Deformed Graph Laplacian for Semisupervised Learning

Chen Gong, Tongliang Liu, Dacheng Tao, *Fellow, IEEE*, Keren Fu, Enmei Tu, and Jie Yang

***Abstract*— Graph Laplacian has been widely exploited in traditional graph-based semisupervised learning (SSL) algorithms to regulate the labels of examples that vary smoothly on the graph. Although it achieves a promising performance in both transductive and inductive learning, it is not effective for handling ambiguous examples (shown in Fig. 1). This paper introduces deformed graph Laplacian (DGL) and presents label prediction via DGL (LPDGL) for SSL. The local smoothness term used in LPDGL, which regularizes examples and their neighbors locally, is able to improve classification accuracy by properly dealing with ambiguous examples. Theoretical studies reveal that LPDGL obtains the globally optimal decision function, and the free parameters are easy to tune. The generalization bound is derived based on the robustness analysis. Experiments on a variety of real-world data sets demonstrate that LPDGL achieves top-level performance on both transductive and inductive settings by comparing it with popular SSL algorithms, such as harmonic functions, AnchorGraph regularization, linear neighborhood propagation, Laplacian regularized least square, and Laplacian support vector machine.**

***Index Terms*— Deformed graph Laplacian (DGL), generalization bound, local smoothness regularizer, parametric sensitivity, semisupervised learning (SSL).**

I. Introduction

IN MANY real-world applications, the quantity of labeled examples is somewhat limited because of high monetary cost or unacceptable labeling time. For example, an interactive image segmentation requires the user to annotate a small number of seed points, because marking all foreground and background pixels manually is intractable. Besides, it often takes months of laboratory work for researchers to identify a single protein's 3-D structure, for instance. However, massive unlabeled examples are often readily available in the above applications. Therefore, semisupervised learning (SSL) was developed to deal with situations in which the labeled examples are scarce but the unlabeled examples are more than adequate. SSL has been intensively investigated in recent decades because of its solid theoretical base and enormous practical value [1]– [4].

Manuscript received July 20, 2013; revised May 13, 2014; accepted November 23, 2014. Date of publication January 15, 2015; date of current version September 16, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61273258, in part by the 973 Program of China under Grant 2015CB856004, and in part by the Australian Research Council under Projects FT-130101457, DP-140102164, and LP-140100569.

C. Gong is with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Centre for Quantum Computation & Information Systems, Faculty of Engineering and IT, University of Technology, Sydney, 235 Jones Street, Ultimo, NSW 2007, Australia (e-mail: goodgongchen@sjtu.edu.cn).

T. Liu and D. Tao are with the Centre for Quantum Computation & Information Systems, Faculty of Engineering and IT, University of Technology, Sydney, 235 Jones Street, Ultimo, NSW 2007, Australia (e-mail: tliang.liu@gmail.com; dacheng.tao@uts.edu.au).

K. Fu, E. Tu, and J. Yang are with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: fkrsuper@sjtu.edu.cn; tuen@sjtu.edu.cn; jieyang@sjtu.edu.cn).

This paper contains supplementary material available online at http://ieeexplore.ieee.org (File size: 1 MB).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2014.2376936

A. Semisupervised Learning

We use the notations $\mathcal{X}$ and $\mathcal{Y}$ to denote the example space and label space, respectively. Given a set of examples $\Psi = \{\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d, i = 1, 2, \ldots, n, n = l + u \text{ with } l \ll u\}$, in which the first l elements are examples with the labels $\{y_i\}_{i=1}^{l} \in \mathcal{Y} \in \{1, -1\}$, and the rest are u unlabeled examples. We use $\mathcal{L} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$ to denote the labeled set drawn from the joint distribution P defined on $\mathcal{X} \times \mathcal{Y}$, and $\mathcal{U} = \{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \ldots, \mathbf{x}_{l+u}\}$ to represent the unlabeled set drawn from the unknown marginal distribution $P_{\mathcal{X}}$ of P.

SSL can be either transductive or inductive [5]. A transductive algorithm aims to find the labels $y_{l+1}, y_{l+2}, \ldots, y_{l+u}$ of every unlabeled examples $\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \ldots, \mathbf{x}_{l+u}$ in $\mathcal{U}$ based on Ψ. In contrast, an inductive algorithm takes Ψ as the training set to train a suitable $f : \mathcal{X} \to \mathcal{Y}$, which is able to predict the label $f(\mathbf{x}_t) \in \mathcal{Y} \in \mathbb{R}$ of an unseen test example $\mathbf{x}_t \in \mathcal{X} \in \mathbb{R}^d$.

The main difference between SSL and traditional supervised learning is that SSL utilizes massive unlabeled examples to enhance classification performance. However, it is worth pointing out that the large quantity of unlabeled examples should be exploited under the correct assumption; otherwise, they will probably damage performance significantly. Two assumptions are commonly adopted for SSL, i.e., cluster assumption and manifold assumption. In cluster assumption, the probability distribution P is such that points in the same cluster are likely to have the same label [6]. It supposes that the classes in the example space are well separated, and the decision boundary will fall into a low-density region. Manifold assumption is also called smoothness assumption, which means that if $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ are close in the intrinsic geometry of the marginal distribution $P_{\mathcal{X}}$, then their labels y_1 and y_2 are similar. In other words, the data distribution is supposed to follow a manifold structure, along which the labels of examples should vary smoothly. It has been widely observed that a smoother solution usually leads to higher classification accuracy under the manifold assumption [5], [6].

Most algorithms based on the cluster assumption are usually variants of traditional support vector machines (SVMs).

Unlike supervised SVM, it is difficult to judge whether an unlabeled example in a semisupervised case is on the right or wrong side of the decision boundary, so hat loss is developed to replace the hinge loss commonly adopted by SVM. Semisupervised SVMs (S3VM) [7] and structural regularized SVM [8] are representatives of this learning assumption. A manifold assumption-based algorithm often establishes a graph to describe the manifold structure and uses the graph Laplacian to approximate the Laplace–Beltrami operator defined on the manifold. A smoothness term, which requires the two examples connected by a strong edge to obtain similar labels, is designed by adopting this graph Laplacian. Harmonic functions (HF) [1], linear neighborhood propagation (LNP) [2], Laplacian SVMs (LapSVM), and Laplacian regularized least squares (LapRLS) [9] belong to this assumption. According to the best of our knowledge, currently no theoretical approach that enables a decision about when to use cluster assumption-based methods or manifold assumption-based methods. The choice of SSL algorithms should fit the practical data distribution.

B. Motivation and Contribution

This paper aims to develop a graph-based SSL algorithm under the manifold assumption, which assumes that there exists a $\mathbb{C}^\infty$ smooth manifold $\mathcal{M}$ without boundary and with an infinitely differentiable embedding in the ambient example space $\mathcal{X}$. We aim to use the limited number of labeled examples $\{\mathbf{x}_i\}_{i=1}^{l} \in \mathbb{R}^d$ and the abundant unlabeled examples $\{\mathbf{x}_i\}_{i=l+1}^{l+u} \in \mathbb{R}^d$ to approximate the embedded manifold. This discovered manifold carries critical information for the distribution of the data set, which can be utilized to accurately classify the unlabeled examples.

As mentioned above, graph-based methods usually introduce a smoothness term to penalize the variation of labels along the manifold. To design such a smoothness term, existing methods usually adopt a standard graph Laplacian to constrain the labels of every pair of examples according to their similarities. The smoothness term defined by the standard graph Laplacian in this case is called a pairwise smoothness term. Unlike these traditional methods, we use the deformed graph Laplacian (DGL) [10] to define a novel smoothness term, and propose an algorithm called LPDGL. Compared with other popular SSL methods, LPDGL has the following three advantages because of the DGL.

1) A novel local smoothness term is introduced naturally, which is critical for our SSL model to better deal with ambiguous examples.
2) LPDGL is able to achieve higher classification accuracy than some state-of-the-art methods for both transductive and inductive settings.
3) LPDGL can be regarded as a unified framework of many popular SSL algorithms.

The local smoothness term mentioned in 1) considers the label smoothness of examples with their neighbors as a whole, and heavily regularizes the example that corresponds to a low degree. This is because an example that has weak edges with its neighbors often confuses the classifier significantly. Such examples can be outliers, or points that are located very close to the decision boundary. These ambiguous examples cannot be reliably classified because there is very little information provided by other examples. Similar idea can be found in [11], which uses the informative examples in dense regions to conduct active learning. The incorrect classification of ambiguous examples is likely to bring about disastrous results. Taking the *DoubleLine* data set, for example (Fig. 1), the red, blue, and black circles in Fig. 1(a) represent positive examples, negative examples, and unlabeled examples, respectively. The examples with y-coordinate 3 form the negative class and the points with y-coordinate 0 correspond to the positive class. The point at (6, 1.5) lies exactly in the middle of the two classes ($d_1 = d_2$), and can be attributed to an arbitrary class. We call this point as bridge point because it will probably serve as a bridge for the mutual transmission of positive and negative labels. In Fig. 1(b), which does not incorporate the local smoothness term, the positive label is mistakenly propagated to the negative class through the bridge point. This is because the labeled positive example [the red circle in Fig. 1(a)] is closer to the bridge point than the labeled negative example (blue circle), so it imposes more effects on the bridge point. As a consequence, the label of the bridge point is 0.1737 [Fig. 1(b)], which strongly influences the point at (6, 3), and leads to the incorrect classification of more than half of the negative examples. By comparison, Fig. 1(c) shows that the proposed LPDGL equipped with the local smoothness constraint successfully prohibits the label information from passing through it, and achieves a reasonable result. We observe that the label of bridge point is suppressed to a very small number (0.0352), significantly weakening the strength of the positive label propagating to the negative points.

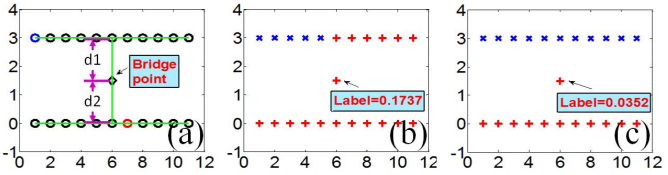


Fig. 1. Local smoothness constraint on *DoubleLine* data set. (a) k-NN graph with $k = 2$ is built and the edges are shown as green lines. (b) Result without incorporating the local smoothness. (c) Result produced by the proposed LPDGL. Labels of bridge point under two different simulations are highlighted in (b) and (c), respectively.

LPDGL is formulated as a regularization framework, through which the globally optimal solution is obtained. LPDGL deals with the transductive situations in Euclidean space, and handles the inductive tasks in reproducing kernel Hilbert space (RKHS). Theoretical analyses show that LPDGL is very robust to the choice of training examples, and the probability of the generalization risk being larger than any positive constant is bounded. Therefore, LPDGL performs accurately and reliably. Moreover, the parametric sensitivity is investigated based on the stability theory of solution of equations, from which we find that the classification performance is very robust to a wide choice of parameters. Therefore, the parameters in LPDGL are easy to tune.

