11 The Geometric Basis of Semi-supervised Learning

Vikas Sindhwani
Misha Belkin
Partha Niyogi

In this chapter, we present an algorithmic framework for semi-supervised inference based on geometric properties of probability distributions. Our approach brings together Laplacian-based spectral techniques, regularization with kernel methods, and algorithms for manifold learning. This framework provides a natural semi-supervised extension for kernel methods and resolves the problem of out-of-sample inference in graph-based transduction. We discuss an interpretation in terms of a family of globally defined data-dependent kernels and also address unsupervised learning (clustering and data representation) within the same framework. Our algorithms effectively exploit both manifold and cluster assumptions to demonstrate state-of-the-art performance on various classification tasks. This chapter also reviews other recent work on out-of-sample extension for transductive graph-based methods.

11.1 Introduction

We start by providing some intuitions for the geometric basis of semi-supervised learning. These intuitions are demonstrated in pictures (Figures 1,2 and 3).

Consider first the two labeled points (marked "+" and "-") in the left panel of Figure 1. Our intuition may suggest that a simple linear separator such as the one shown in Figure 1, is an optimal choice for a classifier. Indeed, considerable effort in learning theory has been invested into deriving optimality properties for such a classification boundary.

The right panel however shows that the two labeled points are in fact located on two concentric circles of unlabeled data. Looking at the right panel, it becomes clear that the circular boundary is more natural given unlabeled data.

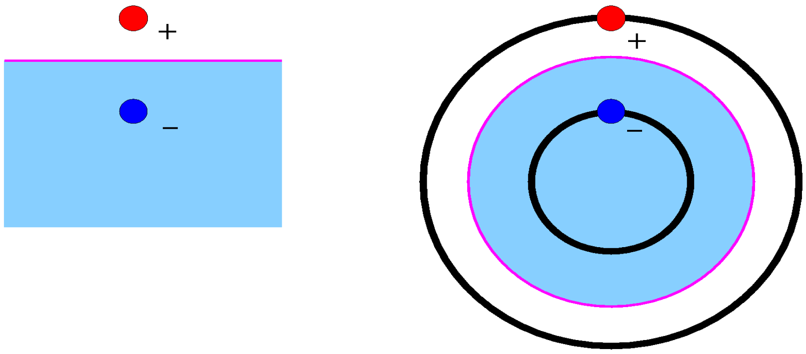


Figure 11.1 Circle

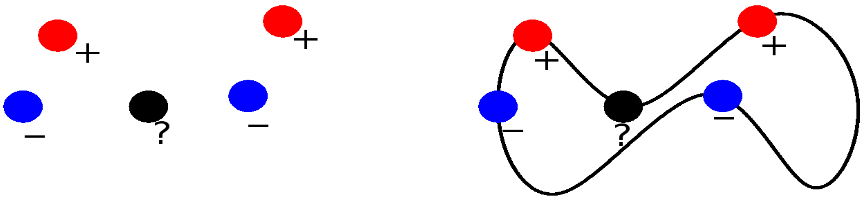


Figure 11.2 Curve

Consider now the left panel in Figure 2. In the absence of unlabeled data the black dot (marked "?") is likely to be classified as blue (marked "-"). The unlabeled data, however, makes classifying it as red (marked "+") seem much more reasonable.

A third example is shown in Figure 3. In the left panel, the unlabeled point may be classified as blue (-) to agree with its nearest neighbor. However, unlabeled data shown as grey clusters in the right panel changes our belief.

Geometry of Unlabeled data

These examples show how the geometry of unlabeled data may radically change our intuition about classifier boundaries. We seek to translate these intuitions into a framework for learning from labeled and unlabeled examples.

Recall now the standard setting of learning from examples. Given a pattern space $\mathcal{X}$, there is a probability distribution $\mathcal{P}$ on $\mathcal{X} \times \mathbb{R}$ according to which examples are generated for function learning. Labeled examples are (x, y) pairs drawn according to $\mathcal{P}$. Unlabeled examples are simply $x \in \mathcal{X}$ sampled according to the marginal distribution $\mathcal{P}_{\mathcal{X}}$ of $\mathcal{P}$.

As we have seen, the knowledge of the marginal $\mathcal{P}_X$ can be exploited for better function learning (e.g., in classification or regression tasks). On the other hand, if there is no identifiable relation between $\mathcal{P}_X$ and the conditional $\mathcal{P}(y|x)$, the knowledge of $\mathcal{P}_X$ is unlikely to be of use.

Two possible connections between $\mathcal{P}_X$ and $\mathcal{P}(y|x)$ can be stated as the following important assumptions (also see the tutorial introduction in Chapter 1 for related discussion):

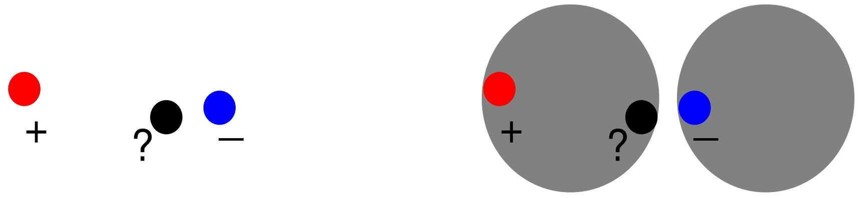


Figure 11.3 Blobs

Assumptions for Semi-supervised Learning

1. **Manifold Assumption:** Suppose that the marginal probability distribution underlying the data is supported on a low-dimensional manifold. Then the family of conditional distributions $P(y|x)$ is smooth, as a function of x, with respect to the underlying structure of the manifold.

2. **Cluster assumption:** The probability distribution $\mathcal{P}$ is such that points in the same "cluster" are likely to have the same label.

We see that the data shown in Figures 1 and 2 satisfy the manifold assumption.

The picture in Figure 3 is meant to show Gaussian clusters. The concentric circles in Figure 1 can also be thought as "clusters", although such clusters are highly non-Gaussian and have an interesting geometric structure. One may conjecture that many clusters in real-world datasets have such non-Gaussian structures. This is evidenced, for example, by frequent superiority of spectral clustering over more traditional methods such as k-means.

In many natural situations, it is clear that the data is supported on a low-dimensional manifold. This is often the case when points are generated by some physical process. For example, in speech production the articulatory organs can be modeled as a collection of tubes. The space of speech sounds is therefore a low-dimensional manifold parameterized by lengths and widths of the tubes. Photographs of an object from various angles form a 3-dimensional submanifold of the image space. In other cases, such as in text retrieval tasks, it may be less clear whether a low-dimensional manifold is present. However, even then, and also for almost any imaginable source of meaningful high-dimensional data, the space of possible configurations occupies only a tiny portion of the total volume available. One therefore suspects that a nonlinear low-dimensional manifold may yield a useful approximation to this structure.

