Manifold Regularization for Structured Outputs via the Joint Kernel

Chonghai Hu and James T. Kwok

***Abstract*— By utilizing the label dependencies among both the labeled and unlabeled data, semi-supervised learning often has better generalization performance than supervised learning. In this paper, we extend a popular graph-based semi-supervised learning method, namely, manifold regularization, to structured outputs. This is performed via the joint kernel directly and allows a unified manifold regularization framework for both unstructured and structured data. Experimental results on various data sets with inter-dependent outputs demonstrate the usefulness of manifold information in improving prediction performance.**

I. Introduction

Kernel methods, such as support vector machines, have been highly successful in machine learning. Traditionally, they are mainly focused on vectorial inputs and outputs. With the tremendous amount of structured data (e.g., sequences, trees, and graphs) available nowadays, there are a lot of recent interests in extending kernel methods for more complex domains with these structured data. In general, the structure information may be present in the inputs and/or outputs. For structured inputs, a wide variety of kernels have been developed. Examples include the family of string kernels commonly used in bioinformatics. Here, we will focus on kernel methods for structured outputs. In contrast to traditional kernels that are defined on the inputs only, an essential ingredient in learning with structured outputs is that the kernel is often defined jointly on both the inputs and outputs [1]. This so-called *joint kernel* allows the many-sided dependencies between inputs and outputs to be captured. Empirically, it has been successfully used for solving many complex structured prediction problems in domains such as natural language processing and computational biology.

On the other hand, while many of these structured data are readily available (such as internet documents residing in a hierarchy and bioinformatics databases containing DNA sequences), typically most of them are unlabeled and only a small amount of the data is labeled. Semi-supervised learning [2], [3] thus aims at improving the generalization performance by utilizing both the labeled and unlabeled data. The label dependencies among patterns are captured by exploiting the intrinsic geometric structure of the data. This can be implemented by using the so-called cluster assumption, which encourages the separating hyperplane to pass through low-density regions [4], [5]. Recently, this is also extended to structured outputs [6]. Another popular smoothness assumption is the manifold assumption, which assumes that the data lie on a low-dimensional manifold. Often, this manifold is approximated by a weighted graph, leading to a battery of graph-based semi-supervised learning algorithms [7], [8]. Besides these, techniques based on co-training [9] and conditional random fields [10], [11] have also been used for semi-supervised learning on structured outputs.

In this paper, we will focus on the graph-based approach, and, in particular, the manifold regularization framework [7]. By defining a data-dependent reproducing kernel Hilbert space (RKHS), manifold regularization incorporates an additional regularizer to ensure that the learned function is smooth on the manifold. Moreover, in contrast to many other graph-based transductive learning methods, this regularization framework is truly semi-supervised and allows generalization to out-of-sample patterns.

Recently, an extension of manifold regularization to structured outputs has been proposed in [12]. However, instead of learning a smooth discriminant function over the input-output pairs directly, it learns a *goodness* function of each explicit "part" of the joint feature map. Moreover, the discriminant function is a weighted sum of predictions from the individual parts. Hence, this can be regarded as a "bag of parts" representation. Analogous to the commonly used "bag of words" representation for text, the structure information among parts is lost. Similarly, the smoothness of the discriminant function over the data manifold is only indirectly enforced through these parts. Besides, it cannot be readily reduced to the standard manifold regularization method for unstructured data.

In this paper, we perform manifold regularization on structured outputs by using the joint kernel directly. The extension is formulated entirely in terms of the joint kernel and thus, as in other kernel methods, does not require knowledge of the explicit (joint) feature map. Moreover, it includes standard manifold regularization as a special case and hence provides a unified framework for both unstructured and structured data.

The rest of this paper is organized as follows. Section II first gives a brief review on the supervised learning of structured outputs. Section III then extends manifold regularization for semi-supervised learning on structured outputs via the joint kernel. This is then followed by some discussions in Section IV. Experimental results are presented in Section V, and the last section gives some concluding remarks.

In the sequel, I_m denotes the $m \times m$ identity matrix, $1_m =$

Chonghai Hu is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong; and the Department of Mathematics, Zhejiang University, Hangzhou, China (email: hino.hu@gmail.com).

James T. Kwok is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (email: jamesk@cse.ust.hk).

$[1,\ldots,1]' \in \mathbb{R}^m$, and $\mathbb{R}_+$ is the set of non-negative real numbers. Moreover, $\mathrm{tr}(A)$ is the trace of matrix A, and $A \otimes B$ is the Kronecker product of the two matrices. Besides, superscripts (or subscripts) $\mathcal{X}, \mathcal{Y}$ and $\mathcal{XY}$ will be used to denote entities corresponding to the input, output and the joint spaces, respectively.

II. Supervised Learning of Structured Outputs

As mentioned in Section I, for learning with structured outputs, it is often more convenient to use a joint feature representation φ that is defined on both the input $\mathcal{X}$ and output $\mathcal{Y}$. The discriminant function is then linear in this joint feature map, as

$$f(x,y) = w'\varphi(x,y). \tag{1}$$

Moreover, as in other kernel methods, this joint feature map is related to a *joint kernel* k as $k((x,y),(\bar{x},\bar{y})) = \varphi(x,y)'\varphi(\bar{x},\bar{y})$.

Given a set of training patterns $\{(x_i,y_i)\}_{i=1}^{\ell}$ where $y_i \in \mathcal{Y}$, the desired discriminant function $w'\varphi(x,y)$ can be obtained by solving

$$\begin{aligned} \min \quad & \frac{1}{\ell}\sum_{i=1}^{\ell}\xi_i + \gamma_A\|w\|^2 \\ \text{s.t.} \quad & w'\delta\varphi_i(y) \geq 1 - \frac{\xi_i}{\Delta(y_i,y)}, \ \forall i, \forall y \neq y_i, \quad \xi_i \geq 0, \ \forall i, \end{aligned} \tag{2}$$

where ξ_i's are slack variables for the errors, $\delta\varphi_i(y) \equiv \varphi(x_i,y_i) - \varphi(x_i,y)$, $\Delta(y_i,y)$ is a loss function penalizing the difference between y_i and y, and γ_A is a user-defined regularization parameter. Note that the slack variables in (2) are scaled with the inverse loss, which is often called *slack re-scaling*. Another approach, as advocated in [13], is called *margin re-scaling* and scales the margin by the loss.