LPDGL is demonstrated to be effective in many tough real-world applications, such as handwritten digit recognition,

unconstrained face recognition, and detection of violent behaviors. Therefore, the proposed algorithm has high practical value.

II. Transduction in Euclidean Space

Given a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ where $\mathcal{V}$ is the vertex set composed of the elements in Ψ, and $\mathcal{E}$ is the edge set recording the relationship among all the vertices. $\mathbf{W}_{n\times n}$ is the adjacency matrix of graph $\mathcal{G}$, in which the element ω_{ij} encodes the similarity between vertices i and j. The degree of the ith vertex is defined by $d_{ii} = \sum_{j=1}^{n} \omega_{ij}$, and $\mathbf{D}$ is a diagonal matrix with $(\mathbf{D})_{ii} = d_{ii}$ for $1 \le i \le n$. Therefore, the volume of graph $\mathcal{G}$ can be further formulated as $\upsilon = \sum_{i=1}^{n} d_{ii}$.

The existing SSL algorithms [1], [9], [12] usually adopt the traditional graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$, to model the smoothness relationship between examples. In particular, if we use the vector $\mathbf{f} = (f_1, f_2, \ldots, f_n)^T$ to record the determined soft labels of all the examples $\{\mathbf{x}_i\}_{i=1}^{n}$ in Ψ, then the smoothness term is formulated as

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij}(f_i - f_j)^2 = \mathbf{f}^T\mathbf{L}\mathbf{f}. \tag{1}$$

However, the pairwise smoothness (1) cannot effectively handle the ambiguous bridge point, as shown in Fig. 1, so we proposes a novel smoothness regularizer defined as

$$\Omega(\mathbf{f}) = \beta\mathbf{f}^T\mathbf{L}\mathbf{f} + \gamma\,\mathbf{f}^T(\mathbf{I} - \mathbf{D}/\upsilon)\mathbf{f} \tag{2}$$

in which β and γ are nonnegative parameters balancing the weights of the above two terms. The first term $\mathbf{f}^T\mathbf{L}\mathbf{f}$ is the traditional pairwise smoothness defined by (1). It evaluates the smoothness between pairs of $\mathbf{x}_i$ and $\mathbf{x}_j$ over the entire data set. The second term is the local smoothness term mentioned in Section I-B, which can be reformulated as

$$\mathbf{f}^T(\mathbf{I} - \mathbf{D}/\upsilon)\mathbf{f} = \sum_{i=1}^{n}(1 - d_{ii}/\upsilon)f_i^2. \tag{3}$$

On a k-NN graph $\mathcal{G}$, d_{ii} records the connective strength among $\mathbf{x}_i$ and its neighbors, so minimizing (3) enforces the example with large d_{ii} to obtain a confident soft label f_i, whereas the example with low degree d_{ii} to receive a relatively weak label.

Actually, a DGL formulated as $\hat{\mathbf{L}} = \mathbf{I} - \kappa\mathbf{W} - \kappa^2(\mathbf{I} - \mathbf{D})$ has been shown in [10], where κ is a free parameter and $\mathbf{I}$ is an $n \times n$ identity matrix. This DGL is an instance of a more general theory of deformed differential operators developed in mathematical physics [13]. The deformation technique was initially proposed for the dilation group, and was applied to many situations afterward, such as Schrödinger operation theory, quantum field theory, and plasma stability theory. Note that the deformed Laplacian $\hat{\mathbf{L}}$ will degenerate to the standard graph Laplacian $\mathbf{L}$ if κ is set to 1. Next, we will shed light upon that the proposed smoothness term (2) is related to $\hat{\mathbf{L}}$. By denoting $\tilde{\mathbf{L}} = \beta\mathbf{L} + \gamma(\mathbf{I} - \mathbf{D}/\upsilon)$, (2) can be expressed as $\Omega(\mathbf{f}) = \mathbf{f}^T\tilde{\mathbf{L}}\mathbf{f}$ and $\tilde{\mathbf{L}}$ here plays an equivalent role as $\mathbf{L}$ in (1). Considering that $\mathbf{L} = \mathbf{D} - \mathbf{W}$, we have

$$\begin{aligned}\tilde{\mathbf{L}} &= \gamma\mathbf{I} - \beta\mathbf{W} + (\beta - \gamma/\upsilon)\mathbf{D} \\ &= (\gamma + \beta - \gamma/\upsilon)\left[\mathbf{I} - \frac{\beta\upsilon}{\gamma\upsilon + \beta\upsilon - \gamma}\mathbf{W} - \frac{\beta\upsilon - \gamma}{\gamma\upsilon + \beta\upsilon - \gamma}(\mathbf{I} - \mathbf{D})\right]\end{aligned} \tag{4}$$

which equals to $\hat{\mathbf{L}}$ when $\gamma/\beta = \upsilon(\upsilon - 2)/\upsilon - 1$, and followed by the division by the coefficient $\gamma + \beta - \gamma/\upsilon$.

Based on the novel smoothness term, we derive the transductive model of LPDGL in the Euclidean space. Suppose $\mathbf{y} = (y_1, y_2, \ldots y_n)^T$ is a vector indicating the initial states of all examples, in which $y_i = 1, -1, 0$ when $\mathbf{x}_i$ is a positive example, negative example, and unlabeled example, respectively. Moreover, we define a diagonal matrix $\mathbf{J}_{n\times n}$ with the ith $(1 \le i \le n)$ diagonal element 1 if $\mathbf{x}_i$ is labeled and 0 otherwise, then the regularization framework of transductive LPDGL is

$$\min_{\mathbf{f}} Q(\mathbf{f}) = \frac{1}{2}\left[\beta\mathbf{f}^T\mathbf{L}\mathbf{f} + \gamma\,\mathbf{f}^T(\mathbf{I} - \mathbf{D}/\upsilon)\mathbf{f} + \|\mathbf{J}(\mathbf{f} - \mathbf{y})\|_2^2\right]. \tag{5}$$

The first term in the bracket of (5) is the pairwise smoothness term, which indicates that if two examples $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ distribute nearby in the example space $\mathcal{X}$, then their labels y_1 and y_2 should be also very similar in the label space $\mathcal{Y}$. Compared to the first term that simply evaluates the smoothness between two examples simultaneously, the second local smoothness term, which has been introduced above, considers the smoothness of examples and their k neighbors in a local region. As already revealed by Fig. 1(c), this manipulation makes the bridge point obtain a less reliable soft label, which effectively prevents the mutual transmission of labels belonging to different classes. The third term is a fidelity function, which guarantees that the labels of initially labeled examples $\{\mathbf{x}_i\}_{i=1}^{l}$ remain consistent with its initial conditions $\{y_i\}_{i=1}^{l}$ after transduction. To find the minimizer of (5), we set the derivative of $Q(\mathbf{f})$ with respect to $\mathbf{f}$ to $\mathbf{0}$, and obtain

$$\beta\mathbf{L}\mathbf{f} + \gamma(\mathbf{I} - \mathbf{D}/\upsilon)\mathbf{f} + \mathbf{J}\mathbf{f} - \mathbf{J}\mathbf{y} = \mathbf{0}. \tag{6}$$

Therefore, the optimal $\mathbf{f}$ is expressed as

$$\mathbf{f} = [\mathbf{J} + \beta\mathbf{L} + \gamma(\mathbf{I} - \mathbf{D}/\upsilon)]^{-1}\mathbf{y}. \tag{7}$$

Based on (7), the label of $\mathbf{x}_i \in \mathcal{U}$ is further determined as 1 if $f_i > 0$, and -1 otherwise.

Next, we investigate the parametric sensitivity of the proposed LPDGL. Parametric sensitivity evaluates the impact of a parameter on the final output of the model. If the output remains substantially unchanged with the wide range of a parameter, we say that the output is insensitive to the choice of this parameter.

In the proposed LPDGL, β and γ are two critical parameters to be tuned. This section aims to verify that the classification results of LPDGL are insensitive to the variation of either of them. The theoretical results provided here will be empirically demonstrated in Section V-C. Let $\mathbf{B} = \mathbf{J} + \beta\mathbf{L} + \gamma(\mathbf{I} - \mathbf{D}/\upsilon)$, then the impacts of β and γ on $\mathbf{f}$ are studied by investigating the equations $\mathbf{B}\mathbf{f} = \mathbf{y}$ about how $\mathbf{f}$ is affected when the coefficient matrix $\mathbf{B}$ is slightly disturbed. Before discussing

$$\frac{\|\delta\mathbf{B}\|}{\|\mathbf{B}\|}=\frac{\delta\gamma\,\|\mathbf{I}-\mathbf{D}/\upsilon\|}{\|\mathbf{J}+\beta\mathbf{L}+\gamma\,(\mathbf{I}-\mathbf{D}/\upsilon)\|}=\frac{\delta\gamma\sqrt{\sum_{i=1}^{n}(1-d_{ii}/\upsilon)^2}}{\sqrt{\beta^2\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\omega_{ij}^2+\sum_{i=1}^{n}[\beta d_{ii}+\gamma\,(1-d_{ii}/\upsilon)]^2+\xi}}$$
$$=\frac{\delta\gamma\sqrt{\sum_{i=1}^{n}(1-d_{ii}/\upsilon)^2}}{\sqrt{\beta^2\sum_i\sum_j\omega_{ij}^2+\beta\sum_i d_{ii}[(\beta-2\gamma/\upsilon)d_{ii}+2\gamma]+\xi+\gamma^2\sum_i(1-d_{ii}/\upsilon)^2}} \tag{9}$$

$$\frac{\|\delta\mathbf{B}\|}{\|\mathbf{B}\|}=\frac{\delta\beta\,\|\mathbf{L}\|}{\|\mathbf{J}+\beta\mathbf{L}+\gamma\,(\mathbf{I}-\mathbf{D}/\upsilon)\|}=\frac{\delta\beta\sqrt{\sum_{i=1}^{n}d_{ii}^2+\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\omega_{ij}^2}}{\sqrt{\beta^2\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\omega_{ij}^2+\sum_{i=1}^{n}\left[\beta d_{ii}+\gamma\,(1-d_{ii}/\upsilon)\right]^2+\xi}}$$
$$=\frac{\delta\beta\sqrt{\sum_{i=1}^{n}d_{ii}^2+\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\omega_{ij}^2}}{\sqrt{\gamma^2\sum_i\left(1-\frac{d_{ii}}{\upsilon}\right)^2+\gamma\sum_i\left\{\left(1-\frac{d_{ii}}{\upsilon}\right)\left[\left(2\beta-\frac{\gamma}{\upsilon}\right)d_{ii}+\gamma\right]\right\}+\xi+\beta^2\left(\sum_i d_{ii}^2+\sum_i\sum_j\omega_{ij}^2\right)}} \tag{10}$$

the parametric sensitivity of β and γ, we first provide a useful lemma.

