w([r,v])}{g(r)}}\varphi(r)\psi([r,v])\\
&= \sum_{r\in V}\varphi(r)\Bigg(\sum_{u\sim r}\sqrt{\frac{w([u,r])}{g(r)}}\psi([u,r]) - \sum_{v\sim r}\sqrt{\frac{w([r,v])}{g(r)}}\psi([r,v])\Bigg)\\
&= \sum_{r\in V}\varphi(r)\sum_{u\sim r}\sqrt{\frac{w([u,r])}{g(r)}}\Big(\psi([u,r]) - \psi([r,u])\Big).
\end{aligned}$$

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

The last equality implies (1.4). ■

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

### 1.2.3 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 1.5** *The graph Laplacian is an operator $\Delta : \mathcal{H}(V) \to \mathcal{H}(V)$ defined by* 2

$$\Delta\varphi := -\frac{1}{2}\operatorname{div}(\nabla\varphi). \tag{1.5}$$

Substituting (1.1) and (1.4) into (1.5), we have

$$\begin{aligned}(\Delta\varphi)(v) &= \frac{1}{2}\sum_{u\sim v}\sqrt{\frac{w([u,v])}{g(v)}}\Big((\nabla\varphi)([u,v]) - (\nabla\varphi)([v,u])\Big)\\ &= \sum_{u\sim v}\sqrt{\frac{w([u,v])}{g(v)}}\left(\sqrt{\frac{w([u,v])}{g(v)}}\varphi(v) - \sqrt{\frac{w([u,v])}{g(u)}}\varphi(u)\right)\\ &= \varphi(v) - \sum_{u\sim v}\frac{w([u,v])}{\sqrt{g(u)g(v)}}\varphi(u). \end{aligned}\tag{1.6}$$

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:

$$\langle\Delta\varphi,\varphi\rangle = \frac{1}{2}\langle -\operatorname{div}(\nabla\varphi),\varphi\ \rangle = \frac{1}{2}\langle\nabla\varphi,\nabla\varphi\rangle = \mathcal{S}_2(\varphi) \geq 0. \tag{1.7}$$

It immediate follows from (1.7) that

**Theorem 1.6** $2\Delta\varphi = D_\varphi\mathcal{S}_2$.

**Remark 1.7** *Equation (1.6) shows that our graph Laplacian defined by (1.5) is identical to the Laplace matrix in [3] 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. The matrix $L = D - W$ is often referred to as the combinatorial or unnormalized graph Laplacian. It can also*

[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 (1.5) is due to each edge being counted twice.

*be derived in a similar way. Specifically, define a graph gradient by*

$$(\nabla\varphi)([u,v]) := \sqrt{w([u,v])}(\varphi(v)-\varphi(u)), \ \textit{for all } [u,v]\in E,$$

*and then the rest proceeds as the above.*

**Remark 1.8** *For the connection between graph Laplacians (including the Laplacian we presented here) and the usual Laplacian in the continuous case, we may refer the readers to [14, 6, 2]. The main point is that, if we assume the vertices of a graph are sampled from some distribution, then the combinatorial graph Laplacian does not converge to the usual Laplacian when the sampling size goes to infinity unless the distribution is uniform.*

### 1.2.4 Curvature Operator

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

**Definition 1.9** *The graph curvature is an operator $\kappa : \mathcal{H}(V) \to \mathcal{H}(V)$ defined by*

$$\kappa\varphi := -\frac{1}{2}\operatorname{div}\left(\frac{\nabla\varphi}{\|\nabla\varphi\|}\right). \tag{1.8}$$