Using the method of Lagrange multipliers, the dual of (2) can be obtained as:

$$\begin{aligned} \max \quad & \sum_{i,y\neq y_i}\alpha_{iy} - \frac{1}{2}\sum_{i,y\neq y_i}\sum_{j,\bar{y}\neq y_j}\alpha_{iy}\alpha_{j\bar{y}}J_{iy,j\bar{y}} \\ \text{s.t.} \quad & \sum_{y\neq y_i}\frac{\alpha_{iy}}{\Delta(y_i,y)} \leq \frac{1}{2\ell\gamma_A}, \ \forall i, \\ & \alpha_{iy} \geq 0, \ \forall i, \forall y \neq y_i, \end{aligned} \tag{3}$$

where

$$\begin{aligned} J_{iy,j\bar{y}} &= \delta\varphi_i(y)'\delta\varphi_j(\bar{y}) \\ &= k((x_i,y_i),(x_j,y_j)) - k((x_i,y_i),(x_j,\bar{y})) \\ &\quad -k((x_i,y),(x_j,y_j)) + k((x_i,y),(x_j,\bar{y})). \end{aligned}$$

It can be shown that $\frac{1}{\ell}\sum_i \xi_i$ is an upper bound of the empirical risk at optimality. Moreover, while the optimization problem may potentially have an exponential number of constraints, an arbitrarily close approximation to the solution can be efficiently obtained in polynomial time by the use of cutting plane methods [1].

III. Semi-Supervised Learning of Structured Outputs

In semi-supervised learning, we have both labeled patterns $\{(x_i,y_i)\}_{i=1}^{\ell}$ drawn i.i.d. from the joint distribution $P_{\mathcal{XY}}$ and a set of unlabeled examples $\{x_i\}_{i=\ell+1}^{n}$ drawn i.i.d. from the marginal distribution $P_{\mathcal{X}}$. A typical smoothness assumption used in many semi-supervised learning methods is that two nearby patterns in a high-density region should share similar labels [2]. When the data lie on a manifold, it is common to represent this manifold by a weighted graph defined on all the labeled and unlabeled data. The graph's weight matrix $A = [a_{ij}]$ encodes the similarities between data samples. Label smoothness over the graph is then enforced by the so-called *manifold regularizer*, which controls the gradient of the target function w.r.t. the marginal distribution of the data. Empirically, this manifold regularizer can be approximated as

$$\sum_{i,j=1}^{n} a_{ij}(f(x_i) - f(x_j))^2 = f'Lf, \tag{4}$$

where $f = [f(x_1),\ldots,f(x_n)]'$ and L is the graph Laplacian matrix.

A. Primal Formulation

In this section, we extend the manifold regularization framework to structured outputs via the use of the joint kernel introduced in Section II. Unlike traditional graph-based semi-supervised learning methods where the nodes of the graph are the labeled and unlabeled examples $\{x \in \mathcal{X}\}$, here the nodes in this "joint manifold" are the tuples $\{(x,y) : x \in \mathcal{X}, y \in \mathcal{Y}\}$. Consequently, smoothness on the manifold is also dependent on both the input x and output y, and the similarity between two graph nodes (x_i,y^k) and (x_j,y^l) is now $a((x_i,y^k),(x_j,y^l))$. Note that the similarities defined on all the graph nodes can be put into a two-dimensional matrix A, with entries

$$\begin{aligned} A_{iy^kjy^l} &\equiv A(i+(k-1)|\mathcal{Y}|, j+(l-1)|\mathcal{Y}|) \\ &= a((x_i,y^k),(x_j,y^l)). \end{aligned}$$

As in Section II, the discriminant function $f(x,y)$ for structured data is dependent on both x and y. We require this f to be smooth on the (joint) manifold w.r.t. the similarity matrix A. Let D be the diagonal matrix with elements $D_{iy} = \sum_{j=1}^{n}\sum_{\bar{y}\in\mathcal{Y}} A_{iyj\bar{y}}$. The manifold regularizer in (4) can thus be extended to structured outputs as:

$$\begin{aligned} &\frac{1}{2}\sum_{y,\bar{y}\in\mathcal{Y}}\sum_{i,j=1}^{n} A_{iyj\bar{y}}(f(x_i,y) - f(x_j,y))^2 \\ &= \frac{1}{2}\sum_{y,\bar{y}\in\mathcal{Y}}\sum_{i,j=1}^{n} A_{iyj\bar{y}}(w'\varphi(x_i,y) - w'\varphi(x_j,y))^2 \\ &= w'\Phi L\Phi' w, \end{aligned} \tag{5}$$

where $\Phi = [\varphi(x_i,y)]_{i=1,\ldots,n;y\in\mathcal{Y}}$ and $L = D - A$ is the corresponding (joint) graph Laplacian. Obviously, this includes the regularizer in (4) as a special case.

Recall from Section II that for structured prediction, the loss function has to be incorporated into structural risk minimization. In this paper, we will focus on the slack re-scaling formulation. Extension to margin re-scaling is straight-forward and will not be discussed here. Adding the manifold regularizer in (5) to the supervised structured prediction problem of (2), we obtain the following optimization problem which corresponds to the manifold regularization for structured outputs:

$$\begin{aligned} \min_{w,\xi\geq 0} \quad & \frac{1}{\ell}\sum_{i=1}^{\ell}\xi_i + \gamma_A w'w + \gamma_I w'\Phi L\Phi' w \qquad (6)\\ \text{s.t.} \quad & w'(\varphi(x_i,y_i)-\varphi(x_i,y)) \geq 1-\frac{\xi_i}{\Delta(y_i,y)},\\ & i=1,\ldots,\ell, y\neq y_i. \end{aligned}$$

Here, $\xi = [\xi_1,\ldots,\xi_\ell]'$ and $\gamma_I > 0$ is a tradeoff parameter for the manifold regularizer. Obviously, in the special case where $\gamma_I = 0$, (6) reduces to (2).

By using a similar orthogonality argument as in [7], it is easy to obtain the following representer theorem:

Lemma 1: For problem (6), its optimal w is of the form $w = \sum_{i=1}^{n}\sum_{y\in\mathcal{Y}}\alpha_{iy}\varphi(x_i,y)$, where $\alpha_{iy}\in\mathbb{R}$.