To proceed with our discussion, we will make a specific assumption about the connection between the marginal and the conditional distributions. We will assume that if two points $x_1, x_2 \in \mathcal{X}$ are *close* in the *intrinsic* geometry of $\mathcal{P}_{\mathcal{X}}$, then

Smoothness with respect to marginal distribution

the conditional distributions $\mathcal{P}(y|x_1)$ and $\mathcal{P}(y|x_2)$ are similar. In other words, the conditional probability distribution $\mathcal{P}(y|x)$ varies smoothly along the geodesics in the intrinsic geometry of $\mathcal{P}_{\mathcal{X}}$. A more formal statement for this smoothness property is that $\int \|\nabla \mathcal{P}(y|x)\|^2 d\mu_{\mathcal{X}}$ is small, where μ is the probability distribution over the manifold. That last quantity can be rewritten as $\langle \mathcal{L}\mathcal{P}(y|x), \mathcal{P}(y|x)\rangle$, where $\mathcal{L}$ is the weighted Laplacian associated to probability measure μ. We will elaborate on these objects later in the chapter.

We will introduce a new framework for data-dependent regularization that exploits the geometry of the probability distribution. It is important to note that the resulting algorithms will take into account both manifold and cluster assumption. While this framework allows us to approach the full range of learning problems from unsupervised to supervised, we focus on the problem of semi-supervised learning. This chapter gathers material from [3, 5, 19, 20].

11.2 Incorporating Geometry in Regularization

We will now assume that the marginal distribution $\mathcal{P}_{\mathcal{X}}$ is supported on a low-dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^N$. We will be interested in constructing spaces of functions which are attuned to the geometric structure of $\mathcal{P}_{\mathcal{X}}$. More specifically we will want to control the gradient of the functions of interest with respect to the measure $\mathcal{P}_{\mathcal{X}}$: $\int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 d\mathcal{P}_{\mathcal{X}}$. Here the gradient is taken with respect to the underlying Riemannian manifold $\mathcal{M}$ and the integral is weighted by the measure on that manifold.

Laplace-Beltrami operator

If the manifold $\mathcal{M}$ has no boundary or if the probability distribution $\mathcal{P}_{\mathcal{X}}$ vanishes at the boundary, it can be shown that

$$\int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 d\mathcal{P}_{\mathcal{X}} = \int_{\mathcal{M}} f \mathcal{L}_{\mathcal{P}_{\mathcal{X}}}(f) d\mathcal{P}_{\mathcal{X}} = \langle f, \mathcal{L}_{\mathcal{P}_{\mathcal{X}}}(f)\rangle_{L^2(\mathcal{P}_X)}$$

where $\nabla_{\mathcal{M}}$ is the gradient on $\mathcal{M}$ and $\mathcal{L}_{\mathcal{P}_{\mathcal{X}}}$ is the weighted Laplace-Beltrami operator associated to measure $\mathcal{P}_{\mathcal{X}}$. This operator is key in penalizing functions according to the intrinsic geometry of the probability distribution $\mathcal{P}_{\mathcal{X}}$.

We utilize these geometric intuitions to extend an established framework for function learning. A number of popular algorithms such as SVM, ridge regression, splines, radial basis functions may be broadly interpreted as regularization algorithms with different empirical cost functions and complexity measures in an appropriately chosen Reproducing Kernel Hilbert Space (RKHS) [17, 22, 18].

Recall that for a Mercer kernel $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$, there is an associated RKHS $\mathcal{H}_K$ of functions $\mathcal{X} \to \mathbb{R}$ with the corresponding norm $\| \|_K$. Given a set of labeled examples (x_i, y_i), $i = 1, \ldots, l$ the standard framework estimates an unknown function by minimizing

Learning in RKHS

$$f^* = \operatorname*{argmin}_{f\in\mathcal{H}_K} \frac{1}{l}\sum_{i=1}^{l} V(x_i, y_i, f) + \gamma\|f\|_K^2 \tag{11.1}$$

where V is some loss function, such as squared loss $(y_i - f(x_i))^2$ for RLS or the soft margin loss function $\max[0, 1 - y_i f(x_i)]$ for SVM. Penalizing the RKHS norm imposes smoothness conditions on possible solutions. The classical Representer Theorem states that the solution to this minimization problem exists in $\mathcal{H}_K$ and can be written as

Representer Theorem

$$f^*(x) = \sum_{i=1}^{l} \alpha_i K(x_i, x) \tag{11.2}$$

Therefore, the problem is reduced to optimizing over the finite dimensional space of coefficients α_i, which is the algorithmic basis for SVM, Regularized Least Squares and other regression and classification schemes.

We first consider the case when the marginal distribution is already known.

11.2.1 Marginal Distribution $\mathcal{P}_\mathcal{X}$ is known

Our goal is to extend the kernel framework by incorporating additional information about the geometric structure of the marginal $\mathcal{P}_\mathcal{X}$. We would like to ensure that the solution is smooth with respect to both the ambient space and the marginal distribution $\mathcal{P}_\mathcal{X}$. To achieve that, we introduce an additional regularizer :

Manifold Regularization given the marginal distribution

$$f^* = \underset{f \in \mathcal{H}_K}{\operatorname{argmin}} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \gamma_I \|f\|_I^2 \tag{11.3}$$

where $\|f\|_I^2$ is an appropriate penalty term that should reflect the intrinsic structure of $\mathcal{P}_\mathcal{X}$, e.g., $\langle f, \mathcal{L}_{\mathcal{P}_\mathcal{X}}(f) \rangle_{L^2(\mathcal{P}_\mathcal{X})}$.

Here γ_A controls the complexity of the function in the *ambient* space while γ_I controls the complexity of the function in the *intrinsic* geometry of $\mathcal{P}_\mathcal{X}$. One can derive an explicit functional form for the solution f^* as shown in the following theorem under some fairly general conditions [3]:

Theorem 11.1 *Assume that the intrinsic regularization term is given by*

$$\|f\|_I^2 = \int_\mathcal{X} f D f d\mathcal{P}_\mathcal{X}$$

where D is a bounded operator from the RKHS associated to K to $L^2(\mathcal{P}_\mathcal{X})$. Then the solution f^* to the optimization problem in Eqn. 11.3 above exists and admits the following representation

Representer Theorem given the marginal

$$f^*(x) = \sum_{i=1}^{l} \alpha_i K(x_i, x) + \int_\mathcal{X} \alpha(y) K(x, y)\, d\mathcal{P}_\mathcal{X}(y) \tag{11.4}$$

We note that the Laplace operator as well as any differentiable operator will satisfy the boundedness condition, assuming that the kernel is sufficiently differentiable.

The Representer Theorem above allows us to express the solution f^* directly in terms of the labeled data, the (ambient) kernel K, and the marginal $\mathcal{P}_{\mathcal{X}}$. If $\mathcal{P}_{\mathcal{X}}$ is unknown, we see that the solution may be expressed in terms of an empirical estimate of $\mathcal{P}_X$. Depending on the nature of this estimate, different approximations to the solution may be developed. In the next section, we consider a particular approximation scheme that leads to a simple algorithmic framework for learning from labeled and unlabeled data.