Lemma 1 [14]: Given a set of linear equations $\mathbf{Bf}=\mathbf{y}$, where $\mathbf{B}\in C^{n\times n}$ is the coefficient matrix and $\mathbf{f}$ is the solution. Suppose $\mathbf{y}$ at the right-hand side of equations is accurate and $\mathbf{B}$ is slightly disturbed by $\delta\mathbf{B}$, then the deviation $\delta\mathbf{f}$ from the accurate $\mathbf{f}$ satisfies

$$\frac{\|\delta\mathbf{f}\|}{\|\mathbf{f}\|}\leq\frac{\mathrm{Cond}(\mathbf{B})(\|\delta\mathbf{B}\|/\|\mathbf{B}\|)}{1-\mathrm{Cond}(\mathbf{B})(\|\delta\mathbf{B}\|/\|\mathbf{B}\|)} \tag{8}$$

where $\|\cdot\|$ denotes the Euclidean norm and $\mathrm{Cond}(\mathbf{B})=\|\mathbf{B}\|\|\mathbf{B}^{-1}\|$ is the associated condition number.

A. Sensitivity of γ

Suppose a small deviation $\delta\gamma$ is added to the parameter γ, then $\delta\mathbf{B}$ in (8) is $\delta\mathbf{B}=\delta\gamma\,(\mathbf{I}-\mathbf{D}/\upsilon)$, which leads to the departure $\delta\mathbf{f}$ from the accurate solution $\mathbf{f}$. Therefore, we have (9), as shown at the top of this page, where $\xi=2\sum_{i=1}^{l}[\beta d_{ii}+\gamma\,(1-d_{ii}/\upsilon)]+l>0$. Note that the numerator in (9) is the same as the last term of denominator except the coefficient, and it is a small number compared with the denominator if γ is slightly disturbed, so $\|\delta\mathbf{B}\|/\|\mathbf{B}\|$ in (9) is very close to 0. Moreover, it is clear that $\mathbf{B}$ is a positive definite matrix, which is invertible, so $\mathrm{Cond}(\mathbf{B})$ will not be overly large. Therefore, the value of the right-hand side of (8) is small, which suggests that the performance of LPDGL is not sensitive to the choice of γ.

B. Sensitivity of β

Suppose a small bias $\delta\beta$ is added to β, then $\delta\mathbf{B}$ in (8) is $\delta\mathbf{B}=\delta\beta\mathbf{L}$. Therefore, we compute the value of $\|\delta\mathbf{B}\|/\|\mathbf{B}\|$, and obtain (10), as shown at the top of this page.

Similar to (9), the numerator in (10) is very small compared with the denominator, so $\|\delta\mathbf{B}\|/\|\mathbf{B}\|$ is very close to 0. As a result, according to (8), we know that $\|\delta\mathbf{f}\|$ is negligible in the presence of $\|\mathbf{f}\|$, which indicates that the result of LPDGL is also very robust to the variation of β.

III. Induction in RKHS

Note that $\mathbf{f}=(f_1,f_2,\ldots,f_n)^T$ in (7) only encodes the soft labels of examples that are used to construct the graph $\mathcal{G}$ during the training phase, so it cannot predict the labels of test examples that are unseen in the training phase. Therefore, this section adapts the proposed LPDGL to inductive settings, which requires the decision function f trained on Ψ to perfectly handle the out-of-sample data, and the predicted label, for example, $\mathbf{x}$ is $f(\mathbf{x})\in\mathbb{R}$.

In this paper, we build the LPDGL model for prediction in the RKHS. An RKHS $\mathcal{H}_K$ is a Hilbert space $\mathcal{H}$ of functions on a set $\mathcal{X}$ with the property that for all $x\in\mathcal{X}$ and $f\in\mathcal{H}$, the point evaluations $f\rightarrow f(x)$ are continuous linear functionals [15]. The Moore–Aronszajn theorem [16] indicates that for every RKHS, there exists a unique positive definite kernel on $\mathcal{X}\times\mathcal{X}$. Therefore, by adopting the Riesz representation theorem, the unique reproducing kernel can always be constructed as $K(\cdot,\cdot):\mathcal{X}\times\mathcal{X}\rightarrow\mathbb{R}$, which has an important property that $\forall x_1,x_2\in\mathcal{X}$, $K(x_1,x_2)=\langle K(\cdot,x_1),K(\cdot,x_2)\rangle_{\mathcal{H}}$, from the point evaluation functional.

Remark 1: The linear counterpart of LPDGL in RKHS can be derived using the linear prediction function $f(\mathbf{x})=\omega^T\mathbf{x}$, and then substitute $\mathbf{f}=\mathbf{X}^T\omega$ into (5), where ω is the weight vector to be optimized and $\mathbf{X}=(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n)$ is the data matrix with each column representing an example. The supplementary material shows that LPDGL in RKHS includes the linear LPDGL as a special case, and in particular, LPDGL in RKHS degenerates to the linear LPDGL when the linear kernel $\mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^T\mathbf{x}_j$ is adopted.

Empirically, when data from different classes are linearly nonseparable, LPDGL in RKHS with a nonlinear kernel is superior to its linear counterpart; otherwise, linear LPDGL should be applied.

Suppose $K(\cdot,\cdot)$ is a Mercer kernel associated with RKHS, and the corresponding norm is $\|\cdot\|_{\mathcal{H}}$, then we have the following regularization framework of LPDGL defined in RKHS:

$$\min_{f\in\mathcal{H}_K} Q(f) = \frac{1}{2}\Big[\alpha\,\|f\|_{\mathcal{H}}^2 + \beta\mathbf{f}^T\mathbf{L}\mathbf{f} + \gamma\,\mathbf{f}^T(\mathbf{I}-\mathbf{D}/v)\mathbf{f} + \sum_{i=1}^{l}(f(\mathbf{x}_i)-y_i)^2\Big]. \tag{11}$$

In (11), the tradeoff among the four terms is captured by three nonnegative parameters α, β, and γ. Compared with (5) for transduction, (11) contains one more induction term $\|f\|_{\mathcal{H}}^2$ that controls the complexity of f. This term enhances the generalizability of LPDGL by effectively preventing the overfitting problem.

The extended representer theorem [9] states that the minimizer of (11) can be decomposed as an expansion of kernel functions over both labeled and unlabeled examples

$$f(\mathbf{x}) = \sum_{i=1}^{n} s_i K(\mathbf{x},\mathbf{x}_i). \tag{12}$$

Therefore, by plugging (12) into (11), we obtain a novel objective function with respect to $\mathbf{S} = (s_1,\ldots,s_n)^T$

$$\min_{\mathbf{S}\in\mathbb{R}^n} \widetilde{Q}(\mathbf{S}) = \frac{1}{2}\big[\alpha\mathbf{S}^T\mathbf{K}\mathbf{S} + \beta\mathbf{S}^T\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{S} + \gamma\,\mathbf{S}^T\mathbf{K}(\mathbf{I}-\mathbf{D}/v)\mathbf{K}\mathbf{S} + \|\mathbf{y}-\mathbf{J}\mathbf{K}\mathbf{S}\|^2\big] \tag{13}$$

where $\mathbf{K}$ is an $n\times n$ Gram matrix over all the training examples, with elements $\mathbf{K}_{ij} = K(\mathbf{x}_i,\mathbf{x}_j)$ for $1\le i,j\le n$. It can be easily proved that the objective function in (13) is convex, so we can find the globally optimal $\mathbf{S}$ by calculating the derivative of $\widetilde{Q}(\mathbf{S})$ to $\mathbf{S}$, and then setting the result to $\mathbf{0}$, which is expressed as

$$\mathbf{S} = [\alpha\mathbf{I} + \beta\mathbf{L}\mathbf{K} + \gamma\,(\mathbf{I}-\mathbf{D}/v)\mathbf{K} + \mathbf{J}\mathbf{K}]^{-1}\mathbf{y}. \tag{14}$$

Finally, we substitute (14) into (12), and obtain the function f for predicting the label of $\mathbf{x}$.

A. Robustness Analysis

Robustness is a desirable property for a learning algorithm, because it reflects the sensitivity of the algorithm to the disturbances of training data. Xu and Mannor [17] state that an algorithm is robust if its solution achieves similar performances on a test set and a training set that are close. Based on the notion introduced by [17], this section studies the robustness of LPDGL. The whole sample space is represented by $\mathcal{Z} = \mathcal{X}\times\mathcal{Y}$, where $\mathcal{X}$ is the input example space and $\mathcal{Y}$ is the output label space. Furthermore, we use $z_i = (\mathbf{x}_i, y_i)\in\mathcal{Z}$ to denote the example-label pair, where $\mathbf{x}_i\in\mathcal{X}$ and $y_i\in\mathcal{Y} = \{-1,1\}$. Therefore, the task of LPDGL is to learn a function $f:\mathcal{X}\to\mathcal{Y}$ that maps the elements in the input space $\mathcal{X}$ to the output space $\mathcal{Y}$.

Definition 1 (Covering Number [17]): For a metric space S_ρ with a metric ρ, where T, $\hat{T}\subset S_\rho$ are two sets in S_ρ, we say that $\hat{T}$ is an ε-cover of T, if $\forall t\in T$, $\exists\hat{t}\in\hat{T}$, such that $\rho(t,\hat{t})\le\varepsilon$. The ε-covering number of T is

$$N(\varepsilon,T,\rho) = \min\{|\hat{T}| : \hat{T}\text{ is an }\varepsilon-\text{cover of }T\}. \tag{15}$$

Definition 2 (Robustness [17]): Let Ψ, $L(\cdot)$ denote the training set and loss function of an algorithm $\mathcal{A}$, respectively, then $\mathcal{A}$ is $(\theta,\varepsilon(\Psi))$-robust if $\mathcal{Z}$ can be partitioned into θ disjoint sets, denoted as $\{C_i\}_{i=1}^{\theta}$, such that $\forall\mathbf{x}_1,\mathbf{x}_2\in\Psi$

$$z_1,z_2\in C_i \Rightarrow |L(\mathcal{A}_\Psi,z_1) - L(\mathcal{A}_\Psi,z_2)|\le\varepsilon(\Psi). \tag{16}$$

Based on Definitions 1 and 2, we have the following theorem.