In matrix form, this can be written as

$$w = \Phi\alpha, \qquad (7)$$

where $\alpha = [\alpha_{iy}]_{i=1,\ldots,n;y\in\mathcal{Y}}$. Substituting this into (6), we obtain the following optimization problem:

Proposition 1: Problem (6) is equivalent to

$$\begin{aligned} \min_{\alpha,\xi\geq 0} \quad & \frac{1}{\ell}\sum_{i=1}^{\ell}\xi_i + \alpha'(\gamma_A K + \gamma_I KLK)\alpha \qquad (8)\\ \text{s.t.} \quad & \alpha' K e_i\otimes(\bar{e}_{y_i}-\bar{e}_y) \geq 1-\frac{\xi_i}{\Delta(y_i,y)}\\ & i=1,\ldots,\ell, y\neq y_i, \end{aligned}$$

where $K = \Phi'\Phi$ is the kernel matrix defined using the joint kernel on the training data, $e_i\in\mathbb{R}^{n|\mathcal{Y}|}$ is a vector of all zeroes except that the ith entry is one, and $\bar{e}_y\in\mathbb{R}^{|\mathcal{Y}|}$ is a vector of all zeroes except that the yth entry is one.

Proof: On using (7), the $w'w$ and $w'\Phi L\Phi' w$ terms in the objective of (6) can be written as $\alpha' K\alpha$ and $\alpha' KLK\alpha$, respectively, As for its constraint, define $J_i = e_i\otimes I_{|\mathcal{Y}|}$. Then, $w'\varphi(x_i,y) = \alpha'\Phi'\Phi J_i\bar{e}_y = \alpha' K(e_i\otimes I_{|\mathcal{Y}|})\bar{e}_y = \alpha' K(e_i\otimes\bar{e}_y)$. Plugging all these back into (6), we obtain (8). ■

B. Dual Problem

Denote $\tilde{I}_i = [\bar{e}_y]_{y\neq y_i}$, and $\breve{I}_\ell = [e_1,\ldots,e_\ell]$. The dual of (8) can be obtained from the following proposition.

Proposition 2: The dual of (8) is

$$\begin{aligned} \max_{\beta} \quad & -\frac{1}{4}\beta' J'(\gamma_A K^{-1}+\gamma_I L)^{-1}J\beta + \beta' 1_{\ell(|\mathcal{Y}|-1)} \qquad (9)\\ \text{s.t.} \quad & \sum_{y\neq y_i}\frac{\beta_{iy}}{\Delta(y_i,y)} \leq \frac{1}{\ell}, \quad i=1,\ldots,\ell,\\ & \beta_{iy}\geq 0, \quad i=1,\ldots,\ell, y\neq y_i, \end{aligned}$$

where $\beta = [\beta_{iy}]_{i=1,\ldots,\ell;y\neq y_i}$, and $J = [e_1\otimes\bar{e}_{y_1},\ldots,e_\ell\otimes\bar{e}_{y_\ell}](\breve{I}_\ell\otimes 1'_{|\mathcal{Y}|-1}) - [e_1\otimes\tilde{I}_1,\ldots,e_\ell\otimes\tilde{I}_\ell]$. Moreover, the primal variable α can be recovered from the dual variable β as

$$\alpha = \frac{1}{2}(\gamma_A I_{n|\mathcal{Y}|}+\gamma_I LK)^{-1}J\beta. \qquad (10)$$

Proof: First, we obtain the Lagrangian of (8) as:

$$\begin{aligned} \mathcal{L} =& \frac{1}{\ell}\sum_{i=1}^{\ell}\xi_i + \alpha'(\gamma_A K+\gamma_I KLK)\alpha\\ &+\sum_{i=1}^{\ell}\sum_{y\neq y_i}\beta_{iy}\left[1-\frac{\xi_i}{\Delta(y_i,y)}-\alpha' K e_i\otimes\bar{e}_{y_i}+\alpha' K e_i\otimes\bar{e}_y\right]\\ &-\sum_{i=1}^{\ell}\gamma_i\xi_i, \qquad (11) \end{aligned}$$

where $\beta_{iy}\in\mathbb{R}_+, \gamma_i\in\mathbb{R}_+$ $(i=1,\ldots,\ell, y\neq y_i)$ are the Lagrange multipliers. Setting its derivatives w.r.t. ξ_i to zero, we have:

$$\frac{\partial\mathcal{L}}{\partial\xi_i} = \frac{1}{\ell}-\sum_{y\neq y_i}\frac{\beta_{iy}}{\Delta(y_i,y)}-\gamma_i = 0, \quad i=1,\ldots,\ell. \qquad (12)$$

Moreover, using the fact that $[e_i\otimes\bar{e}_{k_1},\ldots,e_i\otimes\bar{e}_{k_t}] = e_i\otimes[\bar{e}_{k_1},\ldots,\bar{e}_{k_t}]$, then, in (11),

$$\begin{aligned} \sum_{i=1}^{\ell}\sum_{y\neq y_i}\beta_{iy}e_i\otimes\bar{e}_y &= [e_i\otimes\bar{e}_y]_{i=1,\ldots,\ell;y\neq y_i}\beta\\ &= [e_1\otimes\tilde{I}_1,\ldots,e_\ell\otimes\tilde{I}_\ell]\beta. \end{aligned}$$

Similarly, the term $\sum_{i=1}^{\ell}\sum_{y\neq y_i}\beta_{iy}e_i\otimes\bar{e}_{y_i}$ in (11) can be written as

$$\begin{aligned} &[e_1\otimes\bar{e}_{y_1},\ldots,e_\ell\otimes\bar{e}_{y_\ell}]\begin{bmatrix}\sum_{y\neq y_1}\beta_{1y}\\ \vdots\\ \sum_{y\neq y_\ell}\beta_{\ell y}\end{bmatrix}\\ =\;& [e_1\otimes\bar{e}_{y_1},\ldots,e_\ell\otimes\bar{e}_{y_\ell}](\breve{I}_\ell\otimes 1'_{|\mathcal{Y}|-1})\beta. \end{aligned}$$

Moreover, the term $\sum_{i=1}^{\ell}\sum_{y\neq y_i}\beta_{iy}(\alpha' K e_i\otimes\bar{e}_{y_i}-\alpha' K e_i\otimes\bar{e}_y)$ in (11) can be simplified as $\alpha' KJ\beta$. Plugging these and (12) back into (11), we obtain

$$\mathcal{L} = \alpha'(\gamma_A K+\gamma_I KLK)\alpha - \alpha' KJ\beta + \beta' 1_{\ell(|\mathcal{Y}|-1)}. \qquad (13)$$

Setting its derivative w.r.t. α to zero, we obtain $\alpha = \frac{1}{2}(\gamma_A K+\gamma_I KLK)^{-1}KJ\beta$ and thus (10). Plugging (10) back into (13), we obtain (9). ■

C. Prediction

With the learned α, one can perform prediction on a (seen or unseen) example (x,y) as

$$f(x,y) = \sum_{i=1}^{n}\sum_{\bar{y}\in\mathcal{Y}}\alpha_{i\bar{y}}k((x,y),(x_i,\bar{y})). \qquad (14)$$

This is clearly advantageous to other graph-based transductive learning algorithms that cannot be used on unseen test data.