11.2.2 Marginal Distribution $\mathcal{P}_{\mathcal{X}}$ Unknown

In most applications of interest in machine learning the marginal $\mathcal{P}_{\mathcal{X}}$ is not known. Therefore we must attempt to get empirical estimates of $\mathcal{P}_{\mathcal{X}}$ and $\| \ \|_I$. Note that in order to get such empirical estimates it is sufficient to have *unlabeled* examples.

As discussed before the natural penalty on a Riemannian manifold is the Laplace operator. The optimization problem then becomes

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \gamma_I \int_{\mathcal{M}} \langle \nabla_{\mathcal{M}} f, \nabla_{\mathcal{M}} f \rangle$$

It can be shown that the Laplace-Beltrami operator on a manifold can be approximated by graph Laplacian using the appropriate adjacency matrix (see [2, 16] for more details).

Manifold Regularization given unlabeled data

Thus, given a set of l labeled examples $\{(x_i, y_i)\}_{i=1}^{l}$ and a set of u unlabeled examples $\{x_j\}_{j=l+1}^{j=l+u}$, we consider the following optimization problem :

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2} \sum_{i,j=1}^{l+u} (f(x_i) - f(x_j))^2 W_{ij}$$

$$= \operatorname*{argmin}_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2} \mathbf{f}^T L \mathbf{f} \qquad (11.5)$$

Graph Laplacian

where W_{ij} are edge weights in the data adjacency graph, $\mathbf{f} = [f(x_1), \ldots, f(x_{l+u})]^T$, and L is the graph Laplacian given by $L = D - W$. Here, the diagonal matrix D is given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$. The normalizing coefficient $\frac{1}{(u+l)^2}$ is the natural scale factor for the empirical estimate of the Laplace operator (on a sparse adjacency graph, one may normalize by $\sum_{i,j=1}^{l+u} W_{ij}$ instead). The following version of the Representer Theorem shows that the minimizer has an expansion in terms of both labeled and unlabeled examples and is a key to our algorithms.

Representer Theorem given unlabeled data

Theorem 11.2 *The minimizer of optimization problem 11.5 admits an expansion*

$$f^*(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x) \qquad (11.6)$$

in terms of the labeled and unlabeled examples.

The proof is a variation of a standard orthogonality argument [18].

Remark 1: Several natural choices of $\| \, \|_I$ exist. Some examples are:

1. Iterated Laplacians $\mathcal{L}^k$. Differential operators $\mathcal{L}^k$ and their linear combinations provide a natural family of smoothness penalties.

2. Heat semigroup $e^{-\mathcal{L}t}$ is a family of smoothing operators corresponding to the process of diffusion (Brownian motion) on the manifold. For corresponding operators on graphs, see [14]. One can take $\|f\|_I^2 = \int_{\mathcal{M}} f \, e^{\mathcal{L}t}(f)$. We note that for small values of t the corresponding Green's function (the heat kernel of $\mathcal{M}$) can be approximated by a sharp Gaussian in the ambient space.

3. Squared norm of the Hessian (cf. [11]). While the Hessian $\mathbf{H}(f)$ (the matrix of second derivatives of f) generally depends on the coordinate system, it can be shown that the Frobenius norm (the sum of squared eigenvalues) of $\mathbf{H}$ is the same in any geodesic coordinate system and hence is invariantly defined for a Riemannian manifold $\mathcal{M}$. Using the Frobenius norm of $\mathbf{H}$ as a regularizer presents an intriguing generalization of thin-plate splines. We also note that $\mathcal{L}(f) = \mathrm{tr}(\mathbf{H}(f))$.

Remark 2: Note that K restricted to $\mathcal{M}$ (denoted by $K_{\mathcal{M}}$) is also a kernel defined on $\mathcal{M}$ with an associated RKHS $\mathcal{H}_{\mathcal{M}}$ of functions $\mathcal{M} \to \mathbb{R}$. While this might suggest $\|f\|_I = \|f_{\mathcal{M}}\|_{K_{\mathcal{M}}}$ ($f_{\mathcal{M}}$ is f restricted to $\mathcal{M}$) as a reasonable choice for $\|f\|_I$, it turns out, that for the minimizer f^* of the corresponding optimization problem we get $\|f^*\|_I = \|f^*\|_K$, yielding the same solution as standard regularization, although with a different γ. This observation follows from the restriction properties of RKHS [3]. Therefore it is impossible to have an out-of-sample extension without two *different* measures of smoothness. On the other hand, a different ambient kernel restricted to $\mathcal{M}$ can potentially serve as the intrinsic regularization term. For example, a sharp Gaussian kernel can be used as an approximation to the heat kernel on $\mathcal{M}$.

The representer theorem allows us to convert the optimization problem in Eqn 11.5 into a finite dimensional problem of estimating the $(l+u)$ coefficients α^* for the expansion above. A family of algorithms can now be developed with different choices of loss functions, ambient kernels, graph regularizers and optimization strategies.

11.3 Algorithms

11.3.1 Semi-supervised Classification

We now present solutions to the optimization problem posed in Eqn (11.5). To fix notation, we assume we have l labeled examples $\{(x_i, y_i)\}_{i=1}^{l}$ and u unlabeled examples $\{x_j\}_{j=l+1}^{j=l+u}$. We use K interchangeably to denote the kernel function or the Gram matrix.

Laplacian Regularized Least Squares (LapRLS)

The Laplacian Regularized Least Squares algorithm solves Eqn (11.5) with the squared loss function: $V(x_i, y_i, f) = [y_i - f(x_i)]^2$. Since the solution is of the form given by (11.6), the objective function can be reduced to a convex differentiable function of the $(l+u)$-dimensional expansion coefficient vector $\alpha = [\alpha_1, \ldots, \alpha_{l+u}]^T$ whose minimizer is given by :

Laplacian RLS

$$\alpha^* = (JK + \gamma_A l I + \frac{\gamma_I l}{(u+l)^2} LK)^{-1} Y \tag{11.7}$$

Here, K is the $(l+u) \times (l+u)$ Gram matrix over labeled and unlabeled points; Y is an $(l+u)$ dimensional label vector given by: $Y = [y_1, \ldots, y_l, 0, \ldots, 0]$ and J is an $(l+u) \times (l+u)$ diagonal matrix given by: $J = diag(1, \ldots, 1, 0, \ldots, 0)$ with the first l diagonal entries as 1 and the rest 0.

Note that when $\gamma_I = 0$, Eqn (11.7) gives zero coefficients over unlabeled data. The coefficients over labeled data are exactly those for standard RLS.