Theorem 3: Let $\mathcal{X}$ denote the input space, and $\forall\mathbf{x}_i$, $\mathbf{x}_j\in\mathcal{X}$, $\|\mathbf{x}_i-\mathbf{x}_j\|\le\varepsilon$. A k-NN graph is built with the edge weights represented by Radial Basis Function (RBF) kernel $\omega_{ij} = \exp(-\|\mathbf{x}_i-\mathbf{x}_j\|^2/(2\sigma^2))$. Under $N(\varepsilon/2,\mathcal{X},\|\cdot\|_2)<\infty$, the proposed LPDGL is $(8l/\alpha)^{1/2}(1+(l/\alpha)^{1/2})(1-\exp((-\varepsilon^2)/(2\sigma^2)))^{1/2}$-robust.

Proof: Suppose $\mathbf{S}$ in (13) is set to $\mathbf{S}_0 = (0,\ldots,0)^T$, then we have $\widetilde{Q}(\mathbf{S}_0) = \|\mathbf{y}\|^2/2 = l/2$. Moreover, note that all the terms in the bracket in (13) are nonnegative, so we obtain $1/2\ \alpha\ \mathbf{S}^T\mathbf{K}\mathbf{S}\le Q(\mathbf{S})\le Q(\mathbf{S}_0) = l/2$, which reveals that

$$\mathbf{S}^T\mathbf{K}\mathbf{S}\le l/\alpha. \tag{17}$$

For binary classification, we can partition $\mathcal{Z}$ into $\theta = 2N(\varepsilon/2,\mathcal{X},\|\cdot\|_2)$ disjoint sets with a margin ε [17]. Therefore, according to Definition 1, we know that if z_1 and z_2 belong to the same set C_i $(1\le i\le\theta)$, then $\|\mathbf{x}_1-\mathbf{x}_2\|\le\varepsilon$ and $\|y_1-y_2\| = 0$ [17]. We also know that the loss function of LPDGL is

$$L(f) = (f(\mathbf{x})-y)^2 \tag{18}$$

so according to Definition 2 the difference between the losses of f on z_1 and z_2 is

$$|L(f,z_1) - L(f,z_2)| = \big|(y_1-f(\mathbf{x}_1))^2 - (y_2-f(\mathbf{x}_2))^2\big|. \tag{19}$$

By plugging (12) into (19), we obtain

$$\begin{aligned}
&|L(f,z_1) - L(f,z_2)|\\
&= \left|\left[y_1-\sum_{i=1}^{n}s_iK(\mathbf{x}_1,\mathbf{x}_i)\right]^2 - \left[y_2-\sum_{i=1}^{n}s_iK(\mathbf{x}_2,\mathbf{x}_i)\right]^2\right|\\
&\le \left|y_1+y_2-\sum_{i=1}^{n}s_i\,(K(\mathbf{x}_1,\mathbf{x}_i)+K(\mathbf{x}_2,\mathbf{x}_i))\right|\\
&\quad \left|y_1-y_2-\sum_{i=1}^{n}s_i\,(K(\mathbf{x}_1,\mathbf{x}_i)-K(\mathbf{x}_2,\mathbf{x}_i))\right|\\
&= |B_1|\,|B_2|
\end{aligned} \tag{20}$$

in which $B_1 = y_1+y_2-\sum_{i=1}^{n}s_i(K(\mathbf{x}_1,\mathbf{x}_i)+K(\mathbf{x}_2,\mathbf{x}_i))$ and $B_2 = y_1-y_2-\sum_{i=1}^{n}s_i(K(\mathbf{x}_1,\mathbf{x}_i)-K(\mathbf{x}_2,\mathbf{x}_i))$. In the

following derivations, we aim to find the upper bounds of $|B_1|$ and $|B_2|$, respectively. It is easy to show that

$$\begin{aligned}|B_1| &\le |y_1| + |y_2| + |f(\mathbf{x}_1) + f(\mathbf{x}_2)| \\ &\le 2 + 2\max_{\mathbf{x}\in\{\mathbf{x}_1,\mathbf{x}_2\}} \langle f, K(\mathbf{x},\cdot)\rangle_{\mathcal{H}} \\ &\le 2 + 2\max_{\mathbf{x}\in\{\mathbf{x}_1,\mathbf{x}_2\}} \|f\|_{\mathcal{H}}\sqrt{K(\mathbf{x},\cdot)} \\ &\le 2 + 2\max_{\mathbf{x}\in\{\mathbf{x}_1,\mathbf{x}_2\}} \left\|\sum_{i=1}^{n} s_i K(\mathbf{x}_i,\cdot)\right\|_{\mathcal{H}}\sqrt{K(\mathbf{x},\cdot)} \\ &\le 2 + 2\sqrt{\left\langle \sum_{i=1}^{n} s_i K(\mathbf{x}_i,\cdot), \sum_{i=1}^{n} s_j K(\mathbf{x}_j,\cdot)\right\rangle_{\mathcal{H}}} \\ &= 2 + 2\sqrt{\sum_{i,j=1}^{n} s_i s_j K(\mathbf{x}_i,\cdot)K(\mathbf{x}_j,\cdot)} \\ &= 2 + 2\sqrt{\sum_{i,j=1}^{n} s_i K(\mathbf{x}_i,\mathbf{x}_j)s_j} \\ &= 2 + 2\sqrt{\mathbf{S}^T\mathbf{K}\mathbf{S}} \le 2 + 2\sqrt{\frac{l}{\alpha}} \end{aligned} \tag{21}$$

in which the notation $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ denotes the inner product defined in $\mathcal{H}$. Note that in the derivation of (21), we employed the reproducing property of RKHS $f(\mathbf{x}) = \langle f, K(\mathbf{x},\cdot)\rangle_{\mathcal{H}}$ in the second line, the Cauchy–Schwarz inequality $\langle f, K(\mathbf{x},\cdot)\rangle_{\mathcal{H}} \le \|f\|_{\mathcal{H}}(K(\mathbf{x},\cdot))^{1/2}$ in the third line, and the results of (17) in the last line.

Moreover, since $\|f\|_{\mathcal{H}}^2 = \mathbf{S}^T\mathbf{K}\mathbf{S} \le (l/\alpha)$, we have $\|f\|_{\mathcal{H}} \le (l/\alpha)^{1/2}$. By further considering that $\|y_1 - y_2\| = 0$ and $\|\mathbf{x}_1 - \mathbf{x}_2\| \le \varepsilon$, we immediately obtain

$$\begin{aligned}|B_2| &= |f(\mathbf{x}_1) - f(\mathbf{x}_2)| \\ &= |\langle f, K(\mathbf{x}_1,\cdot) - K(\mathbf{x}_2,\cdot)\rangle_{\mathcal{H}}| \\ &\le \|f\|_{\mathcal{H}}\|K(\mathbf{x}_1,\cdot) - K(\mathbf{x}_2,\cdot)\|_{\mathcal{H}} \\ &= \|f\|_{\mathcal{H}}\sqrt{K(\mathbf{x}_1,\mathbf{x}_1) + K(\mathbf{x}_2,\mathbf{x}_2) - 2K(\mathbf{x}_1,\mathbf{x}_2)} \\ &\le \sqrt{l/\alpha}\sqrt{K(\mathbf{x}_1,\mathbf{x}_1) + K(\mathbf{x}_2,\mathbf{x}_2) - 2K(\mathbf{x}_1,\mathbf{x}_2)} \\ &\le \sqrt{l/\alpha}\sqrt{2 - 2\exp[-\|\mathbf{x}_1 - \mathbf{x}_2\|^2/(2\sigma^2)]} \\ &= \sqrt{l/\alpha}\sqrt{2 - 2\exp[-\varepsilon^2/(2\sigma^2)]}. \end{aligned} \tag{22}$$

Finally, we substitute (21) and (22) into (20), and have

$$|L(f,z_1) - L(f,z_2)| \le \sqrt{\frac{8l}{\alpha}}\left(1 + \sqrt{\frac{l}{\alpha}}\right)\sqrt{1 - \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right)} \tag{23}$$

which indicates that LPDGL is $(\theta, (8l/\alpha)^{1/2}(1 + (l/\alpha)^{1/2})(1 - \exp(-\varepsilon^2/2\sigma^2)^{1/2})$-robust. ∎

B. Generalization Risk

Based on the robustness analysis in Section III-A, we derive the generalization bound for LPDGL. The empirical error $L_{\text{emp}}(\mathcal{A}_\Psi)$ is the error of algorithm $\mathcal{A}$ on the training set Ψ. The generalization error $\tilde{L}(\cdot)$ is the expectation of error rate produced by f on the whole sample space $\mathcal{Z}$. Suppose all the examples are i.i.d, and are generated from an unknown distribution P, then the above two errors are defined by $\tilde{L}(\mathcal{A}_\Psi) = E_{\mathbf{x}\sim P}[L(\mathcal{A}_\Psi,\mathbf{x})]$ and $L_{\text{emp}}(\mathcal{A}_\Psi) = 1/n\sum_{\mathbf{x}_i\in\Psi} L(\mathcal{A}_\Psi,\mathbf{x}_i)$, respectively.

Theorem 4 (Generalization Bound [17]): If the training set Ψ consists of n i.i.d. samples, and the algorithm $\mathcal{A}$ is $(\theta,\varepsilon(\Psi))$-robust, then for any $\delta > 0$, with probability at least $1 - \delta$

$$|\tilde{L}(\mathcal{A}_\Psi) - L_{\text{emp}}(\mathcal{A}_\Psi)| \le \varepsilon(\Psi) + M\sqrt{\frac{2\theta\ln 2 + 2\ln(1/\delta)}{n}} \tag{24}$$

where M is the upper bound of loss function $L(\cdot,\cdot)$.

According to Theorem 4, the generalization bound of inductive LPDGL is provided in Theorem 5.

Theorem 5: Let $L(f,\Psi) = (f(\mathbf{x}) - y)^2$ be the loss function of LPDGL, than for any $\delta > 0$, with probability at least $1 - \delta$, the generalization error of LPDGL is

$$\begin{aligned}|\tilde{L}(A_\psi) - L_{\text{emp}}(A_\psi)| \le{}& \sqrt{\frac{8l}{\alpha}}\left(1 + \sqrt{\frac{l}{\alpha}}\right)\sqrt{1 - \exp\left(\frac{-\varepsilon^2}{2\sigma^2}\right)} \\ &+ 2\left(1 + \frac{l}{\alpha}\right)\sqrt{\frac{2\theta\ln 2 + 2\ln(1/\delta)}{n}}. \end{aligned} \tag{25}$$

Proof: To obtain the generalization bound of LPDGL, we need to compute $\varepsilon(\Psi)$, θ, and M that appear in (24). Note that $\varepsilon(\Psi)$ and θ have been already worked out in Section III-A, so our target is to find the upper bound M of the loss function $L(f,\Psi)$. Therefore, we compute

$$\begin{aligned}L(f,\psi) &= (y - f(\mathbf{x}))^2 = y^2 - 2yf(\mathbf{x}) + f^2(\mathbf{x}) \\ &\le 2y^2 + 2f^2(\mathbf{x}) \le 2 + 2l/\alpha. \end{aligned} \tag{26}$$

As a result, the upper bound of the adopted loss function is

$$M = 2 + 2l/\alpha. \tag{27}$$

Finally, by putting (23) and (27) into (24), we complete the proof. ∎

Theorem 5 reveals that the our LPDGL has a profound generalizability with convergence rate of order $\mathcal{O}(1/n)^{1/2}$, which means that the more training examples are available, the lower generalization bound of LPDGL we have, so LPDGL can predict the label of a test example reliably.