IV. Discussions

A. Data-Dependent Kernel Deformed by the Joint Manifold

By defining a data-dependent kernel that captures the underlying geometry of the data manifold, Sindhwani *et al.* [14] showed that the semi-supervised learning problem with traditional manifold regularization is equivalent to a supervised learning problem. This is also the case for the related approach of [12]. Here, we show that a similar kernel, defined over the joint space of $\mathcal{X}$ and $\mathcal{Y}$, can also be obtained in our semi-supervised structured prediction setting.

Let $k((\cdot,\cdot),(\cdot,\cdot))$ be the joint kernel function, and $\mathcal{H}$ the corresponding reproducing kernel Hilbert space (RKHS) with inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. Define a new space of functions $\widetilde{\mathcal{H}}$ with the same elements as $\mathcal{H}$, but with the modified inner product

$$\langle f,g\rangle_{\widetilde{\mathcal{H}}} = \langle f,g\rangle_{\mathcal{H}} + \frac{\gamma_I}{\gamma_A} F' L G, \tag{15}$$

where $F = [f(x_i,y)]_{i=1,\ldots,n;y\in\mathcal{Y}}$, and $G = [g(x_i,y)]_{i=1,\ldots,n;y\in\mathcal{Y}}$.

Lemma 2: $\widetilde{\mathcal{H}}$ is an RKHS.

Proof: Since $\mathcal{H}$ is an RKHS, there exists a constant $C>0$ that bounds the norm of the evaluation functional, i.e., $|f(x,y)| \le C\|f\|_{\mathcal{H}}$. Thus, by (15), we have $|f(x,y)| \le C\|f\|_{\widetilde{\mathcal{H}}}$, which ensures the boundedness of the evaluation functional of $\widetilde{\mathcal{H}}$. Let $\lambda_{\max}(\cdot)$ be the maximum eigenvalue of the matrix argument. From (15), we have

$$\begin{aligned}
\|f\|^2_{\widetilde{\mathcal{H}}} =& \langle f,f\rangle_{\mathcal{H}} + \frac{\gamma_I}{\gamma_A} F' L F \\
\le& \langle f,f\rangle_{\mathcal{H}} + \frac{\gamma_I}{\gamma_A}\lambda_{\max}(L)\|F\|^2 \ (\text{as } \max_x \tfrac{x'Ax}{\|x\|^2} = \lambda_{\max}(A)) \\
\le& \|f\|^2_{\mathcal{H}} + \frac{\gamma_I}{\gamma_A}\lambda_{\max}(L)(n|\mathcal{Y}|)C^2\|f\|^2_{\mathcal{H}} \\
=& \left(1 + \frac{\gamma_I}{\gamma_A}\lambda_{\max}(L)(n|\mathcal{Y}|)C^2\right)\|f\|^2_{\mathcal{H}}.
\end{aligned}$$

Moreover, by (15) we have $\|f\|_{\mathcal{H}} \le \|f\|_{\widetilde{\mathcal{H}}}$, then $\|f\|_{\mathcal{H}} \le \|f\|_{\widetilde{\mathcal{H}}} \le \bar{C}\|f\|_{\mathcal{H}}$. Therefore, the Cauchy sequence in the modified norm is also Cauchy in the original norm. This ensures that $\widetilde{\mathcal{H}}$ is also complete and thus $\widetilde{\mathcal{H}}$ is an RKHS. ■

Denote the kernel of $\widetilde{\mathcal{H}}$ by $\tilde{k}$. The following proposition shows that the kernel function $\tilde{k}$ is deformed from the original kernel k by the manifold.

Proposition 3: The kernel evaluation of $\tilde{k}$ on any $(\bar{x},\bar{y})$ is given by

$$\begin{aligned}
\tilde{k}((x,y),(\bar{x},\bar{y})) =& k((x,y),(\bar{x},\bar{y})) \\
& - K'_{\bar{x},\bar{y}}\left(\frac{\gamma_A}{\gamma_I} I_{n|\mathcal{Y}|} + LK\right)^{-1} L K_{x,y},
\end{aligned} \tag{16}$$

where $K_{x,y} = [k((x,y),(x_i,\bar{y}))]_{i=1,\ldots,n;\bar{y}\in\mathcal{Y}}$.

Proof: In the following, we use $k_{x,y}(\cdot,\cdot)$ as a shorthand for $k((x,y),(\cdot,\cdot))$. Decompose $\mathcal{H}$ as $\mathcal{H} = \mathcal{P}\oplus\mathcal{P}^{\perp}$, where $\mathcal{P}^{\perp}$ is the subspace in $\mathcal{H}$ orthogonal to

$$\mathcal{P} = \text{span}\{k_{x_i,y}(\cdot,\cdot) \mid i=1,\ldots,n;\ y\in\mathcal{Y}\}.$$

Suppose that $f\in\mathcal{P}^{\perp}$. Then $f(x_i,y) = \langle f, k_{x_i,y}(\cdot,\cdot)\rangle_{\mathcal{H}} = 0$, for $i=1,\ldots,n, y\in\mathcal{Y}$. Hence, $F=0$ in (15) and $\langle f,g\rangle_{\widetilde{\mathcal{H}}} = \langle f,g\rangle_{\mathcal{H}}\ \forall g\in\mathcal{H}$. In particular, let $g = \tilde{k}_{x,y}(\cdot,\cdot)$. We then have

$$\langle f, k_{x,y}(\cdot,\cdot)\rangle_{\mathcal{H}} = f(x,y) = \langle f,\tilde{k}_{x,y}(\cdot,\cdot)\rangle_{\widetilde{\mathcal{H}}} = \langle f,\tilde{k}_{x,y}(\cdot,\cdot)\rangle_{\mathcal{H}},$$