Substituting (1.1) and (1.4) into (1.8), we obtain

$$\begin{aligned}(\kappa\varphi)(v) &= \sum_{u\sim v}\sqrt{\frac{w([u,v])}{g(v)}}\left(\frac{\nabla\varphi}{\|\nabla\varphi\|}([u,v]) - \frac{\nabla\varphi}{\|\nabla\varphi\|}([v,u])\right)\\ &= \sum_{u\sim v}\frac{w([u,v])}{\sqrt{g(v)}}\left[\frac{1}{\|\nabla_u\varphi\|}\left(\frac{\varphi(v)}{\sqrt{g(v)}} - \frac{\varphi(u)}{\sqrt{g(u)}}\right) - \frac{1}{\|\nabla_v\varphi\|}\left(\frac{\varphi(u)}{\sqrt{g(u)}} - \frac{\varphi(v)}{\sqrt{g(v)}}\right)\right]\\ &= \frac{1}{2}\sum_{u\sim v}\frac{w([u,v])}{\sqrt{g(v)}}\left(\frac{1}{\|\nabla_u\varphi\|} + \frac{1}{\|\nabla_v\varphi\|}\right)\left(\frac{\varphi(v)}{\sqrt{g(v)}} - \frac{\varphi(u)}{\sqrt{g(u)}}\right). \qquad (1.9)\end{aligned}$$

Unlike the graph Laplacian (1.5), the graph curvature is a non-linear operator.

As in Theorem 1.6, we have

**Theorem 1.10** $\kappa\varphi = D_\varphi \mathcal{S}_1$.

***Proof***

$$\begin{aligned}(D_\varphi\mathcal{S}_1)(v) &= \sum_{u\sim v}\left[\frac{w([u,v])}{\|\nabla_u\varphi\|}\left(\frac{\varphi(v)}{g(v)} - \frac{\varphi(u)}{\sqrt{g(u)g(v)}}\right) + \frac{w([u,v])}{\|\nabla_v\varphi\|}\left(\frac{\varphi(v)}{g(v)} - \frac{\varphi(u)}{\sqrt{g(u)g(v)}}\right)\right]\\ &= \sum_{u\sim v} w([u,v])\left(\frac{1}{\|\nabla_u\varphi\|} + \frac{1}{\|\nabla_v\varphi\|}\right)\left(\frac{\varphi(v)}{g(v)} - \frac{\varphi(u)}{\sqrt{g(u)g(v)}}\right)\\ &= \sum_{u\sim v}\frac{w([u,v])}{\sqrt{g(v)}}\left(\frac{1}{\|\nabla_u\varphi\|} + \frac{1}{\|\nabla_v\varphi\|}\right)\left(\frac{\varphi(v)}{\sqrt{g(v)}} - \frac{\varphi(u)}{\sqrt{g(u)}}\right).\end{aligned}$$

Comparing the last equality with (1.9) completes the proof. ■

### 1.2.5 $p$-Laplace Operator

In this section, we generalize the graph Laplacian and curvature to an operator, which can be thought of as the discrete analogue of the $p$-Laplacian in the continuous case [5, 7].

**Definition 1.11** *The graph $p$-Laplacian is an operator $\Delta_p : \mathcal{H}(V) \to \mathcal{H}(V)$ defined by*

$$\Delta_p \varphi := -\frac{1}{2} \operatorname{div}(\|\nabla\varphi\|^{p-2}\nabla\varphi). \tag{1.10}$$

Clearly, $\Delta_1 = \kappa$, and $\Delta_2 = \Delta$. Substituting (1.1) and (1.4) into (1.10), we obtain

$$(\Delta_p\varphi)(v) = \frac{1}{2}\sum_{u\sim v} \frac{w([u,v])}{\sqrt{g(v)}}(\|\nabla_u\varphi\|^{p-2} + \|\nabla_v\varphi\|^{p-2})\left(\frac{\varphi(v)}{\sqrt{g(v)}} - \frac{\varphi(u)}{\sqrt{g(u)}}\right), \tag{1.11}$$

which generalizes (1.6) and (1.9).

