Beyond the Point Cloud: from Transductive to Semi-supervised Learning

Vikas Sindhwani VIKASS@CS.UCHICAGO.EDU
Partha Niyogi NIYOGI@CS.UCHICAGO.EDU
Mikhail Belkin MISHA@CS.UCHICAGO.EDU
Department of Computer Science, University of Chicago, Chicago, IL 60637

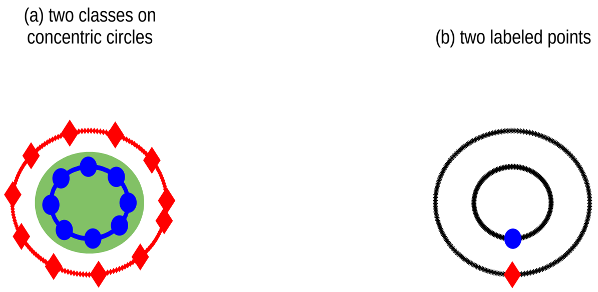
Abstract

Due to its occurrence in engineering domains and implications for natural learning, the problem of utilizing unlabeled data is attracting increasing attention in machine learning. A large body of recent literature has focussed on the *transductive* setting where labels of unlabeled examples are estimated by learning a function defined only over the point cloud data. In a truly *semi-supervised* setting however, a learning machine has access to labeled and unlabeled examples and must make predictions on data points never encountered before. In this paper, we show how to turn transductive and standard supervised learning algorithms into semi-supervised learners. We construct a family of data-dependent norms on Reproducing Kernel Hilbert Spaces (RKHS). These norms allow us to warp the structure of the RKHS to reflect the underlying geometry of the data. We derive explicit formulas for the corresponding new kernels. Our approach demonstrates state of the art performance on a variety of classification tasks.

1. Introduction

To set the stage for the developments that will follow, consider the picture shown in Fig. 1(a). Shown in that figure are two classes of data points in the plane ($\mathbb{R}^2$) such that all data points lie on one of two concentric circles. This represents a two class pattern classification problem where each class is identified with one of the circles. The decision boundary separating the two classes is non-linear. A typical kernel based approach for pattern classification would be to use the Gaussian (RBF) kernel $k(x, z) = e^{-\frac{\|x-z\|^2}{2\sigma^2}}$. The kernel k naturally defines a unique Reproducing Kernel Hilbert Space (RKHS) of functions, which we will denote by $\mathcal{H}$, on the two-dimensional plane.

Figure 1. A binary classification problem : Classes (diamonds and circles) lie on two concentric circles.



Suppose we are given a small number, l, of labeled example pairs (x_i, y_i) where each $x_i \in \mathbb{R}^2$ and $y_i \in \{-1, +1\}$. Then, in order to "learn" a good classifier from the labeled examples, one may solve the following regularization problem:

$$f = \arg\min_{h \in \mathcal{H}} \frac{1}{l} \sum_{i=1}^{l} V(h, x_i, y_i) + \gamma \|h\|_{\mathcal{H}}^2$$

where $\|h\|_{\mathcal{H}}$ is the norm of the function h in the RKHS and V is a loss function. By the familiar representer theorem, the solution can be expressed as:

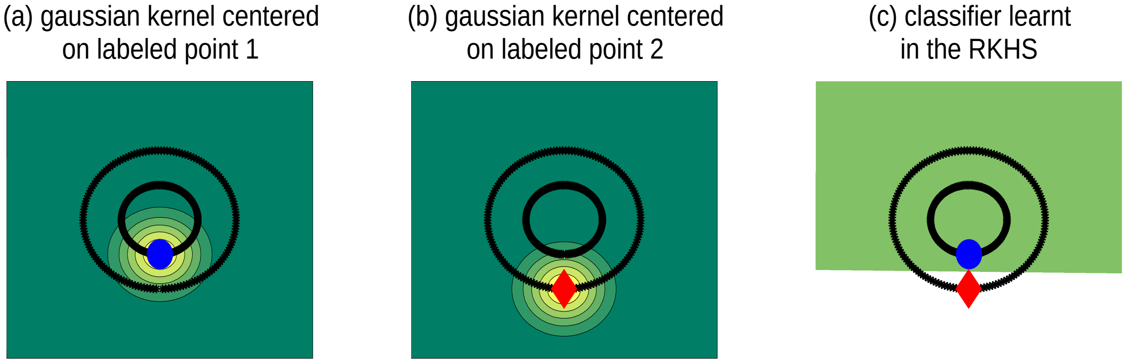
$$f(x) = \sum_{i=1}^{l} \alpha_i k(x, x_i)$$

Choosing V to be the square loss gives rise to Regularized Least Squares (RLS) while choosing it to be the hinge loss produces Support Vector Machines (SVM).