$\forall x\in\mathcal{X}, y\in\mathcal{Y}$. This implies $\langle f, \tilde{k}_{x,y}(\cdot,\cdot) - k_{x,y}(\cdot,\cdot)\rangle_{\mathcal{H}} = 0$, which means $\tilde{k}_{x,y}(\cdot,\cdot) - k_{x,y}(\cdot,\cdot) \in \mathcal{P}$. In other words, there exist coefficients $\beta_{iy'}(x,y)$ such that

$$\tilde{k}_{x,y}(\cdot,\cdot) = k_{x,y}(\cdot,\cdot) + \sum_{i=1}^{n}\sum_{y'\in\mathcal{Y}} \beta_{iy'}(x,y) k_{x_i,y'}(\cdot,\cdot). \tag{17}$$

Let $K_{x_j,\bar{y}} = [k_{x_j,\bar{y}}(x_i,y)]_{i=1,\ldots,n;y\in\mathcal{Y}}$, and $\beta(x,y) = [\beta_{iy}]_{i=1,\ldots,n;y\in\mathcal{Y}}$. Then,

$$\begin{aligned}
k_{x,y}(x_j,\bar{y}) &= \langle k_{x_j,\bar{y}}, \tilde{k}_{x,y}\rangle_{\widetilde{\mathcal{H}}} \\
&= \langle k_{x_j,\bar{y}}, k_{x,y} + \sum_{i=1}^{n}\sum_{y'\in\mathcal{Y}} \beta_{iy'}(x,y) k_{x_i,y'}\rangle_{\widetilde{\mathcal{H}}} \\
&= \langle k_{x_j,\bar{y}}, k_{x,y} + \sum_{i=1}^{n}\sum_{y'\in\mathcal{Y}} \beta_{iy'}(x,y) k_{x_i,y'}\rangle_{\mathcal{H}} \\
&\quad + \frac{\gamma_I}{\gamma_A} K_{x_j,\bar{y}} L(K_{x,y} + K\beta(x,y)).
\end{aligned}$$

Evaluating at all $x_j\in\{x_1,\ldots,x_n\}$ and $\bar{y}\in\mathcal{Y}$, and on gathering all the equations together in matrix form, we have

$$\begin{aligned}
& K\beta(x,y) + \frac{\gamma_I}{\gamma_A} KL(K_{x,y} + K\beta(x,y)) = 0 \\
\Rightarrow\quad & \beta(x,y) = -\left(\frac{\gamma_A}{\gamma_I} I_{n|\mathcal{Y}|} + LK\right)^{-1} L K_{x,y}.
\end{aligned}$$

Substituting this back into (17), we can evaluate the kernel on any $(\bar{x},\bar{y})$ using (16). ■

As in [12], [14], this modified kernel $\tilde{k}$ embodies information from both the original kernel k and the Laplacian L of the joint data manifold. Moreover, as expected, kernel $\tilde{k}$ can be reduced to the deformed kernel in [14].

Recall that the above shows that semi-supervised learning (using both labeled and unlabeled data) with manifold regularization is equivalent to supervised learning (using only the labeled data) with a manifold-deformed kernel. Intuitively, a similar relationship should also hold between the semi-supervised learning problem in (9) and the supervised learning problem in (3). This will be proved in the following.

First, we define the matrix

$$\widetilde{K} = \left(K^{-1} + \frac{\gamma_I}{\gamma_A} L\right)^{-1} \tag{18}$$

which is in the objective function in (9). Note that each entry of $\widetilde{K}$ embodies information from both the original kernel k and the graph Laplacian L. Moreover, it is easy to see that $\widetilde{K}$ can be regarded as a valid kernel matrix.

Lemma 3: $\widetilde{K}$ is symmetric and positive semidefinite.

The following proposition shows that the semi-supervised learning problem in (9) can be equivalently seen as a supervised learning problem (3) with this $\widetilde{K}$.

Proposition 4: Problem (9) is equivalent to problem (3), with the J matrix in (3) defined using the sub-matrix of $\widetilde{K}$ corresponding to the labeled data only.

Proof: Define

$$\breve{\beta} = [\beta_{i1}, \ldots, \beta_{i,y_i-1}, -\sum_{y \neq y_i} \beta_{iy}, \beta_{i,y_i+1}, \ldots, \beta_{i|\mathcal{Y}|}]_{i=1,\ldots,\ell}.$$

After some lengthy but standard algebra, it can be shown that the J in Theorem 4 is equal to

$$J = (\breve{I}_\ell \otimes I_{|\mathcal{Y}|}) \operatorname{diag}(M_1, \cdots, M_\ell),$$

where

$$M_i = \begin{array}{c} \\ y_i \rightarrow \end{array} \overset{\overset{y_i\text{-th column deleted}}{\downarrow}}{\begin{pmatrix} -1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & -1 & & & & & & \\ 1 & \cdots & 1 & & & & 1 & \cdots & 1 \\ & & & & & & -1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & -1 \end{pmatrix}}.$$

Then, for the $\beta' J' \widetilde{K} J \beta$ term in the objective of (9), we have $\beta' J' \widetilde{K} J \beta = \breve{\beta}' [\widetilde{K}]_{\text{lb}} \breve{\beta}$, where $[\widetilde{K}]_{\text{lb}}$ is the sub-matrix of $\widetilde{K}$ composed of the first $\ell|\mathcal{Y}|$ columns and rows (i.e., the part corresponding to the labeled data). It can then be seen that (9) is equivalent to (3) by equating $\beta_{iy} = 2\gamma_A \alpha_{iy}$. ∎

Clearly, this kernel matrix $\widetilde{K}$ should also be consistent with the deformed kernel $\tilde{k}$ in (16), as is confirmed by the following proposition.

Proposition 5: On evaluating the deformed joint kernel $\tilde{k}$ on both the labeled and unlabeled data, the resultant kernel matrix is the $\widetilde{K}$ defined in (18).