Laplacian Support Vector Machines (LapSVM)

Laplacian SVMs solve the optimization problem in Eqn. 11.5 with the soft margin loss function defined as $V(x_i, y_i, f) = \max[0, 1 - y_i f(x_i)], y_i \in \{-1, +1\}$. Introducing slack variables and using standard Lagrange Multiplier techniques used for deriving SVMs [22], we first arrive at the following quadratic program in l dual variables β :

$$\beta^\star = \max_{\beta \in \mathbb{R}^l} \sum_{i=1}^{l} \beta_i - \frac{1}{2}\beta^T Q \beta \tag{11.8}$$

subject to the contraints : $\sum_{i=1}^{l} y_i \beta_i = 0, \quad 0 \le \beta_i \le \frac{1}{l} \quad , i = 1, ...l$, where

$$Q = YJK(2\gamma_A I + 2\frac{\gamma_I}{(u+l)^2} LK)^{-1} J^T Y \tag{11.9}$$

Here, Y is the diagonal matrix $Y_{ii} = y_i$, K is the Gram matrix over both the labeled and the unlabeled data; L is the data adjacency graph Laplacian; J is an $l \times (l+u)$ matrix given by: $J_{ij} = 1$ if $i = j,\ x_i$ is a labeled example and $J_{ij} = 0$ otherwise. To obtain the optimal expansion coefficient vector $\alpha^* \in \mathbb{R}^{(l+u)}$, one has to solve the following linear system after solving the quadratic program above :

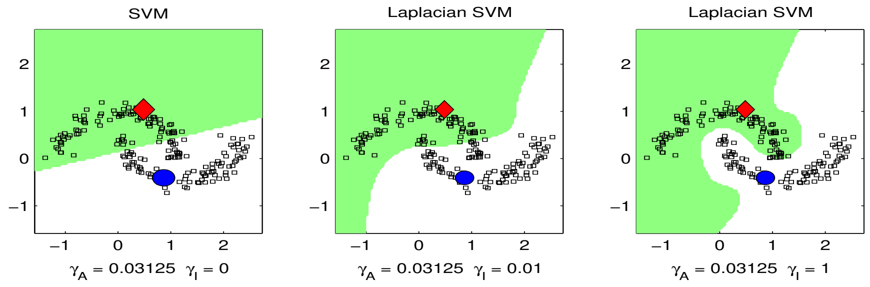
Laplacian SVM

$$\alpha^* = (2\gamma_A I + 2\frac{\gamma_I}{(u+l)^2} LK)^{-1} J^T Y \beta^\star \tag{11.10}$$

One can note that when $\gamma_I = 0$, the SVM QP and Eqns (11.9,11.10), give zero expansion coefficients over the unlabeled data. The expansion coefficients over the labeled data and the Q matrix are as in standard SVM, in this case. Laplacian SVMs can be easily implemented using standard SVM software and packages for solving linear systems.

	Laplacian SVM/RLS
Input:	l labeled examples $\{(x_i, y_i)\}_{i=1}^{l}$, u unlabeled examples $\{x_j\}_{j=l+1}^{l+u}$
Output:	Estimated function $f : \mathbb{R}^n \to \mathbb{R}$
Step 1	Construct data adjacency graph with $(l+u)$ nodes using, e.g, k nearest neighbors. Choose edge weights W_{ij}, e.g., binary weights or heat kernel weights $W_{ij} = e^{-\|x_i - x_j\|^2/4t}$.
Step 2	Choose a kernel function $K(x, y)$. Compute the Gram matrix $K_{ij} = K(x_i, x_j)$.
Step 3	Compute graph Laplacian matrix : $L = D - W$ where D is a diagonal matrix given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$.
Step 4	Choose γ_A and γ_I.
Step 5	Compute α^* using Eqn (11.7) for squared loss (Laplacian RLS) or using Eqns (11.9,11.10) together with the SVM QP solver for soft margin loss (Laplacian SVM).
Step 6	Output function $f^*(x) = \sum_{i=1}^{l+u} \alpha_i^* K(x_i, x)$.
	Equivalently, after step 4 construct the kernel function $\tilde{K}(x, y)$ given by Eqn 11.15, and use it in standard SVM/RLS (or with other suitable kernel methods).

Figure 11.4 Two Moons Dataset: Laplacian SVM with increasing intrinsic regularization.



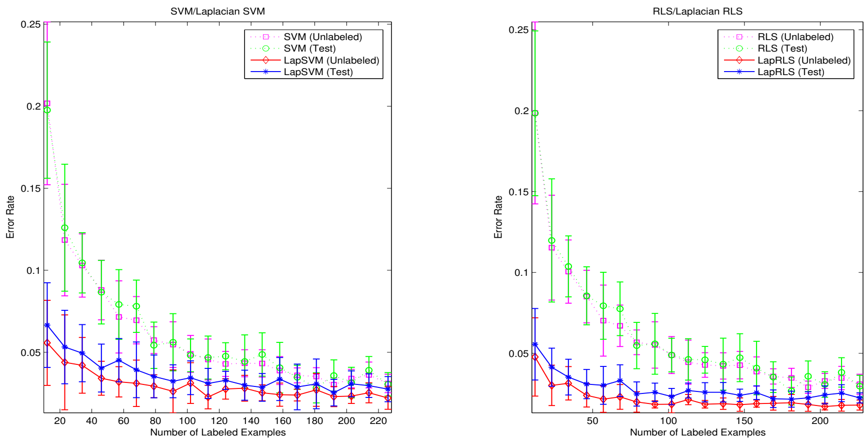
In section 11.4, we will discuss a data-dependent kernel defined using unlabeled examples [20], with which standard supervised SVM/RLS implement Laplacian SVM/RLS. In Table 1, we outline these algorithms.

Effect of increasing γ_I

The choice of the regularization parameters γ_A, γ_I is a subject of future research. If there is enough labeled data, it can be be based on cross-validation or performance on a held-out test set. In Figure 11.4 we provide an intuition towards the role of these parameters on a toy two-moons dataset. When $\gamma_I = 0$, Laplacian SVM recovers standard supervised SVM boundaries. As γ_I is increased, the effect of unlabeled data increases and the classification boundaries are appropriately adjusted.

In Figure 11.3.1 we plot the learning curves for Laplacian SVM/RLS on a two-

Figure 11.5 Image Classification: Laplacian SVM/RLS performance with respect to number of labeled examples on unlabeled and test data.



class image recognition problem. In many such real-world application settings, one may expect significant benefit from utilizing unlabeled data and high-quality out-of-sample extensions with these algorithms. For further empirical results see [3, 20] and elsewhere in this book.