For illustrative purposes, we consider in Fig. 1(b) the case where $l = 2$, i.e., two labeled examples (one positive and one negative) are provided to the learner.

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).

Figure 2.



Then the learned function would be a linear combination of two Gaussians, each centered on one of the two data points. The contours (level sets) of the Gaussian centered at each datapoint are shown in the Fig. 2(a),(b). Because the Gaussian kernel is isotropic, it has a spherical symmetry. As a result, the decision surface is linear, as shown in Fig. 2(c).

It is clear in our setting that the Gaussian with its spherical symmetry is an unfortunate choice for the kernel as it does not conform to the particular geometry of the underlying classes, and is unable to provide a satisfactory decision surface. The question we set for ourselves in this paper is the following:

Can we define a kernel $\tilde{k}$ that is adapted to the geometry of the data distribution?

Such a kernel $\tilde{k}$ must have the property that (i) it is a valid Mercer kernel $\tilde{k} : X \times X \rightarrow \mathbb{R}$ and therefore defines a new RKHS $\tilde{\mathcal{H}}$. (ii) it implements our intuitions about the geometry of the data. Our hope is to obtain an optimization problem over this new RKHS $\tilde{\mathcal{H}}$, given by:

$$g = \arg\min_{h \in \tilde{\mathcal{H}}} \frac{1}{2} \sum_{i=1}^{2} V(h, x_i, y_i) + \|h\|_{\tilde{\mathcal{H}}}^2$$

whose solution $g(x) = \sum_{i=1}^{2} \alpha_i \tilde{k}(x, x_i)$ should be appropriate for our setting.

Notice that g is still a linear combination of two (modified) kernel functions, centered at the two data points in question. Yet, this solution must produce an intuitive decision surface that separates the two circles such as in Fig. 1(a). The form of such a Mercer kernel is not a-priori obvious for our picture.

In this paper, we will show how to deform the original space to obtain a new RKHS $\tilde{\mathcal{H}}$ to satisfy our objectives. Following the philosophy of Manifold Regularization in (Belkin, Niyogi & Sindhwani, 2004; Sindhwani, 2004), the geometry of the underlying marginal distribution may be estimated from unlabeled data and incorporated into the deformation procedure. The resulting new kernel $\tilde{k}$ can be computed explicitly in terms of unlabeled data. Working with only labeled data in this new RKHS, we can use the full power of *supervised* kernel methods for *semi-supervised* inference.

We highlight the following aspects of this paper:

1. As far as we know, we obtain the first truly data-dependent non-parametric kernel for semi-supervised learning. Prior work on data dependent kernels may be roughly classified into two categories: (a) choosing parameters for some parametric family of kernels, and (b) defining a data dependent kernel on the data points alone (transductive setting). See section 3 for a discussion of prior work.

2. We discuss the basic theoretical properties of this kernel and establish that it is a valid Mercer kernel and therefore defines an RKHS.

3. These developments allow a family of algorithms to be developed based on various choices of the original RKHS, deformation penalties, loss functions and optimization strategies.

4. We provide experimental comparisons showing state-of-the-art performance on a variety of classification tasks. In particular, we see that this approach can be used successfully in both transductive and semi-supervised settings.

We now continue the discussion above and describe a general scheme for appropriately warping an RKHS.

2. Warping an RKHS using Point Cloud Norms

Before proceeding we discuss the basic properties of RKHS. Let X be a compact domain in a Euclidean space or a manifold. A complete Hilbert space $\mathcal{H}$ of functions $X \to \mathbb{R}$, with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is a Reproducing Kernel Hilbert Space if point evaluation functionals are bounded, i.e., for any $x \in X, f \in \mathcal{H}$, there is a C, s.t.

$$|f(x)| \ \leq C\|f\|_{\mathcal{H}}$$

A symmetric positive semidefinite kernel $k(x,z)$ can then be constructed using the Riesz representation theorem for the point evaluation functional:

$$f(x) = \langle f, k(x,\cdot)\rangle_{\mathcal{H}} \qquad k(x,z) = \langle k(x,\cdot), k(z,\cdot)\rangle_{\mathcal{H}}$$

We will now show how a very general procedure to deform the norm $\| \ \|_{\mathcal{H}}$ gives a new RKHS $\tilde{\mathcal{H}}$ whose kernel we will denote by $\tilde{k}(x,z)$.