IV. Related Work

SSL has attracted considerable interest since it was developed. Various SSL algorithms have been proposed for different purposes and applications. As mentioned in Section I, existing SSL algorithms can be divided into a transductive approach or inductive approach.

Typical transductive methods include Tikhonov regularization [18], HF [1], local and global consistency (LGC) [12], minimum cut (MinCut) [19], local learning regularization (LLReg) [20], local and global regularization (LGReg) [21], path-based SSL (PBSSL) [22], transductive SVMs (S3VM) [7], safe SSL (S4VM) [23], AnchorGraph regularization (AGR) [24], graph transduction via alternating minimization (GTAM) [25], Laplacian embedded support vector regression [26], semisupervised

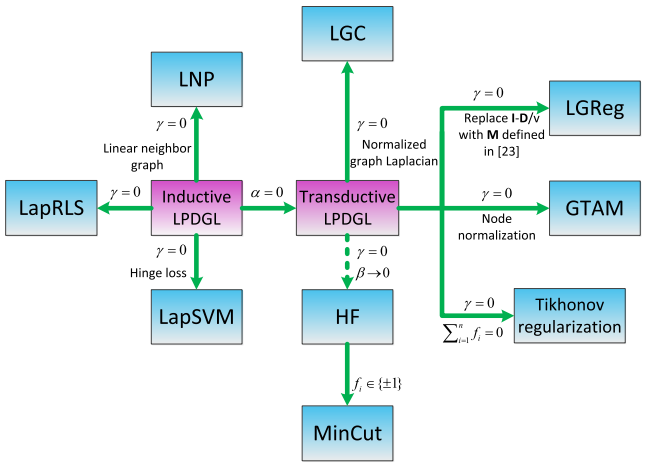


Fig. 2. Evolutionary process from LPDGL to other typical SSL methods. Dashed line: infinitely approach to. Note that our LPDGL is located in the central position and other algorithms are derived from LPDGL by satisfying the conditions alongside the arrows.

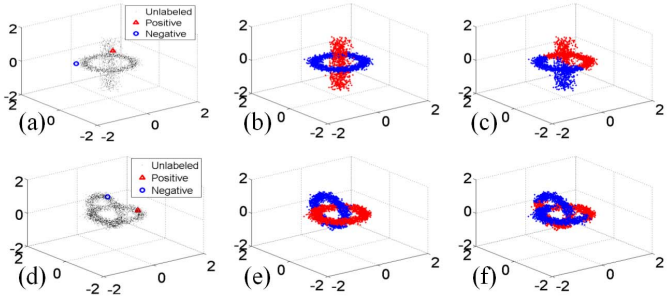


Fig. 3. Transduction on two 3-D data sets. (a) and (d) Initial states of *Cylinder&Ring* and *Knot*, respectively, in which the red triangle denotes a positive example and the blue circle represents a negative example. (b) and (e) Transduction results of developed LPDGL on these two data sets. (c) and (f) Results of LPDGL (linear).

classification based on class membership (SSCCM) [27], and safety-aware SSCCM (SA-SSCCM) [4]. Of these, S3VM, S4VM, SSCCM, and SA-SSCCM are based on the (modified) cluster assumption and the others are developed under the manifold assumption.

Representative inductive SSL algorithms include harmonic mixtures [28], LapSVMs [9], LapRLSs [9], LNP [2], simple SSL [29], and vector-valued manifold regularization [30]. For more detailed explanations about SSL algorithms, the reader is referred to [5] and [6].

All the above methods are formulated as a regularization framework, the same as the proposed LPDGL. The main difference between them is how to design the regularizer. In this sense, most of the above SSL algorithms can be derived from LPDGL by choosing different regularizers or incorporating other constraints, as shown in Fig. 2. For example, if the local smoothness term of LPDGL is removed (i.e., γ is set to 0) and let $\beta \to 0$, the result of LPDGL will get arbitrarily close to HF.[1] If we further require the obtained discrete labels belong to $\{\pm 1\}$, we reach the MinCut algorithm. In addition, LGC can be derived from LPDGL by adopting the normalized graph Laplacian. LGReg, GTAM, and Tikhonov regularization can also be easily derived by employing the techniques alongside the arrows. Some inductive algorithms, including LNP, LapRLS, and LapSVM, are also related to inductive LPDGL. If we set $\gamma = 0$ and adopt the hinge loss instead of the squared loss incorporated by LPDGL, we immediately obtain LapSVM. Similarly, if $\gamma = 0$ and a linear neighborhood graph in [2] is constructed, the proposed LPDGL will have the same formation as LNP. Of particular note is that the only difference between LapRLS and inductive LPDGL lies in the local smoothness term, of which the significance for boosting the accuracy will be demonstrated in Section V. Therefore, the proposed LPDGL has a strong relationship with other popular SSL methodologies, and they can be viewed as special cases of LPDGL.

[1]The precise solution of HF can be obtained by further relaxing the HF model derived from LPDGL as $\min_{\mathbf{f}} \mathbf{f}^T\mathbf{L}\mathbf{f}$, s.t. $f_i = y_i$ for $i = 1, 2, \ldots, l$. The detailed relaxation process is referred to [5].

V. Experiments

In this section, we validate the proposed LPDGL on several synthetic toy data sets, and compare LPDGL with some state-of-the-art SSL algorithms on a number of real-world collections. HF [1], LGC [12], AGR [24], LNP [2], LapRLS [9], LapSVM [9], LLReg [20], PBSSL [22], S4VM (RBF kernel) [23], and S4VM (linear kernel) [23] were adopted as baselines to evaluate the transductive ability of LPDGL. LNP [2], LapRLS [9], and LapSVM [9] were used for the inductive performance comparison because other algorithms do not have inductive ability. For fair comparison, HF, LGC, LLReg, PBSSL, LapRLS, LapSVM, and LPDGL were trained by the same k-NN graph[2] for each of the data sets appearing in this paper, and all the algorithms were conducted 10 times independently under each l (l represents the size of the labeled set) with randomly selected labeled set $\mathcal{L}$. However, at least one labeled example was selected in each class when $\mathcal{L}$ was generated. The reported accuracies and standard deviations of algorithms were calculated as the mean value of the outputs of these runs. To demonstrate the superiority of the proposed LPDGL over linear LPDGL mentioned in Section III, we also compared the performances of these two models on various data sets.

A. Toy Data

Synthetic 2-D and 3-D data was adopted in this section to visualize the transductive and inductive performance of LPDGL.

1) Transduction on 3-D Data: Two 3-D data sets, *Cylinder&Ring* and *Knot*, were used to test the transductive ability of LPDGL. The *Cylinder&Ring* data set [see Fig. 3(a)–(c)] forms like a cylinder surrounded by a ring, in which the cylinder with radius 0.2 represents the positive class and the ring with radius 0.8 constitutes the negative class.

[2]AGR builds a hyper-graph that is different from other algorithms. The k-NN graph in LNP is not symmetrical, which is different from that in HF, LGC, LLReg, PBSSL, LapRLS, LapSVM and LPDGL. S3VM and S4VM are not graph-based methods, so graph is not needed to train the classifier.

TABLE I

EXPERIMENTAL RESULTS ON THE BENCHMARK DATA SETS FOR THE VARIETY OF SSL ALGORITHMS [THE VALUES IN THE TABLE REPRESENT THE ERROR RATE (%). THE THREE BEST RESULTS FOR EACH DATA SET ARE MARKED IN RED, BLUE, AND GREEN, RESPECTIVELY]

Datasets	*USPS_Imbalanced*		*BCI*		*g241c*		*g241d*		*Digit1*		*COIL*	
l(#*Labeled Examples*)	10	100	10	100	10	100	10	100	10	100	10	100
1NN [31]	16.66	5.81	49.00	48.67	47.88	43.93	46.72	42.45	13.65	3.89	63.36	17.35
SVM [7]	20.03	9.75	49.85	34.31	47.32	23.11	46.66	24.64	30.60	5.53	68.36	22.93
MVU+1NN [32]	23.34	6.50	47.95	47.89	47.15	43.01	45.56	38.20	14.42	2.83	62.62	28.71
LEM+1NN [33]	19.82	7.64	48.74	44.83	44.05	40.28	43.22	37.49	23.47	6.12	65.91	23.27
QC+CMN [1]	13.61	6.36	50.36	46.22	39.96	22.05	46.55	28.20	9.80	3.15	59.63	10.03
Discrete Reg. [12]	16.07	4.68	49.51	47.67	49.59	43.65	49.05	41.65	12.64	2.77	63.38	9.61
TSVM [34]	25.20	9.77	49.15	33.25	24.71	18.46	50.08	22.42	17.77	6.15	67.50	25.80
SGT [19]	25.36	6.80	49.59	45.03	22.76	17.41	18.64	9.11	8.92	2.61	-	-
Cluster-Kernel [35]	19.41	9.68	48.31	35.17	48.28	13.49	42.05	4.95	18.73	3.79	67.32	21.99
Data-Dep. Reg. [36]	17.96	5.10	50.21	47.47	41.25	20.31	45.89	32.82	12.49	2.44	63.65	11.46
LDS [37]	17.57	4.96	49.27	43.97	28.85	18.04	50.63	23.74	15.63	3.46	61.90	13.72
Laplacian RLS [9]	18.99	4.68	48.97	31.36	43.95	24.36	45.68	26.46	5.44	2.92	54.54	11.92
CHM (normed) [38]	20.53	7.65	46.90	36.03	39.03	24.82	43.01	25.67	14.86	3.79	-	-
LPDGL(Linear)	19.77	13.44	43.50	24.90	44.15	34.04	45.11	33.62	38.11	10.19	73.21	70.64
LPDGL	17.88	5.09	48.17	34.52	42.73	21.54	42.01	23.90	5.37	2.23	61.69	7.27

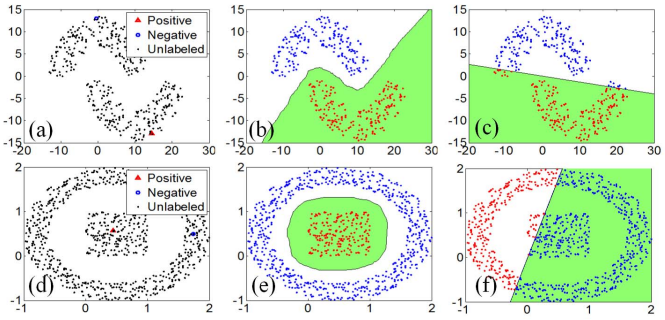


Fig. 4. Induction on *DoubleMoon* and *Square&Ring* data sets. (a) and (d) Initial states with the marked labeled examples. (b) and (e) Induction results, in which the decision boundaries are plotted. (c) and (f) Induction performances produced by LPDGL (linear).