As before, it can be shown that

**Theorem 1.12** $p\Delta_p\varphi = D_\varphi \mathcal{S}_p$.

**Remark 1.13** *There is much literature on the $p$-Laplacian in the continuous case. We refer to [7] for a comprehensive study. There is also some work on discrete analogue of the $p$-Laplacian, e.g., see [16], 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.*

## 1.3 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 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

$$f^* = \underset{f\in\mathcal{H}(V)}{\operatorname{argmin}}\{\mathcal{S}_p(f) + \mu\|f-y\|^2\}, \tag{1.12}$$

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.

### 1.3.1 Regularization with $p = 2$

When $p = 2$, it follows from Theorem 1.6 that

**Theorem 1.14** *The solution of (1.12) 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

$$c([u,v]) = \frac{1}{1+\mu} \frac{w([u,v])}{\sqrt{g(u)g(v)}}, \text{ if } u \neq v; \text{ and } c([v,v]) = \frac{\mu}{1+\mu}. \tag{1.13}$$

We can show that the iteration

$$f^{(t+1)}(v) = \sum_{u \sim v} c([u,v]) f^{(t)}(v) + c([v,v]) y(v), \text{ for all } v \in V, \tag{1.14}$$

where $t$ indicates the iteration step, converges to a closed form solution [17]. Moreover, the iterative result is independent of the setting of the initial value. The iteration can be intuitively thought of as sort of information diffusion. 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 (1.13). In what follows, this iteration approach will be generalized to arbitrary $p$.

**Remark 1.15** *It is easy to see that the regularizer of* $p = 2$ *can be rewritten into*

$$\frac{1}{2} \sum_{u,v} w([u,v]) \left( \frac{f(u)}{\sqrt{g(u)}} - \frac{f(v)}{\sqrt{g(v)}} \right)^2, \tag{1.15}$$

*which we earlier suggested for transductive inference [17]. A closely related one is*

$$\frac{1}{2} \sum_{u,v} w([u,v]) (f(u) - f(v))^2, \tag{1.16}$$

*which appeared in [8, 1, 19]. From the point of view of spectral clustering, the former regularizer is derived from the normalized cut [11], and the later is derived from the ratio-cut [4].*

**Remark 1.16** *One can construct many other similar regularizers. For instance,* *one might consider [9]*

$$\frac{1}{2}\sum_{u,v}\left(f(v)-\sum_{u\sim v}p([u,v])f(u)\right)^2, \tag{1.17}$$

*where the function $p : E \rightarrow \mathbb{R}_+$ is defined to be $p([u,v]) = w([u,v])/g(u)$. Note that $p$ is not symmetric. This regularizer measures the difference of function $f$ at vertex $v$, and the average of $f$ at the neighbors of $v$.*

### 1.3.2 Regularization with $p = 1$

When $p = 1$, it follows from Theorem 1.10 that

**Theorem 1.17** *The solution of (1.12) satisfies that $\kappa f^* + 2\mu(f^* - y) = 0$.*

As we have mentioned before, the curvature $\kappa$ 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 (1.9) into the equation in the theorem, we have

$$\sum_{u\sim v}\frac{w([u,v])}{\sqrt{g(v)}}\left(\frac{1}{\|\nabla_u f^*\|}+\frac{1}{\|\nabla_v f^*\|}\right)\left(\frac{f^*(v)}{\sqrt{g(v)}}-\frac{f^*(u)}{\sqrt{g(u)}}\right)+2\mu(f^*(v)-y(v))=0. \tag{1.18}$$

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

$$m([u,v]) = w([u,v])\left(\frac{1}{\|\nabla_u f^*\|}+\frac{1}{\|\nabla_v f^*\|}\right). \tag{1.19}$$

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 $c : E \rightarrow \mathbb{R}_+$ by

$$c([u,v]) = \frac{\dfrac{m([u,v])}{\sqrt{g(u)g(v)}}}{\sum\limits_{u\sim v}\dfrac{m([u,v])}{g(v)}+2\mu}, \text{ if } u \neq v; \text{ and } c([v,v]) = \frac{2\mu}{\sum\limits_{u\sim v}\dfrac{m([u,v])}{g(v)}+2\mu}. \tag{1.20}$$