Let $\mathcal{V}$ be a linear space with a positive semi-definite inner product (quadratic form) and let $S : \mathcal{H} \to \mathcal{V}$ be a bounded linear operator. We define $\tilde{\mathcal{H}}$ to be the space of functions from $\mathcal{H}$ with the modified inner product

$$\langle f, g\rangle_{\tilde{\mathcal{H}}} = \langle f, g\rangle_{\mathcal{H}} + \langle Sf, Sg\rangle_{\mathcal{V}}$$

Proposition 2.1 $\tilde{\mathcal{H}}$ *is a Reproducing Kernel Hilbert Space.*

Proof. It is clear that $\tilde{\mathcal{H}}$ is complete, since a Cauchy sequence in the modified norm is also Cauchy in the original norm and therefore converges to an element of $\mathcal{H}$. For the same reason it is clear that point evaluations are bounded as $|f(x)| \leq C\|f\|_{\mathcal{H}}$ implies that $|f(x)| \leq C\|f\|_{\tilde{\mathcal{H}}}$.

We will be interested in the case when S and $\mathcal{V}$ depend on the data. We notice that while Proposition 2.1 is very general, and holds for any choice of S and $\mathcal{V}$, it is not usually easy to connect the kernels k and $\tilde{k}$.

However, as we will show below, for a class of what may be termed "point-cloud norms" this connection can be expressed explicitly.

Given the data points $x_1, \ldots, x_n$, let $S : \mathcal{H} \to \mathbb{R}^n$ be the evaluation map $S(f) = (f(x_1), \ldots, f(x_n))$. Denote $\mathbf{f} = (f(x_1), \ldots, f(x_n))$. The (semi-)norm on $\mathbb{R}^n$ will be given by a symmetric positive semi-definite matrix M:

$$\|Sf\|^2_{\mathcal{V}} = \mathbf{f}^t M \mathbf{f}$$

We will derive the exact form for $\tilde{k}(x,z)$. Note that $\tilde{\mathcal{H}}$ can be orthogonally decomposed as

$$\tilde{\mathcal{H}} = span\left\{\tilde{k}(x_1,\cdot), \ldots, \tilde{k}(x_n,\cdot)\right\} \oplus \tilde{\mathcal{H}}^\perp$$

where $\tilde{\mathcal{H}}^\perp$ consists of functions vanishing at all data points. It is clear that for any $f \in \tilde{\mathcal{H}}^\perp$, $Sf = 0$ and therefore $\langle f, g\rangle_{\tilde{\mathcal{H}}} = \langle f, g\rangle_{\mathcal{H}}$ for any function g in the space.

We therefore see that for any such $f \in \tilde{\mathcal{H}}^\perp$, we have

$$\begin{aligned} f(x) &= \langle f, \tilde{k}(x,\cdot)\rangle_{\tilde{\mathcal{H}}} \ \text{(reproducing property in } \tilde{\mathcal{H}}) \\ &= \langle f, k(x,\cdot)\rangle_{\mathcal{H}} \ \text{(reproducing property in } \mathcal{H}) \\ &= \langle f, k(x,\cdot)\rangle_{\tilde{\mathcal{H}}} \ \ \text{since } f \in \tilde{\mathcal{H}}^\perp \end{aligned}$$

Thus, for any $f \in \tilde{\mathcal{H}}^\perp$, we have $\langle f, k(x,\cdot) - \tilde{k}(x,\cdot)\rangle_{\tilde{\mathcal{H}}} = 0$ or $k(x,\cdot) - \tilde{k}(x,\cdot) \in (\tilde{\mathcal{H}}^\perp)^\perp$. In other words,

$$k(x,\cdot) - \tilde{k}(x,\cdot) \in span\left\{(\tilde{k}(x_1,\cdot), \ldots, \tilde{k}(x_n,\cdot)\right\}$$

On the other hand, for any $x_i \in X$ and $f \in \tilde{\mathcal{H}}^\perp$ from the definition of the inner product on $\tilde{\mathcal{H}}$ we see $\langle k(x_i,\cdot), f\rangle_{\tilde{\mathcal{H}}} = 0$. Thus, $k(x_i,.) \in (\tilde{\mathcal{H}}^\perp)^\perp$. Therefore, we see that

$$span\{k(x_i,\cdot)\}_{i=1}^n \subseteq span\{(\tilde{k}(x_i,\cdot)\}_{i=1}^n$$

Also decomposing, $\mathcal{H} = span\,\{k(x_i,\cdot)\}_{i=1}^n \oplus \mathcal{H}^\perp$, it is easy to check that $\tilde{k}(x_i,.) \in (\mathcal{H}^\perp)^\perp$ so that:

$$span\{\tilde{k}(x_i,\cdot)\}_{i=1}^n \subseteq span\{k(x_i,\cdot)\}_{i=1}^n$$

Thus, the two spans are same and we conclude that

$$\tilde{k}(x,\cdot) = k(x,\cdot) + \sum_j \beta_j(x) k(x_j,\cdot)$$

where the coefficients β_j depend on x.