11.3.2 Unsupervised Learning and Data Representation

Regularized Spectral Clustering

The unsupervised case can be viewed as a special case of semi-supervised learning where one is given a collection of unlabeled data points $x_1, \ldots, x_u$ and no labeled examples. Our basic algorithmic framework embodied in the optimization problem in Eqn. 11.3 has three terms: (i) fit to labeled data, (ii) extrinsic regularization and (iii) intrinsic regularization. Since no labeled data is available, the first term does not arise anymore. Therefore we are left with the following optimization problem:

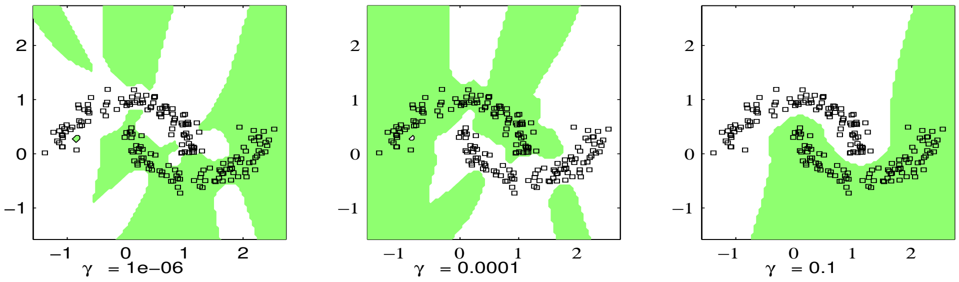
$$\min_{f \in \mathcal{H}_K} \gamma_A \|f\|_K^2 + \gamma_I \|f\|_I^2 \tag{11.11}$$

Of course, only the ratio $\frac{\gamma_A}{\gamma_I}$ matters. As before $\|f\|_I^2$ can be approximated using the unlabeled data. Choosing $\|f\|_I^2 = \int_{\mathcal{M}} \langle \nabla_{\mathcal{M}} f, \nabla_{\mathcal{M}} f \rangle$ and approximating it by the empirical Laplacian, we are left with the following optimization problem :

Clustering

$$f^* = \operatorname*{argmin}_{\substack{\sum_i f(x_i)=0;\ \sum_i f(x_i)^2=1 \\ f \in \mathcal{H}_K}} \gamma \|f\|_K^2 + \sum_{i \sim j} (f(x_i) - f(x_j))^2 \tag{11.12}$$

Figure 11.6 Two Moons Dataset: Regularized Clustering



Note that without the additional constraints (cf. [4]) the above problem gives degenerate solutions.

As in the semi-supervised case, a version of the empirical Representer theorem holds showing that the solution to Eqn. 11.12 admits a representation of the form

$$f^* = \sum_{i=1}^{u} \alpha_i K(x_i, \cdot)$$

By substituting back in Eqn. 11.12, we come up with the following optimization problem:

$$\alpha = \operatorname*{argmin}_{\substack{\mathbf{1}^T K\alpha=0 \\ \alpha^T K^2\alpha=1}} \gamma\|f\|_K^2 + \sum_{i\sim j}(f(x_i) - f(x_j))^2$$

where $\mathbf{1}$ is the vector of all ones and $\alpha = (\alpha_1, \ldots, \alpha_u)$ and K is the corresponding Gram matrix.

Letting P be the projection onto the subspace of $\mathbb{R}^u$ orthogonal to $K\mathbf{1}$, one obtains the solution for the constrained quadratic problem, which is given by the generalized eigenvalue problem

Eigenvalue problem

$$P(\gamma K + KLK)P\mathbf{v} = \lambda P K^2 P\mathbf{v} \tag{11.13}$$

The final solution is given by $\alpha = P\mathbf{v}$, where $\mathbf{v}$ is the eigenvector corresponding to the smallest eigenvalue.

Effect of increasing γ

The method sketched above is a framework for regularized spectral clustering. The regularization parameter γ controls the smoothness of the resulting function in the ambient space. We also obtain a natural out-of-sample extension for clustering points not in the original data set. Figure 11.5 shows this method on a toy two-moons clustering problem. Unlike recent work [6, 7] on out-of-sample extensions, our method is based on a Representer theorem for RKHS.

Regularized Laplacian Eigenmaps

Dimensionality Reduction

One can take multiple eigenvectors of the system in Eqn. 11.13 and represent a point x in $\mathbb{R}^m$ as:

$$x \mapsto \left[\sum_{i=1}^{u} \alpha_i^1 K(x_i, x), \ldots, \sum_{i=1}^{u} \alpha_i^m K(x_i, x) \right]$$

where $(\alpha_1^j \ldots \alpha_u^j)$ is the j^{th} eigenvector.

This leads to new method for dimensionality reduction and data representation that provides a natural out-of-sample extension of Laplacian Eigenmaps [2]. The new representation of the data in $\mathbb{R}^m$ in optimal in the sense that it best preserves its local structure (as estimated by the graph) in the original ambient space.

11.3.3 Fully Supervised Learning

The fully supervised case represents the other end of the spectrum of learning. Since standard supervised algorithms (SVM and RLS) are special cases of manifold regularization, our framework is also able to deal with a labeled dataset containing no unlabeled examples. Additionally, manifold regularization can augment supervised learning with intrinsic regularization, possibly in a class-dependent manner, which suggests the following learning problem:

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) \; + \; \gamma_A \|f\|_K^2 \; + \; \gamma_I^+ \mathbf{f}_+^T L_+ \mathbf{f}_+ \; + \; \gamma_I^- \mathbf{f}_-^T L_- \mathbf{f}_- \quad (11.14)$$

Supervised Manifold Regularization

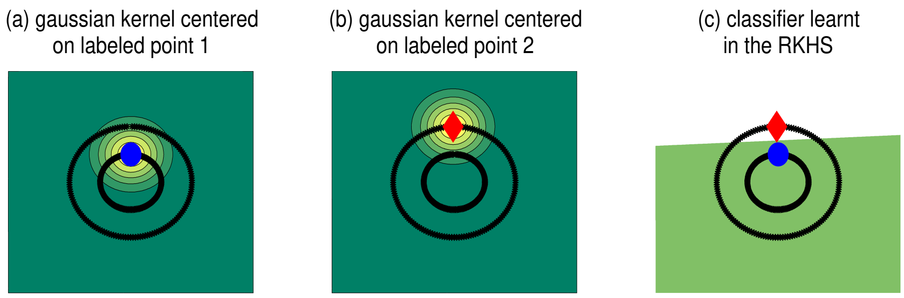
Here we introduce two intrinsic regularization parameters γ_I^+, γ_I^- and regularize separately for the two classes : $\mathbf{f}_+$, $\mathbf{f}_-$ are the vectors of evaluations of the function f, and L_+, L_- are the graph Laplacians, on positive and negative examples respectively. The solution to the above problem for RLS and SVM can be obtained by replacing $\gamma_I L$ by the block-diagonal matrix $\begin{pmatrix} \gamma_I^+ L_+ & 0 \\ 0 & \gamma_I^- L_- \end{pmatrix}$ in the Laplacian SVM and Laplacian RLS algorithms.