Proof: By fixing x, y and evaluating $\tilde{k}_{x,y}(\bar{x}, \bar{y})$ on all $\bar{x} \in \{x_1, \ldots, x_n\}$ and $\bar{y} \in \mathcal{Y}$, we have

$$\begin{aligned} &[\tilde{k}_{x,y}(x_i, \bar{y})]'_{i=1,\ldots,n;y\in\mathcal{Y}} \\ &= [k_{x,y}(x_i, \bar{y})]'_{i=1,\ldots,n;y\in\mathcal{Y}} \\ &\quad -K \left(\frac{\gamma_A}{\gamma_I} I_{n|\mathcal{Y}|} + LK\right)^{-1} L[k_{x,y}(x_j, \bar{y})]'_{j=1,\ldots,n;y\in\mathcal{Y}}. \end{aligned}$$

Evaluating on $x \in \{x_1, \ldots, x_n\}$ and $y \in \mathcal{Y}$, we can write in matrix form as

$$\begin{aligned} \widetilde{K} &= K - K \left(\frac{\gamma_A}{\gamma_I} I_{n|\mathcal{Y}|} + LK\right)^{-1} L \frac{\gamma_A}{\gamma_I} I_{n|\mathcal{Y}|} \\ &= \gamma_A (\gamma_A K^{-1} + \gamma_I L)^{-1}. \end{aligned}$$

∎

B. Tensor Product Decomposition of the Joint Kernel and Laplacian

In general, there can be different ways of defining the joint kernel and the corresponding joint feature map φ. A popular construction is via the tensor product $\varphi(x, y) = \phi(x) \otimes \lambda(y)$, where ϕ and λ are feature maps defined on $\mathcal{X}$ and $\mathcal{Y}$, respectively [1]. It can be shown that the resultant joint kernel is simply a product of the corresponding input kernel $\kappa(x, \bar{x}) = \phi(x)'\phi(\bar{x})$ and output kernel $\Lambda(y, \bar{y}) = \lambda(y)'\lambda(\bar{y})$. Subsequently, the kernel matrix is then

$$K = K^{\mathcal{X}} \otimes K^{\mathcal{Y}},$$

where $K^{\mathcal{X}}, K^{\mathcal{Y}}$ are the input and output kernel matrices, respectively.

Similarly, the affinity matrix $[A_{iyj\bar{y}}]$ on the joint manifold can also be constructed as a tensor product of the affinity matrices of the input and output spaces. In other words, $A_{iyj\bar{y}} = A^{\mathcal{X}}_{ij} A^{\mathcal{Y}}_{y\bar{y}}$, where $A^{\mathcal{X}}_{ij}$ and $A^{\mathcal{Y}}_{y\bar{y}}$ are the affinities in the input and output, respectively. By defining $D^{\mathcal{X}} = \operatorname{diag}(A^{\mathcal{X}} 1_n)$ and $D^{\mathcal{Y}} = \operatorname{diag}(A^{\mathcal{Y}} 1_{|\mathcal{Y}|})$, the (joint) graph Laplacian matrix can be written as

$$L = D^{\mathcal{X}} \otimes D^{\mathcal{Y}} - A^{\mathcal{X}} \otimes A^{\mathcal{Y}},$$

where $L^{\mathcal{X}}$ and $L^{\mathcal{Y}}$ are the input and output Laplacian matrices, respectively.

C. Output Structure is Absent

In the special case when there is no structure on the outputs, the tensor-product joint kernel reduces to

$$k((x, y), (\bar{x}, \bar{y})) = \kappa(x, \bar{x}) \delta(y, \bar{y}) \tag{19}$$

and $K = K^{\mathcal{X}} \otimes I_{|\mathcal{Y}|}$. Similarly, $L = L^{\mathcal{X}} \otimes I_{|\mathcal{Y}|}$. In this case, the problem reduces to standard manifold regularization in the multiclass setting. As is expected, the structured manifold-deformed kernel in (16) reduces to the unstructured manifold-deformed kernel in [14]. This is confirmed by the following proposition.

Proposition 6: If there is no output structure, kernel (16) reduces to the unstructured manifold-deformed data-dependent kernel in [14].

Proof: Let $\kappa_x = [\kappa(x, x_1), \ldots, \kappa(x, x_n)]'$ and $\delta_y = [\delta(y, \bar{y})]_{\bar{y}\in\mathcal{Y}}$. The RHS of (16) becomes

$$\begin{aligned} &\kappa(x, \bar{x})\delta(y, \bar{y}) - (\kappa'_z \otimes \delta'_{\bar{y}}) \left(\frac{\gamma_A}{\gamma_I} I_n + L^{\mathcal{X}} K^{\mathcal{X}}\right)^{-1} \otimes \\ &\qquad I_{|\mathcal{Y}|} (L^{\mathcal{X}} \otimes I_{|\mathcal{Y}|})(\kappa_x \otimes \delta_y) \\ &= \kappa(x, \bar{x})\delta(y, \bar{y}) \\ &\quad - \left(\kappa'_z (\frac{\gamma_A}{\gamma_I} I_n + L^{\mathcal{X}} K^{\mathcal{X}})^{-1} L^{\mathcal{X}} \kappa_x\right) \otimes (\delta'_{\bar{y}} I_{|\mathcal{Y}|} \delta_y) \\ &= \left(\kappa(x, \bar{x}) - \kappa'_z (\frac{\gamma_A}{\gamma_I} I_n + L^{\mathcal{X}} K^{\mathcal{X}})^{-1} L^{\mathcal{X}} \kappa_x)\right) \delta(y, \bar{y}), \\ &= \breve{\kappa}(x, \bar{x})\delta(y, \bar{y}). \end{aligned} \tag{20}$$

Here $\breve{\kappa}(\cdot, \cdot)$ is the unstructured manifold-deformed kernel in [14]. ∎

Moreover, as the outputs are now independent of each other, the regularizer $\|f\|^2_{\widetilde{\mathcal{H}}}$ corresponding the manifold-deformed RKHS $\widetilde{\mathcal{H}}$ is just a summation of all the unstructured manifold-deformed regularizers in the individual y spaces.

Proposition 7: Let the RKHS corresponding to $\breve{\kappa}(\cdot, \cdot)$, the unstructured manifold-deformed kernel in [14], by $\breve{\mathcal{H}}$. If there is no output structure, $\|f\|^2_{\widetilde{\mathcal{H}}} = \sum_{y\in\mathcal{Y}} \|f(\cdot, y)\|^2_{\breve{\mathcal{H}}}$.