To find $\beta_j(x)$, we look at a system of linear equations generated by evaluating $k(x_i,.)$ at x:

$$\begin{aligned} k_{x_i}(x) &= \langle k(x_i,.), \tilde{k}(x,\cdot)\rangle_{\tilde{\mathcal{H}}} \\ &= \langle k(x_i,.), k(x,\cdot) + \sum_j \beta_j(x) k(x_j,\cdot)\rangle_{\tilde{\mathcal{H}}} \\ &= \langle k(x_i,.), k(x,\cdot) + \sum_j \beta_j(x) k(x_j,\cdot)\rangle_{\mathcal{H}} \\ &\quad + \ \ \mathbf{k_{x_i}}^t M \mathbf{g} \end{aligned}$$

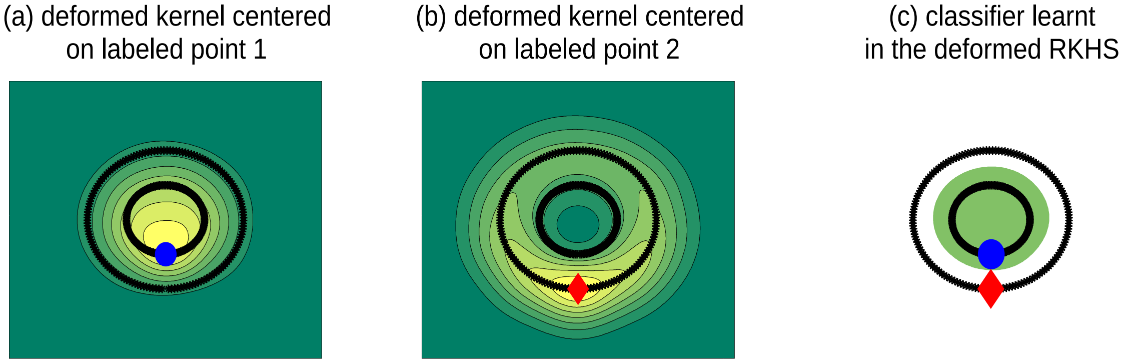
where $\mathbf{k_{x_i}} = (k(x_i,x_1)\ldots k(x_i,x_n))^t$ and $\mathbf{g}$ is the vector given by the components $\mathbf{g}_k = k(x,x_k) + \sum_j \beta_j(x) k(x_j,x_k)$. This formula provides the following system of linear equations for the coefficients $\beta(x) = (\beta_1(x)\ldots\beta_n(x))^T$:

$$(I + MK)\beta(x) = -M\mathbf{k}_x$$

where K is the matrix $K_{ij} = K(x_i,x_j)$ and $\mathbf{k}_x$, as before, denotes the vector $(k(x_1,x)\ldots k(x_n,x))^t$.

Finally, we obtain the following explicit form for $\tilde{k}$:

Figure 3.



Proposition 2.2 *Reproducing kernel of $\tilde{\mathcal{H}}$:*

$$\tilde{k}(x,z) = k(x,z) - \mathbf{k}_x^t (I + MK)^{-1} M \mathbf{k}_z$$

One can observe that the matrix $(I + MK)^{-1}M$ is symmetric. When M is invertible, it equals $(M^{-1} + K)^{-1}$ which is clearly symmetric. When M is singular, one adds a small ridge term to M and then uses a continuity argument.

We see that modifying the RKHS with a point-cloud norm deforms the kernel along a finite-dimensional subspace given by the data.

2.1. Choosing the Point Cloud Norm

The key issue now is the choice of M, so that the deformation of the kernel induced by the data-dependent norm, is motivated with respect to our intuitions about the data. Such intuitions may be inspired by forms of prior knowledge (e.g, transformation invariances), or, in the case of semi-supervised learning, by the form of the marginal distribution as described by unlabeled data.

In this paper, we will follow (Belkin, Niyogi & Sindhwani, 2004; Sindhwani, 2004) and utilize the Laplacian associated to the point cloud. This choice implements a smoothness assumption with respect to an empirical estimate of the geometric structure of the marginal distribution.

We set $M = L^p$, where p is an integer and L is the Laplacian matrix of a graph that models the underlying geometry. The graph Laplacian is defined as $L = D - W$ where $W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$, if x_i and x_j are adjacent and zero otherwise and D is a diagonal degree matrix given by $D_{ii} = \sum_i W_{ij}$. The graph Laplacian provides the following smoothness penalty on the graph: $\mathbf{f}^t L \mathbf{f} = \sum_{i,j=1}^n (f(x_i) - f(x_j))^2 W_{ij}$. Typically we use nearest neighbors to construct the matrix. The neighborhood relationship can vary depending on our understanding of the data. This matrix implements an empirical version of the Laplace-Beltrami operator, when the underlying space is a manifold. See (Belkin,2003; Lafon, 2004) for more details.