11.4 Data-dependent Kernels for Semi-supervised Learning

Warping an RKHS

By including an intrinsic regularization term $\|f\|_I$ in addition to the prior measure of complexity $\|f\|_K$ of a function f in the RKHS $\mathcal{H}_K$, the algorithmic framework presented above reflects how unlabeled data may alter our complexity beliefs. This data-dependent modification of the norm can be viewed as an attempt to appropriately warp an RKHS to conform to the geometry of the marginal distribution (for a discussion, see [20]). This is made precise in the following discussion. The set of functions in $\mathcal{H}_K$ has an associated inner product $\langle f, g \rangle_{\mathcal{H}_K}$ for $f, g \in \mathcal{H}_K$. Given

Figure 11.7 Learning in an RKHS



unlabeled data, the space of functions $\tilde{\mathcal{H}}_{\tilde{K}}$ containing functions in $\mathcal{H}_K$ but with the following modified inner product:

$$\langle f, g\rangle_{\tilde{\mathcal{H}}_{\tilde{K}}} = \langle f, g\rangle_{\mathcal{H}} + \frac{\gamma_I}{\gamma_A}\mathbf{f}^T L\mathbf{g}$$

can be shown to be an RKHS with an associated kernel $\tilde{K}$. The regularization term $\gamma_A\|f\|_{\tilde{H}_{\tilde{K}}}$ in this RKHS provides the same complexity penalty as the joint intrinsic and ambient regularization terms in $\mathcal{H}_K$. Thus, once the kernel $\tilde{K}$ is available, one can the employ standard machinery of kernel methods designed for supervised learning, for semi-supervised inference. The form of the new kernel $\tilde{K}$ can be derived in terms of the kernel function K using reproducing properties of an RKHS and orthogonality arguments (see [19, 20] for a derivation) and is given by :

Kernels for Semi-supervised Learning

$$\tilde{K}(x, z) = K(x, z) - \mathbf{k}_x^T(I + \frac{\gamma_I}{\gamma_A}LK)^{-1}L\mathbf{k}_z \tag{11.15}$$

where $\mathbf{k}_x$ (and similarly $\mathbf{k}_z$) denotes the vector $[K(x_1, x), \ldots, K(x_{l+u}, x)]^T$. The standard representer theorem can be now be invoked to show that the minimizer of optimization problem 11.5 admits the following expansion in terms of labeled examples only:

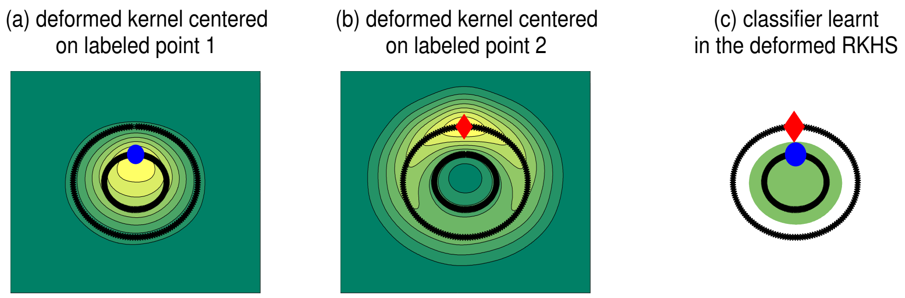
$$f^*(x) = \sum_{i=1}^{l} \alpha_i \tilde{K}(x_i, x) \tag{11.16}$$

Other Algorithms

With the new kernel $\tilde{K}$, this representer theorem reduces the minimization problem 11.5 to that of estimating the l expansion coefficients α^*. . In addition to recovering the algorithms in Section 11.3, this kernel can also be used to implement, e.g., semi-supervised extensions of support vector regression, one-class SVM and Gaussian processes (see [21]).

To develop an intuition towards how the intrinsic norm warps the structure of an RKHS, consider the pictures shown in Figure 11.4. A practitioner of kernel methods

Figure 11.8 Warping an RKHS



would approach the two-circles problem posed in Figure 11.1 by choosing a kernel function $K(x, y)$; and then taking a particular linear combination of this kernel centered at the two labeled points in order to construct a classifier. Figures 11.6 (a,b) show this attempt with the popular Gaussian kernel. The resulting linear decision surface, shown in Figure 11.6 (c), is clearly inadequate for this problem.

Warping Interpretation in pictures

In Figures 11.7(a) and 11.7(b) we see level sets for the deformed kernel $\tilde{K}$ centered on the two labeled points in the two-circles problem. The kernel deforms along the circle under the influence of the unlabeled data. Using this kernel, instead of $K(x, y)$, produces a satisfactory class boundary with just two labeled points, as shown in Figure 11.7 (c).

The procedure described above is a general non-parametric approach for constructing data-dependent kernels for semi-supervised learning. This approach differs from prior constructions that have largely focussed on data-dependent methods for parameter selection to choose a kernel from some parametric family, or by defining a kernel matrix on the data points alone (transductive setting).

11.5 Linear methods for Large Scale Semi-supervised Learning

To turn semi-supervised learning into a technology, one needs to address issues of scalability of algorithms and applicability to large datasets. The algorithms we have described deal with dense matrices of size $n \times n$ and have $O(n^3)$ training complexity with naive implementations. The expansion over labeled or unlabeled examples is in general not sparse, even for Laplacian SVMs. One can possibly employ, for example, various reduced set methods, low-rank kernel approximations or sparse greedy methods (see [18] for a discussion of general implementation issues in kernel methods) for efficient implementation of these algorithms.

Linear Manifold Regularization

Due to their potential for dealing with massive datasets and wide-spread applicability, linear semi-supervised methods generate special interest. The algorithms described above can easily be specialized for constructing linear classifiers by choosing the linear kernel $K(x, y) = x^T y$. However, if the data-dimensionality d is much smaller than the number of examples or the data is highly sparse, one can much more efficiently solve the primal problem directly, once the graph regularizer is constructed. We can learn a weight vector $w \in \mathbb{R}^d$ defining the linear classifier $f(x) = \text{sign}(w^T x)$ as follows:

$$w^\star = \underset{w \in \mathbb{R}^d}{\text{argmin}} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, w^T x_i) + \gamma_A \|w\|^2 + \frac{\gamma_I}{(u+l)^2} w^T X^T L X w \quad (11.17)$$

Here, X is the $(l+u) \times d$ data matrix.

Linear Laplacian RLS

For Linear Laplacian RLS, taking V to be the squared loss and setting the gradient of the objective function to 0, we immediately obtain a linear system that can be solved to obtain the desired weight vector:

$$(X_l^T X_l + \gamma_A l I + \frac{\gamma_I l}{(l+u)^2} X^T L X) w = X_l^T Y \quad (11.18)$$

Here X_l is the sub-matrix of X corresponding to labeled examples and Y is the vector of labels. This is a $d \times\ d$ system which can be easily solved when d is small. When d is large but feature vectors are highly sparse, we can employ Conjugate Gradient (CG) methods to solve this system. CG techniques are Krylov methods involve repeated multiplication of a candidate solution z by A for solving a linear system $Ax = b$. The matrix A need not be explicitly constructed so long as the matrix vector product Az can be computed. In the case of linear Laplacian RLS, we can construct the matrix-vector product fast due to the sparsity of X and L[1].