Then

$$f^*(v) = \sum_{u\sim v}c([u,v])f^*(v) + c([v,v])y(v). \tag{1.21}$$

Thus we can use the iteration

$$f^{(t+1)}(v) = \sum_{u \sim v} c^{(t)}([u,v]) f^{(t)}(v) + c^{(t)}([v,v]) y(v), \text{ for all } v \in V \tag{1.22}$$

to obtain the solution, in which the coefficients $c^{(t)}$ are updated according to (1.20) and (1.19). This iterative result is independent of the setting of the initial value. Compared with the iterative algorithm (1.14) 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.

### 1.3.3 Regularization with Arbitrary $p$

For arbitrary $p$, it follows from Theorem 1.12 that

**Theorem 1.18** *The solution of (1.12) satisfies that* $p\Delta_p f^* + 2\mu(f^* - y) = 0$.

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

$$f^{(t+1)}(v) = \sum_{u \sim v} c^{(t)}([u,v]) f^{(t)}(v) + c^{(t)}([v,v]) y(v), \text{ for all } v \in V, \tag{1.23}$$

where

$$c^{(t)}([u,v]) = \frac{\dfrac{m^{(t)}([u,v])}{\sqrt{g(u)g(v)}}}{\sum\limits_{u \sim v} \dfrac{m^{(t)}([u,v])}{g(v)} + \dfrac{2\mu}{p}}, \text{ if } u \neq v; \text{ and } c^{(t)}([v,v]) = \frac{\dfrac{2\mu}{p}}{\sum\limits_{u \sim v} \dfrac{m^{(t)}([u,v])}{g(v)} + \dfrac{2\mu}{p}}, \tag{1.24}$$

and

$$m^{(t)}([u,v]) = \frac{w([u,v])}{p} \left( \|\nabla_u f^{(t)}\|^{p-2} + \|\nabla_v f^{(t)}\|^{p-2} \right). \tag{1.25}$$

It is easy to see that the iterative algorithms in Sections 1.3.1 and 1.3.2 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.

## 1.4 Conclusion

In this chapter, we proposed the discrete analogues of a family of differential operators, and the discrete analogue of classical regularization theory based on those discrete differential operators. A family of transductive inference algorithms corresponding different discrete differential operators was naturally derived from the discrete regularization framework.

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 [18], 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 [3].

# References

1. M. Belkin, I. Matveeva, and P. Niyogi. Regression and regularization on large graphs. In *Proc. 17th Annual Conf. on Learning Theory*, 2004.
2. O. Bousquet, O. Chapelle, and M. Hein. Measure based regularization. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.
3. F. Chung. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, Providence, RI, 1997.
4. L. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE. Trans. on Computed Aided Desgin*, 11:1074–1085, 1992.
5. 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.
6. M. Hein, J. Audibert, and U. von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph laplacians. In *Proc. 18th Conf. on Learning Theory*, pages 470–485, 2005.
7. J. Heinonen, T. Kilpeläinen, and O. Martio. *Nonlinear Potential Theory of Degenerate Elliptic Equations*. Oxford University Press, Oxford, 1993.
8. T. Joachims. Transductive learning via spectral graph partitioning. In *Proc. 20th Intl. Conf. on Machine Learning*, 2003.
9. S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
10. B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
11. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
12. A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill-posed Problems*. W. H. Winston, Washington, DC, 1977.
13. V.N. Vapnik. *Statistical Learning Theory*. Wiley, NY, 1998.
14. U. von Luxburg, O. Bousquet, and M. Belkin. Limits of spectral clustering. In *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

15. G. Wahba. *Spline Models for Observational Data.* Number 59 in CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia, 1990.

16. M. Yamasaki. Ideal boundary limit of discrete Dirichlet functions. *Hiroshima Math. J.*, 16(2):353–360, 1986.

17. 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.

18. 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.

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