2.2. Back to Concentric Circles

The result of modifying the kernel by using the graph Laplacian for the particular case of two circles[1] is shown in Fig. 3.

In Fig. 3(a) and Fig. 3(b) we see level lines for modified kernels centered on two points on smaller and larger circles respectively. We see that as expected the kernel becomes extended along the circle. This distortion of the kernel reflects two intuitions about the natural data: what may be termed "the manifold assumption", i.e. the notion that our regression/classification function is smooth with respect to the underlying probability distribution and the related "cluster assumption" (see e.g (Chapelle & Zien, 2005)), which suggests that classes form distinct "clusters" separated by low density areas. The kernel, such as shown in Fig. 3, heavily penalizes changes along the circle, while imposing little penalty on changes in the orthorgonal direction.

Finally, Fig.3(c) shows the class boundary obtained using this new kernel.

2.3. Algorithms

By setting $M = \frac{\gamma_I}{\gamma_A} L^p$ the modified kernel allows us to reconstruct algorithms for semi-supervised classification presented in (Belkin, Niyogi & Sindhwani, 2004) and re-interpret them within the standard framework

[1] Each consisting of 150 evenly spaced unlabeled points, with one labeled example

of kernel methods. γ_A and γ_I are regularization parameters whose ratio controls the extent of the deformation. In particular, Laplacian SVM (LapSVM) and Laplacian RLS (LapRLS) become standard RLS and SVM using these kernels. One can also employ a suite of kernel-based algorithms, such as support vector regression and one-class SVM, together with their optimization strategies and implementations to solve various learning problems. Additionally, based on different choices of M, our approach provides a general algorithmic framework for incorporating useful domain structures (e.g invariances) in kernel methods.

3. Related Work

Kernel methods (Schoelkopf & Smola, 2002; Vapnik, 1998) have become very popular in recent years. There is a large body of literature on how to choose the parameters of a kernel based on data, e.g by cross-validation. However, these methods, which include some classical statistical procedures, require an a-priori parametrization of the kernel.

Related work on developing data-dependent kernels on the set of labeled and unlabeled examples includes (Belkin, Matveeva & Niyogi, 2004; Joachims, 2003; Zhou et al, 2004; Zhu, Gharamani & Lafferty 2003) and references therein.

In particular, we note (Chapelle, Weston & Schoelkopf, 2003; Chapelle & Zien, 2005) where kernels are designed to implement the cluster assumption.

These approaches do not define an RKHS whose domain is the whole space X. Some related ideas for out of sample extension were proposed in (Delalleau, Bengio & Le Roux, 2005; Vert & Yamanishi, 2005). Also see (Smola & Schoelkopf, 1998; Bousquet, Chapelle & Hein, 2004) for related theoretical frameworks.

Algorithmic extensions of SVMs to handle unlabeled data have been proposed, e.g in (Joachims, 1999). Manifold learning was recently combined with boosting in (Kegl & Wang, 2005). Finally, (Wu & Amari, 2002) produced a family of data dependent kernels with some interesting geometric interpretations.

4. Experiments

The purpose of the experiments is to evaluate the quality of transductive and semi-supervised learning with SVMs and RLS using the data-dependent semi-supervised kernel in comparison to their standard versions. Additional comparisons are made based on results reported in (Chapelle & Zien, 2005) and (Joachims, 2003) with transductive graph methods (abbreviated Graph-Trans) such as Graph Regularization (Belkin, Matveeva & Niyogi, 2004) and Spectral Graph Transduction (SGT) (Joachims, 2003); the implementation of Transductive SVMs (Vapnik, 1998) in (Joachims, 1999) (TSVM) and in (Chapelle & Zien, 2005) (∇TSVM)); and with other methods proposed in (Chapelle & Zien, 2005) : training an SVM on a graph-distance derived kernel (Graph-density) and Low Density Separation (LDS). We will also observe the roles of the original RKHS norm and the point cloud norms in empirical performance. For the purpose of reproducability, the matlab scripts and datasets used in these experiments are available at :
http://www.cs.uchicago.edu/∼vikass/research.html.

Data Sets

Experiments were performed on five well-known datasets described in Table 1.

Table 1. Datasets used in the experiments : c is the number classes, d is the data dimensionality, l is the number of labeled examples, n is the total number of examples in the dataset from which labeled, unlabeled and test examples, when required, are drawn.