For Linear Laplacian SVMs, we can rewrite problem 11.17 as:

$$w^* = \underset{w\ in\mathbb{R}^d}{\text{argmin}} \gamma_A w^T T T^T w + \frac{1}{l} \sum_{i=1}^{l} \max\left[0, 1 - y_i(w^T x_i)\right]$$

Linear Laplacian SVM

in terms of the Cholesky factorization TT^T of the positive definite matrix $(\gamma_A I + \frac{\gamma_I}{(l+u)^2} X^T L X)$. Changing variables by $\tilde{w} = T^T w$ and $\tilde{x} = T^{-1} x$, we can convert the above problem into a standard SVM running only on the labeled examples that are pre-processed with T^{-1}. When d is small, the pre-processing matrix can be computed cheaply. The re-parameterized SVM then runs only on a small number of labeled examples and returns a weight vector $\tilde{w}^*$. We obtain the solution of the original problem by setting $w^* = \left(T^T\right)^{-1} \tilde{w}^*$. We note in passing that the inner product in the pre-processed space is given by $\tilde{x}^T \tilde{z} = x^T (TT^T)^{-1} z$. An application of the Woodbury formula to compute the inverse $(TT^T)^{-1}$ followed by appropriate manipulations gives a simple "feature-space" derivation of the data-dependent kernel in Section 11.4. For high-dimensional sparse datasets, we can use the large scale training algorithm in [12] for L_2-SVM. At the core of this algorithm are RLS iterations implemented using conjugate gradient techniques. In conjunction

1. Fast matrix-vector products can also be formed for dense graph regularizers given by a power series in the (sparse) graph Laplacian

with Linear Laplacian RLS for large sparse datasets, this algorithm can also be extended for large scale semi-supervised learning.

11.6 Connections to Other Algorithms and Related Work

The broad connections of our approach to graph-based learning techniques and kernel methods are summarized in Table 1 through a comparison of objectives. When $\gamma_I = 0$, our algorithms ignore unlabeled data and perform standard regularization, e.g in SVMs and RLS. By optimizing over an RKHS of functions defined everywhere in the ambient space, we get out-of-sample extension for graph regularization, when $\gamma_A \to 0, \gamma_I > 0$. In the absence of labeled examples, we perform a regularized version of spectral clustering that is often viewed as a relaxation of the discrete graph min-cut problem. We can also obtain useful data representations within the same framework by regularized Laplacian eigenmaps.

Table 11.1 Objective Functions for comparison (In the third column for unsupervised algorithms, additional constraints are added to avoid trivial or unbalanced solutions). In addition to these learning problems, the framework also provides the regularized Laplacian eigenmaps algorithm for dimensionality reduction and data representation.

Supervised	Partially Supervised	Clustering
Kernel-based Classifiers	*Graph Regularization*	*Graph Mincut*
$\operatorname{argmin}_{f\in\mathcal{H}_K}$	$\operatorname{argmin}_{\mathbf{f}\in\mathbb{R}^{(l+u)}}$	$\operatorname{argmin}_{\mathbf{f}\in\{-1,+1\}^u}$
$\frac{1}{l}\sum_{i=1}^{l} V(y_i, f(x_i)) + \gamma\|f\|_K^2$	$\frac{1}{l}\sum_{i=1}^{l} V(y_i, \mathbf{f}_\mathbf{i}) + \gamma\mathbf{f}^T L\mathbf{f}$	$\frac{1}{4}\sum_{i,j=1}^{u} W_{ij}(\mathbf{f}_i - \mathbf{f}_j)^2$
	Out-of-sample Extn.	*Spectral Clustering*
	$\operatorname{argmin}_{f\in\mathcal{H}_K}$	$\operatorname{argmin}_{\mathbf{f}\in\mathbb{R}^u} \frac{1}{2}\mathbf{f}^T L\mathbf{f}$
	$\frac{1}{l}\sum_{i=1}^{l} V(y_i, \mathbf{f}_\mathbf{i}) + \gamma\mathbf{f}^T L\mathbf{f}$	*Out-of-sample Extn.*
	Manifold Regularization	$\operatorname{argmin}_{f\in\mathcal{H}_K} \frac{1}{2}\mathbf{f}^T L\mathbf{f}$
	$\operatorname{argmin}_{f\in\mathcal{H}_K}$	*Reg. Spectral Clust.*
	$\frac{1}{l}\sum_{i=1}^{l} V(y_i, f(x_i)) +$	$\operatorname{argmin}_{f\in\mathcal{H}_K}$
	$\gamma_A\|f\|_K^2 + \frac{\gamma_I}{(l+u)^2}\mathbf{f}^T L\mathbf{f}$	$\frac{1}{2}\mathbf{f}^T L\mathbf{f} + \gamma\|f\|_K^2$

The conceptual framework of our work is close, in spirit, to the *measure-based regularization* approach of [8]. The authors consider a gradient based regularizer that encourages smoothness with respect to the data density. While [8] use the gradient $\nabla f(x)$ in the ambient space, we use the gradient over a submanifold $\nabla_{\mathcal{M}} f$. In a situation where the data truly lies on or near a submanifold $\mathcal{M}$, the difference between these two penalizers can be significant since smoothness in the normal direction to the data manifold is irrelevant to classification or regression.

The intuition of incorporating a graph-based regularizer in the design of semi-supervised variants of inductive algorithms has been also been explored in [24, 15, 13]. In [24], a least squares algorithm is proposed that provides an out-of-sample extension for graph transduction in the span of a fixed set of basis functions $\{\phi_i : \mathcal{X} \mapsto \mathbb{R}\}_{i=1}^{s}$. Thus, the optimization problem in 11.5 is solved over this span for the squared loss leading to a linear system such as Eqn. 11.18 (set $X_{ij} = \phi_j(x_i)$ and $\gamma_A = 0$) whose size is given by the number of basis functions s. For a small set of basis functions, this system can be solved more efficiently. [24] also discusses data representation within this framework.

In [15], the authors impose a prior derived from the graph Laplacian, over parameters of a multinomial logistic regression model. For an r-class problem, the class probabilities are modeled as:

$$P(y^{(j)} = 1|x) = \frac{e^{w^{(i)T}x}}{\sum_{i=1}^{r} e^{w^{(i)T}x}} \qquad 1 \leq j \leq r$$

where $y^{(j)}$ is an indicator variable for class j and $w^{(i)} \in \mathbb{R}^d$ is the weight vector for class i. The prior on weight vector $w^{(i)}$ is given by:

$$P(w^{(i)}) \propto \exp\left\{\frac{-w^{(i)T}\left(\gamma_I^{(i)} X^T L X + D^{(i)}\right) w^{(i)T}}{2}\right\}$$

where $D^{(i)}$ is a parameterized diagonal matrix providing extra regularization similar to the ambient penalty term in Manifold regularization. Bayesian inference is performed to learn the maximum *a posteriori* (MAP) estimate of the model parameters with an Expectation-Maximization algorithm.