Proof: On using (14) and (20),

$$\begin{aligned}\langle f, f\rangle_{\widetilde{\mathcal{H}}} &= \sum_{i,j=1}^{n}\sum_{y,\bar{y}\in\mathcal{Y}} \alpha_{iy}\alpha_{j\bar{y}}\tilde{k}((x_i,y),(x_j,\bar{y})) \\ &= \sum_{i,j=1}^{n}\sum_{y,\bar{y}\in\mathcal{Y}} \alpha_{iy}\alpha_{j\bar{y}}\breve{\kappa}(x_i,x_j)\delta(y,\bar{y}) \\ &= \sum_{y\in\mathcal{Y}}\sum_{i,j=1}^{n} \alpha_{iy}\alpha_{jy}\breve{\kappa}(x_i,x_j) \\ &= \sum_{y\in\mathcal{Y}} \|f(\cdot,y)\|^2_{\breve{\mathcal{H}}}.\end{aligned}$$

■

By this proposition, it is worth noting that our model (6) will turn into the multiclass SVM [15] with the data-dependent kernel $\breve{\kappa}(\cdot,\cdot)$ for the input space. Hence, the proposed formulation naturally extends the multiclass manifold-regularized SVM to the case when the outputs have structure.

V. Experiments

In this section, we demonstrate the usefulness of manifold information on two popular data sets, teapot and newsgroups. The data manifold is approximated by a weighted graph, which is constructed by using the 8-nearest-neighbors of each pattern. The weight a_{ij} between two neighbors x_i and x_j is defined in a similar manner as in [16]:

$$a_{ij} = \exp\left(-\frac{d(x_i,x_j)^2}{\max\{h(x_i),h(x_j)\}}\right),$$

where $d(x_i,x_j)$ is the distance between x_i, x_j, and $h(x_i)$ is the 8-nearest-neighbor distance of x_i. Moreover, we fix $\gamma_A = 5\times10^{-6}$ and $\gamma_I = 5\times10^{-5}$ for all data sets.

In the experiments, we use the popularly-used tensor product kernel and Laplacian as described as Section IV-B. Moreover, for the data sets in Sections V-A and V-B, the structure is in the output but not in the input. Hence, unlike the data sets used in [12], the input cannot be further decomposed into sub-components. In this case, it is natural to define each (x,y) (for all possible y's) as a "part", the method in [12] then becomes a special case of our approach.

A. Teapot Data

The teapot data[1] contains 400 teapot images (each of size 76×101) rotated from $1^\circ - 360^\circ$ to form a ring (Figure 1(a)). In this experiment, we group the images into 10 clusters, each with 30 images. 100 images are removed from the data set to ensure that the clusters are well separated. Then, from each cluster, 20 images are randomly selected to form a partially labeled training set and the remaining 10 are used as unseen test data. The number of labeled examples from each cluster is varied from 1 to 5. We use the Gaussian kernel on the input. As for the output kernel matrix, we use the 10×10 matrix $\begin{bmatrix} 2 & 1 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & & & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 1 & 2 \end{bmatrix}$ as suggested for ring structures in [17]. The ring loss, which is defined as the distance between the true and predicted labels along the ring, is used on training. Moreover, to reduce statistical variability, the experiment is repeated 200 times.

Results on the remaining unlabeled training data and unseen test data are shown in Tables I and II, respectively. Besides using the zero-one loss and ring loss as performance measures, we also report the commonly used information retrieval metrics including precision P, recall R and $F1 = 2PR/(P+R)$. As can be seen, the use of manifold information significantly improves the performance (with 95% confidence according to the paired student t-test).

B. Newsgroups Data

The second experiment is performed on the popular 20-newsgroups data[2]. We select 7 of these newsgroups which have a clear hierarchical structure (Figure 1(b)). From each newsgroup, 100 examples are randomly selected to form a partially labeled training set and another 50 are used as unseen test data. The number of labeled examples from each newsgroup is varied as 1,2,4,8 and 16. We use the linear kernel on the input. As for the output kernel, we use the following feature map defined in [1]. Let Z be the set of nodes in the hierarchy, and let the hierarchy structure be represented by the partial order $\prec$, where $z \prec \bar{z}$ means that node z is a parent of node $\bar{z}$. A feature λ_z is then defined with every node z, as

$$\lambda_z(\bar{z}) = \begin{cases} 1 & z \prec \bar{z} \text{ or } z = \bar{z}, \\ 0 & \text{otherwise.} \end{cases}.$$

The tree loss, which is defined as half of the length of the shortest (undirected) path connecting z and $\bar{z}$ in the hierarchy, is used on training. Again, to reduce statistical variability, the experiment is repeated 200 times.

Results on the unlabeled training data and unseen test data are shown in Tables III and IV, respectively. Again, the use of manifold information significant boosts the performance.

VI. Conclusion

In this paper, we extend manifold regularization to structured outputs via the joint kernel. This allows a unified framework for both unstructured and structured data. Desirable properties of traditional manifold regularization, such as the equivalence between semi-supervised learning and supervised learning with a data-dependent, manifold-deformed kernel, are also shown. Experimental results on real-world data sets with ring-structured and hierarchical outputs demonstrate that the prediction performance on both the unlabeled training data and unseen test data can be significantly improved (across all the metrics) with the use of manifold information.

Acknowledgments

This research was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region under grant 615209.

[1] http://www.it.usyd.edu.au/∼lesong/cluhsic_datasets.html

[2] http://people.csail.mit.edu/jrennie/20Newsgroups/

TABLE I

PERFORMANCE ON THE UNLABELED TRAINING DATA OF THE TEAPOT DATA SET.

		0-1 loss	ring loss	precision	recall	F1
#labeled = 5	w/ manifold	**0.055**	**0.041**	**0.953**	**0.945**	**0.949**
	w/o manifold	0.111	0.092	0.908	0.889	0.898
#labeled = 4	w/ manifold	**0.076**	**0.058**	**0.935**	**0.924**	**0.930**
	w/o manifold	0.145	0.122	0.881	0.855	0.868
#labeled = 3	w/ manifold	**0.113**	**0.088**	**0.903**	**0.887**	**0.895**
	w/o manifold	0.200	0.166	0.837	0.800	0.818
#labeled = 2	w/ manifold	**0.171**	**0.135**	**0.853**	**0.829**	**0.841**
	w/o manifold	0.281	0.229	0.772	0.719	0.744
#labeled = 1	w/ manifold	**0.276**	**0.226**	**0.760**	**0.724**	**0.741**
	w/o manifold	0.385	0.312	0.684	0.615	0.647

TABLE II

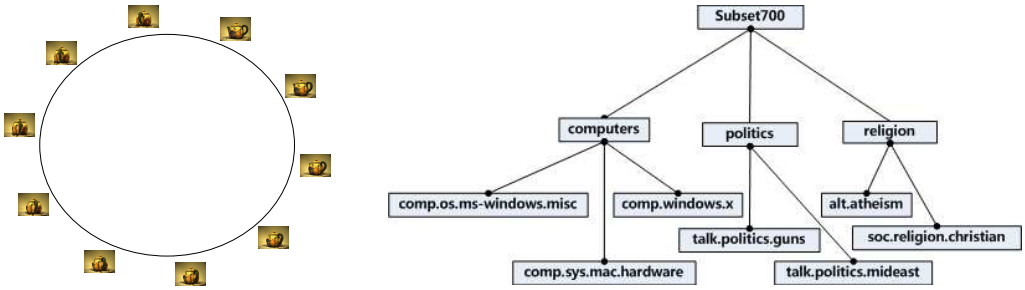
PERFORMANCE ON THE UNSEEN TEST DATA OF THE TEAPOT DATA SET.