Dataset	c	d	l	n
g50c	2	50	50	550
Coil20	20	1024	40	1440
Uspst	10	256	50	2007
mac-windows	2	7511	50	1946
Webkb (page)	2	3000	12	1051
Webkb (link)	2	1840	12	1051
Webkb (page+link)	2	4840	12	1051

g50c is an artificial dataset generated from two unit-covariance normal distributions with equal probabilities. The class means are adjusted so that the true bayes error is 5%, and 550 examples are drawn. Coil20 and Uspst datasets pose multiclass image classification problems. Coil20 consists of 32×32 gray scale images of 20 objects viewed from varying angles and Uspst is taken from the USPS (test) dataset for handwritten digit recognition. The text data consists of binary classification problems: mac-win is taken from the 20-newsgroups dataset and the task is to categorize newsgroup documents into two topics: *mac* or *windows*; the WebKB dataset is a subset of web documents of the computer science departments of four universities. This dataset has been extensively used for semi-supervised learning experiments (Joachims, 2003; Nigam, 2001). The two categories are *course* or *non-course*. For each document, there are two representations: the textual content of the webpage (which we will call *page* representation) and the anchortext on links on other webpages pointing to the webpage (*link* representation). Following (Nigam, 2001), we generated bag-of-words feature vectors for both representations as follows: Documents were tokenized using the Rainbow Text toolkit (McAllum, 1996) ; HTML-tagged content was skipped and no stoplist or stemming was used; numbers were included in the tokenization. For the page representation, 3000 features were selected according to information gain. For the link representation, 1840 features were generated with no feature selection. The columns of the document-word matrix were scaled based on inverse document frequency weights (IDF) for each word and the resulting TFIDF feature vectors were length normalized. We also considered a joint (*page+link*) representation by concatenating the features.

In the discussion ahead, by a *training set* we will mean the union of the *labeled set* and the *unlabeled set* of examples available to transductive and semi-supervised learners. *Test sets* comprise of examples never seen before.

Transductive Setting

In the transductive setting, the training set comprises of n examples, l of which are labeled (n, l are specified in Table 1). In Table 2, we lay out a performance comparison of several algorithms in predicting the labels of the $n - l$ unlabeled examples. The experimental protocol is based on (Joachims, 2003) for the WebKB dataset and (Chapelle & Zien, 2005) for other datasets.

Protocol: For datasets other than WebKB, performance is evaluated by error rates averaged over 10 random choices of the labeled set. Each random set samples each class at least once (twice for coil20). Results for Graph-Reg, TSVM,∇TSVM,Graph-density, and LDS are taken from (Chapelle & Zien, 2005) where models were selected by optimizing error rates on the unlabeled set giving these methods an unfair advantage. For, LDS, a cross-validation protocol was used in (Chapelle & Zien, 2005). For LapRLS, LapSVM we preferred to fix $\gamma_A = 10^{-6}, \gamma_I = 0.01$ to reduce the complexity of model selection. Gaussian base kernels and euclidean nearest neighbor graphs with gaussian weights were used. The three parameters : number of nearest neighbors (nn), the degree (p) of the graph Laplacian, and the width (σ) of the Gaussian are chosen based on 5-fold cross-validation performance in a small grid of parameter values. Together, these parameters specify the deformed kernel that incorporates the unlabeled data.

For WebKB, we evaluated performance by precision-recall breakeven points. Linear Kernels and cosine nearest neighbor graphs with gaussian weights were used. In this case, we fixed $nn = 200$ (as in (Joachims, 2003)), $p = 5$ (unoptimized), and σ as the mean edge length in the graph. Since the labeled set is very small for this dataset, we performed model selection (including γ_A, γ_I for LapSVM, LapRLS) for all algorithms by optimizing performance on the unlabeled set.

Discussion: Using the proposed kernel, SVM and RLS return the best performance in four of the five datasets. In g50c, performance is close to the bayes optimal. We obtain significant performance gains on Coil20 and Uspst where there are strong indications of a manifold structure. On WebKB, the methods outperform other methods in the *page+link* representation. We also tried the following novel possibility: the point cloud norm was constructed from the mean graph Laplacian over the three representations and used for deforming RKHS in each representation. With this multi-view regularizer, the method significantly outperforms all other methods for all representations. Finally, note that one can recover the original base kernel by setting $\gamma_I = 0$. With a good model selection, the proposed methods should never perform worse than inductive methods.

Semi-supervised Setting

In the semi-supervised setting, the training set comprises of $l + u$ examples (l labeled as before and u unlabeled) and the test set comprises of $n - l - u$ examples. Experiments were performed to observe the performance of LapSVM and LapRLS on the test and unlabeled sets to see how well these methods extend to novel out-of-sample examples.

Protocol: We performed a variation of 4-fold cross-validation. The data was divided into four equal chunks: three chunks were combined to form the training set and the remaining formed the test set. Each chunk therefore appeared in the training data thrice and as a test set once. Table 3,4 report mean performance of LapSVM and LapRLS in predicting the labels of each chunk as a subset of the unlabeled set and as a test set. γ_A, γ_I are optimized for best mean performance; and the other parameters are set as before. For WebKB, is is natural for the four chunks to correspond to the four universities: training on three universities and testing on the fourth. The detailed performance for each university is reported in Table 3 for LapSVM (performance is similar for LapRLS).