The *Knot* data set is shaped like a knot composed of two crossing rings with radius of 0.8, and each ring represents a class [see Fig. 3(d)–(f)]. Both data sets are contaminated by the Gaussian noise of variance 0.1, and each class only has one labeled example, as shown in Fig. 3(a) and (d).

We adopted (5) and the linear model $f(\mathbf{x}) = \omega^T \mathbf{x}$, respectively, to train a transductive LPDGL to classify all the examples, given very few labeled examples. The parameters in LPDGL were $\sigma = 2$, $k = 5$, $\beta = 1$, $\gamma = 0.001$ for *Cylinder&Ring*, and $\sigma = 0.5$, $k = 5$, $\beta = 1$, $\gamma = 1$ for *Knot*. From Fig. 3(b) and (e), we observe that LPDGL can effectively detect the geometric structure of the data distribution, which leads to encouraging performances on both synthetic data sets. Therefore, the proposed algorithm has a satisfactory transductive ability. Comparatively, the LPDGL (linear) generates disastrous results [see Fig. 3(c) and (f)] because both data sets are highly nonlinear.

2) Visualization of Generalizability: LPDGL cannot only handle the transductive problems, but also shows great potential for dealing with inductive tasks. The *DoubleMoon* data set contains 400 examples, which are equally divided into two moons centered at (0, 0) and (10, 0), respectively. Each moon represents a class. The data distribution is displayed in Fig. 4(a), in which the labeled examples are marked in color. In *Square&Ring*, a square centered at (0.5, 0.5) is surrounded by a ring with the same center. The radius of the outer ring is 1.3, and the length of each side of the inner square is 1 [see Fig. 4(d)].

In these two data sets, only one labeled example was selected for each class. The training set Ψ was made up of these few labeled examples and the abundant unlabeled examples, based on which (13) was utilized to train an inductive LPDGL. In LPDGL, we set $\sigma = 5$ and $\alpha = \beta = \gamma = 1$ for both data sets, and established the 9-NN and 7-NN graphs for *DoubleMoon* and *Square&Ring*, respectively. Fig. 4(b) and (e) reveals that the white and green regions partitioned by the learned decision boundary are consistent with the geometry of the training examples. Consequently, the proposed LPDGL correctly classifies all the training examples, and a good generalizability is also guaranteed.

Besides, we provide the empirical illustrations on both synthetic data sets to show that the inductive LPDGL derived in RKHS performs better than the linear LPDGL. Fig. 4(c) and (f) presents the transductive results and the decision boundaries on each data set, which clearly reveal that the linear function $f(\mathbf{x}) = \omega^T \mathbf{x}$ cannot obtain as good performance as the LPDGL in RKHS for nonlinear data sets. Therefore, we suggest using (7) to implement transduction, and adopting (12) for induction.

B. Real Benchmark Data

This section compares the transductive accuracy of LPDGL with the results reported in [6] on six real benchmark data sets, including *USPS_Imbalanced*, *BCI*, *g241c*, *g241d*, *Digit1*, and *COIL*. The detailed information about these data sets and the performances of different SSL algorithms are provided in [6].

All the algorithms are implemented under $l = 10$ and $l = 100$ for each data set, and the reported accuracies are the mean values of the outputs of 12 independent runs. In each run, the labeled and unlabeled examples are randomly generated. However, the 12 different partitions of labeled and unlabeled sets in each data set are identical for all the compared algorithms. The parameters of LPDGL are optimally tuned to obtain the best performance. We set the number of neighbors of every data point

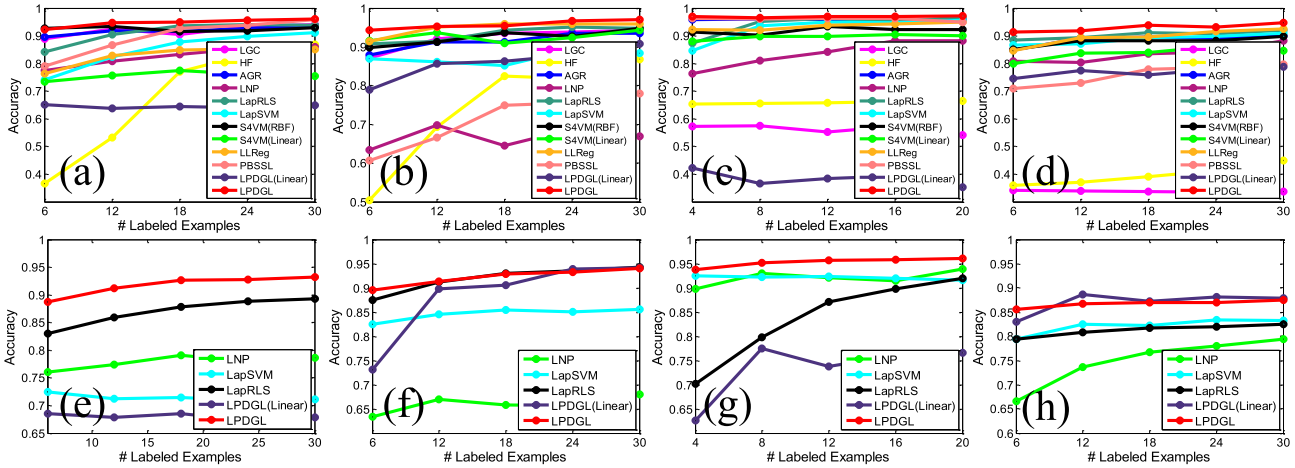


Fig. 5. Experimental results on four UCI data sets. (a) and (e) *Iris*. (b) and (f) *Wine*. (c) and (g) *BreastCancer*. (d) and (h) *Seeds*. Subplots in the first row compare the transductive performance of the algorithms, and the subplots in the second row compare their inductive performance.

TABLE II
SUMMARY OF FOUR UCI DATA SETS

	Iris	*Wine*	*BreastCancer*	*Seeds*
#Instances	150	178	683	210
#Attributes	4	13	10	7
#Classes	3	3	2	3

$k = 10, 8, 40, 30, 20, 7$ for *USPS_Imbalanced*, *BCI*, *g241c*, *g241d*, *Digit1*, and *COIL*, respectively, and the widths of RBF kernel are $\sigma = 2, 1, 1, 1, 5, 2$ correspondingly. Table I shows the error rates of different algorithms, which reveals that LPDGL achieves comparable performances with the typical state-of-the-art SSL algorithms. In particular, we observe that the linear LPDGL is outperformed by nonlinear model in all the data sets except *BCI*. We also want to mention that the relative size of positive and negative classes in *USPS_Imbalanced* is 1:4, so the experimental results on *USPS_Imbalanced* demonstrate that LPDGL can perfectly handle the situations when the examples of different classes are imbalanced.

C. UCI Data

We chose four University of California Irvine (UCI) machine learning repository data sets [39], *Iris*, *Wine*, *BreastCancer*, and *Seeds*, to compare the performance of LPDGL with other baselines. The detailed information of the four data sets is summarized in Table II. Throughout this paper, we adopt the one-versus-rest strategy to deal with multiclass classifications.

We first evaluated the transductive abilities of HF, LGC, AGR, LNP, LLReg, PBSSL, LapRLS, LapSVM, S4VM, and LPDGL by observing the classification accuracies with respect to different l for each data set. The reported results are averaged over the outputs of 10-independent runs under each l. In AGR, we chose the number of anchor points $s = 40, 40, 30, 50$ for *Iris*, *Wine*, *BreastCancer*, and *Seeds* data sets, respectively, and 5-NN graphs were constructed on these anchor points. The regression matrix in AGR was established by local anchor embedding, which was recommended by the authors [24]. The parameter λ in LLReg was set to 1 for all the UCI data sets. In *Iris*, *Wine*, *BreastCancer*, and *Seeds*, we constructed identical 8-NN, 6-NN, 7-NN, and 9-NN graphs correspondingly for HF, LGC, LLReg, PBSSL, LapRLS, LapSVM, and LPDGL. Parameters β and γ in LPDGL were set to 1, and two simulations of S4VM with RBF kernel and linear kernel were conducted on the four UCI data sets. The transductive results are presented in Fig. 5(a)–(d). We observe that some of the baselines achieve very encouraging performances on these data sets, e.g., LGC on *Iris*, S4VM on *Wine*, AGR on *BreastCancer*, and so on. However, the accuracies obtained by these baselines can still be improved by the proposed LPDGL, which demonstrates the strength of our algorithm.

To test inductive ability, we adopted LNP, LapRLS, and LapSVM as baselines because they are state-of-the-art inductive SSL algorithms. We not only compared the classification accuracies of LPDGL and other baselines, but also used the 5×2 cross-validation F-test (5×2 cv F-test) proposed by [40] to make statistical comparisons. The F-statistics value produced by the 5×2 cv F-test is to identify whether two algorithms achieve the same performance on the test set. The null hypothesis is that they do obtain the same test accuracy, and we reject this hypothesis with 95% confidence if the F-statistics value is >4.74. For conducting the 5×2 cv F-test, five replications of twofold cross validation were performed, and the four data sets were equally split randomly into training and test sets in each replication; however, the splits in the five replications were identical for all the compared algorithms. In the training and test sets, the number of examples belonging to a certain class is proportional to the number of examples of this class in the entire data set. Given $m_i^{(j)}$ as the difference of error rates generated by two algorithms on fold j ($j = 1, 2$) of replication i ($i = 1, 2, \ldots, 5$), then the mean error rate and the variance of replication i are $\bar{m}_i = (m_i^{(1)} + m_i^{(2)})/2$ and $s_i^2 = (m_i^{(1)} - \bar{m}_i)^2 + (m_i^{(2)} - \bar{m}_i)^2$, respectively. Therefore, according to [40], the F-statistics value $F = \sum_{i=1}^{5} \sum_{j=1}^{2} (m_i^{(j)})^2 / 2 \sum_{i=1}^{5} s_i^2$ obeys the F-distribution with 10 and 5 degrees of freedom.