		0-1 loss	ring loss	precision	recall	F1
#labeled = 5	w/ manifold	**0.057**	**0.043**	**0.950**	**0.943**	**0.947**
	w/o manifold	0.100	0.081	0.916	0.900	0.908
#labeled = 4	w/ manifold	**0.075**	**0.059**	**0.935**	**0.925**	**0.930**
	w/o manifold	0.127	0.104	0.893	0.873	0.883
#labeled = 3	w/ manifold	**0.111**	**0.088**	**0.904**	**0.889**	**0.897**
	w/o manifold	0.184	0.147	0.848	0.816	0.831
#labeled = 2	w/ manifold	**0.164**	**0.134**	**0.858**	**0.836**	**0.847**
	w/o manifold	0.262	0.209	0.784	0.738	0.760
#labeled = 1	w/ manifold	**0.266**	**0.221**	**0.766**	**0.734**	**0.750**
	w/o manifold	0.370	0.295	0.696	0.630	0.661

TABLE III

PERFORMANCE ON THE UNLABELED TRAINING DATA OF THE NEWSGROUPS DATA SET.

		0-1 loss	tree loss	precision	recall	F1
#labeled = 16	w/ manifold	**0.099**	**0.128**	**0.885**	**0.883**	**0.884**
	w/o manifold	0.120	0.145	0.859	0.861	0.860
#labeled = 8	w/ manifold	**0.132**	**0.166**	**0.845**	**0.848**	**0.846**
	w/o manifold	0.188	0.221	0.786	0.791	0.788
#labeled = 4	w/ manifold	**0.186**	**0.226**	**0.803**	**0.781**	**0.791**
	w/o manifold	0.278	0.329	0.701	0.693	0.696
#labeled = 2	w/ manifold	**0.247**	**0.307**	**0.769**	**0.703**	**0.733**
	w/o manifold	0.355	0.430	0.626	0.607	0.616
#labeled = 1	w/ manifold	**0.342**	**0.430**	**0.704**	**0.607**	**0.648**
	w/o manifold	0.419	0.516	0.563	0.539	0.549



(a) Teapot data. (b) Newsgroup data.

Fig. 1. Output structures of the data sets.

TABLE IV

PERFORMANCE ON THE UNSEEN TEST DATA OF THE NEWSGROUPS DATA SET.

		0-1 loss	tree loss	precision	recall	F1
#labeled = 16	w/ manifold	**0.110**	**0.145**	**0.879**	**0.873**	**0.876**
	w/o manifold	0.132	0.171	0.852	0.850	0.851
#labeled = 8	w/ manifold	**0.155**	**0.197**	**0.823**	**0.820**	**0.822**
	w/o manifold	0.200	0.250	0.776	0.774	0.774
#labeled = 4	w/ manifold	**0.220**	**0.278**	**0.762**	**0.738**	**0.749**
	w/o manifold	0.292	0.355	0.681	0.669	0.674
#labeled = 2	w/ manifold	**0.288**	**0.364**	**0.708**	**0.657**	**0.680**
	w/o manifold	0.364	0.446	0.612	0.589	0.599
#labeled = 1	w/ manifold	**0.375**	**0.477**	**0.628**	**0.565**	**0.593**
	w/o manifold	0.422	0.528	0.547	0.530	0.537

REFERENCES

[1] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, Dec. 2005.

[2] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.

[3] X. Zhu, "Semi-supervised learning literature survey," Department of Computer Sciences, University of Wisconsin - Madison, Tech. Rep. 1530, 2007.

[4] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Barbados, Jan. 2005.

[5] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proceedings of the Sixteenth International Conference on Machine Learning*, San Francisco, CA, USA, 1999, pp. 200–209.

[6] A. Zien, U. Brefeld, and T. Scheffer, "Transductive support vector machines for structured variables," in *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, Corvallis, Oregon, USA, Jun. 2007, pp. 1183–1190.

[7] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, Nov. 2006.

[8] D. Zhou, O. Bousquet, T. Lal, J. Weston, , and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[9] U. Brefeld and T. Scheffer, "Semi-supervised learning for structured output variables," in *Proceedings of the Twenty-Third International Conference on Machine Learning*, Pittsburgh, PA, USA, 2006, pp. 145–152.

[10] J. Lafferty, X. Zhu, and Y. Liu, "Kernel conditional random fields: Representation and clique selection," in *Proceedings of the Twenty-First International Conference on Machine Learning*, Banff, Alberta, Canada, Jul. 2004.

[11] C.-H. Lee, S. Wang, F. Jiao, D. Schuurmans, and D. Greiner, "Learning to model spatial dependency: Semi-supervised discriminative random fields," in *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press, 2007.

[12] Y. Altun, D. McAllester, and M. Belkin, "Maximum margin semi-supervised learning for structured variables," in *Advances in Neural Information Processing Systems 18*. Cambridge, MA: MIT Press, 2006.

[13] B. Taskar, "Learning structured prediction models: A large margin approach," Ph.D. dissertation, Stanford University, 2004.

[14] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: from transductive to semi-supervised learning," in *Proceedings of the Twenty-Second International Conference on Machine Learning*, Bonn, Germany, Aug. 2005, pp. 825–832.

[15] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2002.

[16] M. Hein and M. Maier, "Manifold denoising," in *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press, 2007.

[17] L. Song, A. Smola, A. Gretton, and K. Borgwardt, "A dependence maximization view of clustering," in *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, Corvallis, Oregon, USA, Jun. 2007.