Discussion: For g50c, mac-win, and WebKB the performance on unlabeled and test subsets is almost in-

Table 2. Transductive Setting: Error Rates (100-PRBEP for WebKb) on unlabeled examples. Results on which Laplacian SVMs (LapSVM) and Laplacian RLS (LapRLS) outperform all other methods are shown in bold. LapSVM$_{joint}$, LapRLS$_{joint}$ use the sum of graph laplacians in each WebKB representation. Results for Graph-Trans, TSVM,∇TSVM,Graph-density, and LDS are taken from (Chapelle & Zien, 2005)

Dataset →	g50c	Coil20	Uspst	mac-win	WebKB	WebKB	WebKB
Algorithm ↓					(link)	(page)	(page+link)
SVM (full labels)	4.0 (2.9)	0.0 (0.0)	2.8 (0.8)	2.4 (1.3)	5.1 (2.8)	5.3 (4.0)	0.7 (1.4)
RLS (full labels)	4.0 (2.7)	0.0 (0.0)	2.5 (1.3)	2.8 (1.7)	5.6 (2.8)	6.4 (3.8)	2.2 (3.0)
SVM (l labels)	9.7 (1.7)	24.6 (1.7)	23.6 (3.3)	18.9 (5.7)	28.1 (16.1)	24.3 (15.0)	18.2 (15.5)
RLS (l labels)	8.5 (1.5)	26.0 (1.5)	23.6 (3.5)	18.8 (5.7)	30.3 (16.5)	30.2 (15.3)	23.9 (16.1)
Graph-Trans	17.3	6.2	21.3	11.7	22.0	10.7	6.6
TSVM	6.9	26.3	26.5	7.4	14.5	8.6	7.8
Graph-density	8.3	6.4	16.9	10.5	-	-	-
∇TSVM	5.8	17.6	17.6	5.7	-	-	-
LDS	5.6	4.9	15.8	5.1	-	-	-
LapSVM	**5.4** (0.6)	**4.0** (2.3)	**12.7** (2.3)	10.4 (1.1)	17.2 (9.0)	10.9 (1.2)	**6.4** (0.9)
LapRLS	**5.2** (0.7)	**4.3** (1.3)	**12.7** (2.4)	10.0 (1.3)	19.2 (10.0)	11.2 (1.1)	7.5 (1.4)
LapSVM$_{joint}$	-	-	-	-	**5.7** (1.5)	**6.6** (1.3)	**5.1** (0.9)
LapRLS$_{joint}$	-	-	-	-	**6.7** (6.2)	**8.9** (3.9)	**5.9** (2.9)

Table 3. Semi-supervised Setting: (WebKB) 100-PRBEP on unlabeled and test examples

View →	link	page	page+link
University ↓	unlab	unlab	unlab
	test	test	test
Cornell	26.1	14.4	8.0
	27.3	14.3	8.0
Texas	18.8	19.0	4.7
	17.3	17.8	5.1
Washington	12.8	8.7	4.8
	13.8	8.4	4.5
Wisconsin	18.6	14.5	7.1
	19.3	15.7	7.0

Table 4. Semi-supervised Setting: Error rates on unlabeled and test examples.

Dataset →	g50c	Coil20	Uspst	mac-win
Algorithm ↓	unlab	unlab	unlab	unlab
	test	test	test	test
SVM	9.7	21.7	21.6	20.9
	9.7	22.6	22.1	20.9
RLS	9.1	21.8	22.5	20.9
	9.6	22.6	23.0	20.4
LapSVM	4.9	8.7	14.9	9.9
	5.0	14.6	17.7	9.7
LapRLS	4.9	9.40	14.3	9.4
	4.9	12.9	17.0	9.3

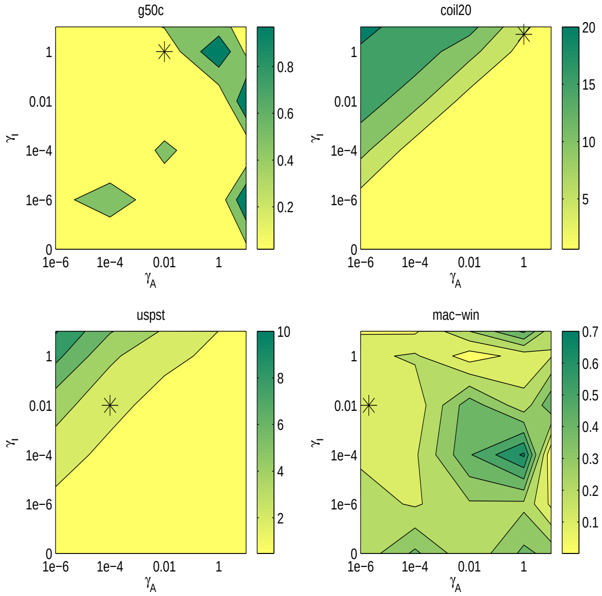
distinguishable. The out-of-sample extension is high quality also for Uspst. For Coil20, we observe an over-deformation phenomenon : the in-sample performance is significantly better than out-of-sample performance. A smoother base kernel and appropriate degree of deformation can remove this difference for coil20.