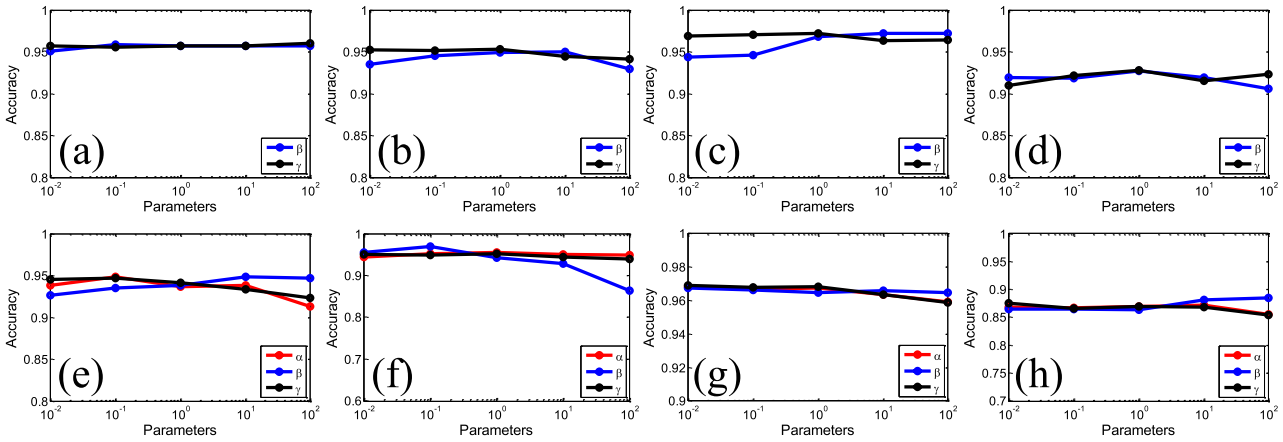


Fig. 6. Empirical studies on the parametric sensitivity of LPDGL. (a) and (e) *Iris*. (b) and (f) *Wine*. (c) and (g) *BreastCancer*. (d) and (h) *Seeds*. Subplots in the first row show the transductive results, and the subplots in the second row display the inductive results.

TABLE III
F-STATISTICS VALUES OF INDUCTIVE ALGORITHMS VERSUS LPDGL ON UCI DATA SETS (THE RECORDS <4.74 ARE MARKED IN RED, WHICH MEAN THAT THE NULL HYPOTHESIS IS ACCEPTED)

	l	LNP	LapSVM	LapRLS
Iris	6	16.69	26.31	6.53
	12	17.23	160.72	19.69
	18	8.0	132.11	28.17
	24	9.7	120.23	30.68
	30	5.5	343.85	19.01
Wine	6	13.87	23.12	2.94
	12	12.70	14.81	1.94
	18	19.18	49.67	2.67
	24	12.89	22.38	1.13
	30	11.25	43.12	1.74
BreastCancer	4	4.91	2.52	999.31
	8	1.81	11.99	384.48
	12	6.08	24.06	121.20
	16	2.40	24.51	23.79
	20	4.03	16.57	24.21
Seeds	6	20.85	36.36	8.80
	12	19.62	7.03	24.95
	18	32.18	31.78	52.62
	24	7.61	7.56	73.38
	30	8.76	12.01	33.81

In the four data sets, the established graphs for induction were the same as those for transduction. The weight α of the inductive term in (13) was set to 1 on all UCI data sets, and the parameters in LapRLS and LapSVM were also tuned properly to achieve the best performance. We reported the test accuracies as the mean outputs of five replications of twofold cross validation in every data set, and they are plotted in Fig. 5(e)–(h). We observe that LPDGL outperforms LNP, LapRLS, and LapSVM significantly on the UCI data sets with the exception of *Wine*. On the *Wine* data set, LPDGL achieves comparable performance with LapRLS. The F-statistics values of baselines versus LPDGL are listed in Table III. The acceptable cases are marked in color, which means that the performance of the two algorithms is comparable. Note that the null hypothesis is rejected in most cases, so the superiority of LPDGL to the compared algorithms is statistically demonstrated. However, the null hypothesis is accepted on *Wine* for LapRLS, because there is no significant difference between the error rates of LPDGL and LapRLS, as revealed by Fig. 5(f), so the performances of the two algorithms on the *Wine* data set are considered to be essentially identical.

We studied the parametric sensitivity for both transductive and inductive tasks in particular. We observed accuracies under $l = 30$ on *Iris*, *Wine*, *Seeds*, and $l = 20$ on *BreastCancer*, with a wide choice of parameters α, β, and γ (Fig. 6). Recall that we have theoretically proved that the transductive performance of LPDGL is very robust to the variations of β and γ. Here, we empirically verify this point by examining the accuracies with one parameter changed and the other fixed to 1. Fig. 6(a)–(d) suggests that these two parameters have little impact on the transductive performance, which is consistent with our theoretical understanding in Sections II-A and II-B. The parametric sensitivity under the inductive case was also investigated by varying one of α, β, and γ, and fixing the remaining two parameters to 1. Fig. 6(e)–(h) reveals that although the parameters cover a wide range, i.e., 10^{-2}–10^{2}, the accuracy remains substantially unchanged on the four data sets. Therefore, we conclude that LPDGL shows profound parametric sensitivity for both transductive and inductive settings.

D. Handwritten Digit Recognition

The *USPS* data set[3] was adopted to assess the ability of algorithms to recognize handwritten digits. This data set contains 9298 digit images belonging to 10 classes, i.e., digits 0–9. The resolution of all images is 16×16, so the pixelwise feature we adopted was 256 dimensions, in which every dimension represents the gray value of corresponding pixels.

We used the whole data set to test the transductive performances of various algorithms. A number of the examples were selected as a labeled set, and the rest were taken as the unlabeled examples. The classification accuracies were particularly observed when l changed from 100 to 500. The 10-NN graph was established, and the parameter σ for computing the edge weights was set to 5. For AGR, 300 anchor

[3]http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

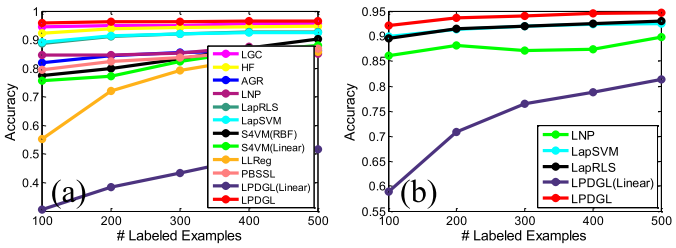


Fig. 7. Experimental results on *USPS* data set. (a) Transductive results. (b) Inductive results.

points were automatically generated by K-means clustering, and a 7-NN graph was constructed on these anchor points. In addition, β and γ in LPDGL were tuned to 10 and 1, and we used the default parameter settings in S4VM.

Fig. 7(a) shows the transductive results. It is observed that LPDGL achieves higher classification accuracy than other baselines. By comparison, LGC and HF achieve slight lower accuracies than LPDGL, i.e., 95% approximately. LapSVM and LapRLS achieve similar results. The performances of LNP, LLReg, PBSSL, S4VM, and AGR are not as satisfactory as the other five methods. Therefore, the proposed LPDGL is very effective on handwritten digit recognition, though only a small number of labeled examples are given.

To test the inductive abilities of algorithms, we split the original data set into a training set and a test set. 600 examples per class were extracted to form the training set, and the remaining 3298 examples served as the test set. The weight α of the inductive term in (13) was tuned to 1 to extend LPDGL to out-of-sample data. Fig. 7(b) reveals that LPDGL is superior to LNP, LapRLS, and LapSVM in terms of inductive accuracy. Moreover, it can be observed that the inductive performance of LPDGL does not decrease the transductive settings too much. The reason is that LPDGL has successfully discovered the manifold from the training set in advance, so even though the test data are previously unseen, LPDGL can precisely predict their labels according to their locations on the manifold. Therefore, LPDGL achieves similar performances on transductive and inductive settings, which again demonstrates generalizability. Comparatively, the LPDGL (linear) is significantly outperformed by the kernerlized LPDGL for both transductive and inductive settings, which also shows the strength of the developed nonlinear model.

E. Face Recognition

Face recognition has been widely studied as a traditional research area of computer vision because of the extensive practical demands. LPDGL was performed on two face data sets: *Yale*[4] and *Labeled Face in the Wild (LFW)*[5] [41]. The face images in *Yale* are collected in a laboratory environment. In contrast, the images in *LFW* are directly downloaded from the web, and faces are presented in natural scenes.

1) Yale: The *Yale* face data set contains 165 grayscale images of 15 individuals. Each individual has 11 face images covering a variety of facial expressions and configurations, including: center light, wearing glasses, happy, left light, wearing no glasses, normal, right light, sad, sleepy, surprised, and wink. The resolution of every image is 64×64, so we directly rearranged each image to a 4096-D long vector as input for all the algorithms.

[4]http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[5]http://vis-www.cs.umass.edu/lfw/

TABLE IV
TRANSDUCTIVE COMPARISON ON *Yale* DATA SET

	$l = 30$	$l = 60$
LGC	0.66 ± 0.06	0.76 ± 0.02
HF	0.65 ± 0.04	0.79 ± 0.01
AGR	0.50 ± 0.03	0.64 ± 0.02
LNP	0.32 ± 0.05	0.34 ± 0.04
LapRLS	0.63 ± 0.05	0.71 ± 0.03
LapSVM	0.63 ± 0.05	0.72 ± 0.03
S4VM(Linear)	0.27 ± 0.07	0.52 ± 0.06
S4VM(RBF)	0.11 ± 0.02	0.23 ± 0.04
LLReg	0.65 ± 0.08	0.79 ± 0.09
PBSSL	0.51 ± 0.05	0.67 ± 0.02
LPDGL(Linear)	0.65 ± 0.04	0.79 ± 0.01
LPDGL	0.67 ± 0.03	0.81 ± 0.01

TABLE V
INDUCTIVE COMPARISON ON *Yale* DATA SET

	$l = 30$	$l = 60$
LNP	0.10 ± 0.04	0.15 ± 0.05
LapSVM	0.69 ± 0.01	0.77 ± 0.01
LapRLS	0.68 ± 0.01	0.79 ± 0.01
LPDGL(Linear)	0.58 ± 0.06	0.80 ± 0.01
LPDGL	0.69 ± 0.04	0.83 ± 0.03

The transductive abilities of LPDGL and other baselines were first evaluated. In this experiment, we chose $\sigma = 10$ and $k = 5$ for graph construction, and other parameters for LPDGL were $\beta = \gamma = 1$. In AGR, a 7-NN graph was built on the 35 anchor points. In LNP, we established a 9-NN graph to achieve the best performance. In LLReg, λ was optimally tuned to 1. The accuracies of algorithms with different l are listed in Table IV, in which the best record under each l is marked in red. The proposed LPDGL is able to achieve the highest accuracy, and the small standard deviations suggest that LPDGL is very robust to the choice of labeled examples.