In [13], an extension of the Adaboost algorithm is proposed (also discussed elsewhere in this book) that implements similar intuitions within the framework of Boosting techniques. In [1], a generalization of the problem in Eqn. 11.5 is presented for semi-supervised learning of structured variables.

By introducing approximations to avoid graph re-computation, methods for out-of-sample extension have also been suggested without explicitly operating in an ambiently defined function or model space. In [10] an induction formula is derived by assuming that the addition of a test point to the graph does not change the transductive solution over the unlabeled data. In other words, if $\mathbf{f} = [f_1 \ldots f_{l+u}\ f_t]$ denotes a function defined on the augmented graph, with f_t as its value on the node corresponding to the test point, then minimizing the objective function for graph regularization (with L as the regularizer) keeping the values on the original nodes fixed, one can obtain a Parzen windows expression for f_t:

$$f_t = \frac{\sum_i W_{ti} f_i}{\sum_i W_{ti}}$$

where W denotes the adjacency matrix as before. In [25], a test point is classified according to its nearest neighbor on the graph, whose classification is available after transductive inference. In [9], graph kernels are constructed by modifying the spectrum of the gram matrix of a kernel evaluated over labeled and unlabeled examples. Unseen test points are approximated in the span of the labeled and unlabeled data, and this approximation is used to extend the graph kernel.

The regularized Laplacian eigenmap algorithms presented in 11.3.2 has also been simultaneously and independently developed by [23] in the context of extending a partially known graph. The graph inference problem is posed as follows: Suppose a graph $G = (V, E)$ with vertices V and edges E is observed and is known to be a subgraph of an unknown graph $G' = (V', E')$ with $V \subset V'$ and $E \subset E'$. Given the vertices $V' - V$, infer the edges $E' - E$. If the vertices v are elements of some set $\mathcal{V}$ on which a kernel function $K : \mathcal{V} \times \mathcal{V}$ is defined, then one can infer the graph in two steps: Find a map $\psi : \mathcal{V} \mapsto \mathbb{R}^m$ and induce a nearest neighbor graph on the embedded points. To find the map ψ in the RKHS corresponding to K, one can setup an optimization problem (similar to that in regularized classification), involving a graph Laplacian based "data fit" term that measures how well ψ preserves the local structure of the observed graph and the RKHS regularizer that provides ambient smoothness. This is also the objective function of regularized Laplacian Eigenmaps, and involes solving the generalized eigenvalue problem 11.13 for multiple eigenvectors.

11.7 Future Directions

We have discussed a general framework for incorporating geometric structures in the design of learning algorithms. Our framework may be extended to include additional domain structure e.g in the form of invariances and structured outputs. Many directions are being pursued towards improving the scalability and efficiency of our algorithms, while developing extensions to handle unlabeled data in, e.g., support vector regression, one class SVMs and Gaussian processes. We plan to pursue applications of these methods to a variety of real-world learning tasks, and investigate issues concerning generalization analysis and model selection.

References

1. Y. Altun, D. McAllester & M. Belkin, *Maximum Margin Semi-Supervised Learning for Structured Variables*, NIPS 2005
2. M. Belkin, *Problems of Learning on Manifolds* Ph.D. Dissertation, University of Chicago, Dept. of Mathematics, 2003.
3. M. Belkin, P. Niyogi & V. Sindhwani, *Manifold Regularization : A Geometric Framework for Learning for Examples* Technical Report, Univ. of Chicago, Department of Computer Science, TR-2004-06, 2004
4. M. Belkin, I. Matveeva, & P. Niyogi, *Regression and Regularization on Large Graphs* COLT 2004.
5. M. Belkin, P. Niyogi & V. Sindhwani, *On Manifold Regularization* Artificial Intelligence and Statistics, Barbados, 2005
6. Y. Bengio, J-F. Paiement, & P. Vincent,*Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering*, NIPS 2003.
7. M. Brand, *Nonlinear dimensionality reduction by kernel eigenmaps.* Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, pp. 547-552, Acapulco, Mexico, 9-15 August 2003.
8. O. Bousquet, O. Chapelle, M. Hein, (2003) *Measure Based Regularization* NIPS 2003
9. O. Chapelle, J. Weston & B. Schoelkopf, *Cluster Kernels for Semi-Supervised Learning* NIPS 2002.
10. O. Delalleau, Y. Bengio & N. Le Roux, *Efficient Non-Parametric Function Induction in Semi-Supervised Learning* Artificial Intelligence and Statistics, 2005
11. D. L. Donoho & C. E. Grimes, *Hessian Eigenmaps: new locally linear embedding tech- niques for high-dimensional data*, Proceedings of the National Academy of Arts and Sciences vol. 100 pp. 5591-5596, 2003.
12. S. S. Keerthi, D. DeCoste *A Modified Finite Newton Method for Fast Solution of Large Scale Linear SVMs* Journal of Machine Learning Research, 6(Mar):341–361, 2005.
13. B. Kegl and L. Wang, *Boosting on manifolds: adaptive regularization of base classifiers* NIPS 04: Neural Information Processing Systems, 2004
14. R. I. Kondor & J. Lafferty *Diffusion Kernels on Graphs and Other Discrete*

Input Spaces ICML 2002

15. B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, M. Figueiredo, *On Semi-Supervised Classification* Neural Information Processing Systems 17 (NIPS04), 2004
16. S. Lafon, *Diffusion maps and geometric harmonics* Ph.D. dissertation, Yale University, 2004
17. T. Poggio & F. Girosi, *Regularization Algorithms for Learning That Are Equivalent to Multilayer Networks* Science 247:978-982, 1990
18. B. Schoelkopf, & A.J. Smola, *Learning with Kernels* MIT Press, Cambridge, MA 2002
19. V. Sindhwani, *Kernel Machines for Semi-supervised Learning*, Masters Thesis, University of Chicago, November 2004.
20. V. Sindhwani, P. Niyogi, & M. Belkin, *Beyond the Point Cloud: from Transductive to Semi-supervised Learning*, ICML 2005.
21. V. Sindhwani, W. Chu, & S. S. Keerthi, *Semi-supervised Gaussian Processes*, Yahoo! Research Technical Report (in preparation), 2005
22. V. Vapnik, (1998) *Statistical Learning Theory* Wiley-Interscience, 1998.
23. J.-P. Vert & Y. Yamanishi (2004), *Supervised graph inference* Advances in Neural Information Processing Systems 17 (NIPS 2004), 2004
24. K. Yu, V. Tresp, & D. Zhou (2004) *Semi-supervised Induction with Basis Fnctions* Technical Report, Max-Planck Institute, No. 141
25. X. Zhu, J. Lafferty, Z. Ghahramani (2003) *Semi-Supervised Learning: From Gaussian Fields to Gaussian Processes* Tech report CMU-CS-03-175, 2003