Parameters of Deformation

The parameters γ_A, γ_I specify a trade-off between ambient regularization and deformation. In Fig 4 we show the performance difference over test sets and unlabeled subsets as a function on the γ_A, γ_I plane. Also shown is the location of the optimal γ_A, γ_I. For a wide range of parameter settings, the performance difference is less than 1% for g50c and mac-win, and less than 2% for coil20 and uspst. In uspst and coil20 we see an expected behaviour : When γ_I is much larger than γ_A, the point cloud norm dominates the regularization and the in-sample performance is found to be much better than the out-of-sample performance. When γ_A is increased, the difference decreases. In general, the optimal performance strikes a good balance between the ambient norm and the degree of deformation.

Further experimental observations would be required to understand the nature of these deformations and the choice of deformation parameters.

Figure 4. Difference in Error Rates (in percentage on the vertical colorbar) over test sets and unlabeled subsets in the $\gamma_I - \gamma_A$ plane. The optimal mean performance is obtained at the point marked by a black star.



5. Conclusion

We have shown how to warp an RKHS to adapt to the geometry of the data in machine learning tasks. Our framework has particular applicability to the problem of semi-supervised learning, where we have demonstrated state of the art empirical performance. This framework also permits us to incorporate various other domain structures in a large class of algorithms. We will pursue these in future work.

References

Belkin M. (2003) *Problems of Learning on Manifolds.* Ph.D. Dissertation, Dept. of Mathematics, Univ. of Chicago

Belkin M., Niyogi P. & Sindhwani V. (2004) *Manifold Regularization : A Geometric Framework for Learning for Examples.* Technical Report TR-2004-06, Dept. of Computer Science, Univ. of Chicago

Belkin M., Matveeva I., & Niyogi P. (2004) *Regression and Regularization on Large Graphs.* COLT

Bousquet O., Chapelle O. & Hein, M. (2004) *Measure Based Regularization.* NIPS 16

Chapelle O., Weston J., & Schoelkopf B. (2003) *Cluster Kernels for Semi-Supervised Learning.* NIPS 15

Chapelle O. & Zien A. (2005) *Semi-Supervised Classification by Low Density Separation.* AI & Statistics

Delalleau O., Bengio Y. & Le Roux N. (2005) *Efficient Non-Parametric Function Induction in Semi-Supervised Learning.* AI & Statistics

Joachims T. (1999) *Transductive Inference for Text Classification using Support Vector Machines.* ICML

Joachims T. (2003) *Transductive Learning via Spectral Graph Partitioning.* ICML

Kegl B. & Wang L. (2005) *Boosting on manifolds: adaptive regularization of base classifiers.* NIPS 18

Lafon S. (2004) *Diffusion maps and geometric harmonics.* Ph.D. dissertation, Yale University

Nigam, K. (2001) *Using Unlabeled Data to Improve Text Classification.* Doctoral Dissertation, Carnegie Mellon University. TR CMU-CS-01-126

McCallum, A. K. (1996) *Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering* http://www.cs.cmu.edu/∼mccallum/bow

Schoelkopf, B. & Smola A.J. (2002) *Learning with Kernels.* MIT Press, Cambridge, MA

Sindhwani, V. (2004) *Kernel Machines for Semi-supervised Learning.* Masters Thesis, University of Chicago

Smola, A.J., and Schoelkopf B. (1998) *On a Kernel-Based Method for Pattern Recognition, Regression, Approximation, and Operator Inversion.* Algorithmica , Vol. 22, No. 1/2, 211-231.

Wu, S. & Amari, S. (2002) *Conformal Transformation of Kernel Functions: A Data-Dependent Way to Improve Support Vector Machine Classifiers.* Neural Processing Letters, 15, pp. 59-67

Vapnik, V. (1998) *Statistical Learning Theory.* Wiley-Interscience

Vert, J. P. and Yamanishi, Y. (2005), *Supervised graph inference* NIPS 18

Zhou, D., Bousquet, O., Lal, T. N., Weston J., & Schoelkopf, B. (2004) *Learning with Local and Global Consistency.* NIPS 16

Zhu, X., Ghahramani, Z., & Lafferty J. (2003) *Semi-supervised Learning using Gaussian Fields and Harmonic Functions.* ICML