Inductive performance was also studied on the *Yale* data set. We chose the first six examples of every individual to establish the training set, and the other five examples made up the test set; the sizes of the training and test sets were $6 \times 15 = 90$ and $5 \times 15 = 75$, respectively. Labeled sets of size $l = 30$ and 60 were then randomly generated in the training set. The main difficulty for induction on *Yale* is that the expressions or appearances of test faces are never observed in the training set, which requires the classifiers to be immune to illumination or changes in facial expression. The inductive accuracies are compared in Table V and suggest that LPDGL outperforms other baselines when l varies from small to large. In particular, LPDGL achieves 83% accuracy when $l = 60$, which is a very encouraging result. It is widely acknowledged that although faces have different expressions or observation angles, they are actually embedded in a potential manifold [33]. Fortunately, LPDGL is exactly developed on the manifold assumption, so it is able to recognize faces accurately even though their appearance differs dramatically. It is also worth pointing out that we have only adopted the simple pixel-wise gray

TABLE VI
TRANSDUCTIVE COMPARISON ON *LFW* DATA SET

	$l = 50$	$l = 100$	$l = 150$	$l = 200$
LGC	0.50 ± 0.07	0.60 ± 0.05	0.65 ± 0.08	0.69 ± 0.06
HF	0.66 ± 0.03	0.78 ± 0.02	0.83 ± 0.01	0.87 ± 0.01
AGR	0.60 ± 0.03	0.71 ± 0.01	0.76 ± 0.02	0.80 ± 0.01
LNP	0.32 ± 0.07	0.38 ± 0.16	0.57 ± 0.12	0.59 ± 0.11
LapRLS	0.48 ± 0.03	0.62 ± 0.04	0.71 ± 0.03	0.75 ± 0.03
LapSVM	0.57 ± 0.02	0.70 ± 0.03	0.74 ± 0.03	0.76 ± 0.03
S4VM(Linear)	0.56 ± 0.05	0.68 ± 0.03	0.73 ± 0.03	0.77 ± 0.02
S4VM(RBF)	0.45 ± 0.06	0.61 ± 0.02	0.70 ± 0.02	0.73 ± 0.02
LLReg	0.52 ± 0.04	0.69 ± 0.02	0.86 ± 0.02	0.88 ± 0.01
PBSSL	0.33 ± 0.03	0.46 ± 0.02	0.58 ± 0.02	0.68 ± 0.02
LPDGL(Linear)	0.43 ± 0.02	0.59 ± 0.04	0.64 ± 0.02	0.71 ± 0.02
LPDGL	0.71 ± 0.02	0.81 ± 0.02	0.86 ± 0.01	0.90 ± 0.01

TABLE VII
INDUCTIVE COMPARISON ON *LFW* DATA SET

	$l = 50$	$l = 100$	$l = 150$	$l = 200$
LNP	0.30 ± 0.07	0.38 ± 0.09	0.45 ± 0.13	0.45 ± 0.09
LapSVM	0.65 ± 0.01	0.69 ± 0.03	0.75 ± 0.02	0.76 ± 0.01
LapRLS	0.67 ± 0.04	0.73 ± 0.02	0.78 ± 0.01	0.79 ± 0.01
LPDGL(Linear)	0.68 ± 0.04	0.78 ± 0.03	0.81 ± 0.03	0.83 ± 0.01
LPDGL	0.70 ± 0.03	0.78 ± 0.03	0.80 ± 0.02	0.83 ± 0.02

TABLE VIII
TRANSDUCTIVE RESULTS ON *HockeyFight* DATA SET

	$l = 40$	$l = 80$	$l = 120$	$l = 160$
LGC	0.80 ± 0.03	0.82 ± 0.02	0.83 ± 0.02	0.84 ± 0.01
HF	0.80 ± 0.02	0.84 ± 0.01	0.86 ± 0.01	0.87 ± 0.01
AGR	0.79 ± 0.02	0.82 ± 0.01	0.83 ± 0.01	0.83 ± 0.01
LNP	0.61 ± 0.08	0.65 ± 0.10	0.65 ± 0.09	0.67 ± 0.11
LapRLS	0.72 ± 0.02	0.76 ± 0.01	0.79 ± 0.01	0.79 ± 0.01
LapSVM	0.67 ± 0.03	0.66 ± 0.02	0.70 ± 0.02	0.71 ± 0.01
S4VM(Linear)	0.80 ± 0.05	0.84 ± 0.02	0.84 ± 0.03	0.86 ± 0.01
S4VM(RBF)	0.81 ± 0.03	0.84 ± 0.01	0.86 ± 0.01	0.87 ± 0.01
LLReg	0.78 ± 0.04	0.79 ± 0.01	0.82 ± 0.01	0.82 ± 0.01
PBSSL	0.74 ± 0.02	0.76 ± 0.01	0.78 ± 0.01	0.78 ± 0.01
LPDGL(Linear)	0.81 ± 0.02	0.84 ± 0.02	0.87 ± 0.00	0.87 ± 0.02
LPDGL	0.81 ± 0.03	0.85 ± 0.01	0.87 ± 0.01	0.88 ± 0.01

TABLE IX
INDUCTIVE RESULTS ON *HockeyFight* DATA SET

	$l = 40$	$l = 80$	$l = 120$	$l = 160$
LNP	0.58 ± 0.12	0.58 ± 0.08	0.58 ± 0.10	0.59 ± 0.11
LapSVM	0.59 ± 0.02	0.61 ± 0.01	0.61 ± 0.01	0.65 ± 0.01
LapRLS	0.70 ± 0.01	0.73 ± 0.01	0.73 ± 0.01	0.74 ± 0.01
LPDGL(Linear)	0.75 ± 0.04	0.76 ± 0.01	0.76 ± 0.01	0.76 ± 0.01
LPDGL	0.71 ± 0.02	0.73 ± 0.03	0.74 ± 0.02	0.75 ± 0.01

values of images as features. If more high-level features are utilized, the performance of LPDGL is expected to be further improved.

2) LFW: *LFW* is a gigantic collection of face images gathered directly from the web. The facial expressions, observation angle, illumination conditions, and background setting are not intentionally controlled for recognition; therefore, identifying faces in such unconstrained situations is a big challenge. This data set contains >13 000 face images, and each face is labeled with the name of the person. The faces in all the images are detected by the Viola–Jones detector [42].

Most people in the original *LFW* have fewer than five images, which is insufficient for splitting into training and test sets, so we used a subset of *LFW* by choosing persons who have more than 30 face images. We chose the images of politicians Toledo, Sharon, Schwarzenegger, Powell, Rumsfeld, Bush, and Arroyo, and the images of sports stars Agassi, Beckham, and Hewitt. There were thus 392 examples belonging to 10 people in total in the subset. We adopted the 73-D feature developed by [43], which describes the biometrics traits of visual appearance, such as gender, race, age, and hair color.

To test the transductive performance, a 6-NN graph with $\sigma = 5$ was built on the entire data set. Other parameters in LPDGL were $\beta = \gamma = 1$. Algorithms are compared in Table VI, in which the best performance under each l is marked in color. It is observed that LPDGL achieves very satisfactory results and significantly outperforms other methods. In particular, the proposed LPDGL obtains very high accuracy under relatively small l, e.g., 71% under $l = 50$ and 81% under $l = 100$, which further demonstrates the effectiveness of LPDGL.

Inductive experiments were conducted by separating the data set into a training set of 250 examples and a test set of 142 examples. The inductive results of algorithms under different l are listed in Table VII, from which we find that the proposed LPDGL achieves the best performance compared with other baselines. Table VII also reveals that the accuracy of LPDGL exceeds 80% when l is >150, so LPDGL has strong potential for successful unconstrained face recognition.

F. Violent Behavior Detection

In recent years, various intelligent surveillance techniques have been applied to ensure public safety. One desirable application is to permit computers automatically detect violent behavior, such as fighting and robbery, in surveillance videos. In this section, we utilize the proposed LPDGL to detect fight behavior. The *HockeyFight*[6] data set is made up of 1000 video clips collected in ice hockey competitions, of which 500 contain fight behavior and 500 are nonfight sequences. The task is to identify the clips with fighting. As with [44], we adopted the space-time interest points and motion SIFT as action descriptors, and used the bag-of-words approach to represent each video clip as a histogram over 100 visual words. Every clip in the data set was, therefore, characterized by a 100-D feature vector.

A 5-NN graph was exploited to evaluate the transductive performance of HF, LGC, LLReg, PBSSL, LapRLS, LapSVM, and LPDGL. In LNP and AGR, we chose 20 and five neighbors, respectively, for graph construction. Transductive accuracies with different l are listed in Table VIII. We observe that HF, S4VM, and LPDGL already achieve >80% accuracy, which is a very encouraging result. Of particular note is that LPDGL can still improve the performances of S4VM and HF, so its superiority is demonstrated.

Inductive experiments were performed by splitting the original data set into a training set of 600 examples and a test set of 400 test examples. Fight clips and nonfight clips constituted 50% for each of both the training set and the test set. The results of the algorithms are displayed in Table IX, in which the best performance under each l is marked in red. We find that LPDGL obtains very impressive inductive results.

[6]http://visilab.etsii.uclm.es/personas/oscar/FightDetection/index.html

VI. Conclusion

This paper has proposed a manifold-based SSL algorithm called LPDGL. By adopting the DGL, a local smoothness term was naturally incorporated. This term can effectively prevent erroneous label propagation between classes by suppressing the labels of ambiguous examples, such as the bridge point mentioned in our paper. Fig. 2 implies that the only difference between LapRLS and inductive LPDGL is that LPDGL has the local smoothness term but LapRLS does not, so the better performances of LPDGL over LapRLS in experiments also validates the importance of this term.

The proposed method has several profound properties, which lead to the superiority of LPDGL over other SSL algorithms. First, LPDGL is formulated as a convex optimization framework, so the obtained decision function is globally optimal. Second, the classification performance is insensitive to the change of parameters, which indicates that the parameters in LPDGL are very easy to tune. Third, there exists a theoretical bound for the generalization error, so the test examples can be classified reliably and accurately. Fourth, LPDGL can be regarded as a unified framework of various SSL algorithms, so it combines the advantages of different methodologies. Finally, the standard deviations of LPDGL listed in Tables IV–IX are very small, which reflects that the selection of initially labeled examples will not influence the final results significantly.

The primary computational burden of LPDGL lies in the inversion of the matrices in (7) and (14). Fortunately, the order of the matrix to be inverted in (7) is equal to the examples' dimension d, so inverting such a matrix is usually efficient. However, the matrix to be inverted in (14) is of size $n \times n$, which can be quite large if there are massive training examples. Therefore, tackling the big data problem for inductive LPDGL is an important trend for future investigation.

References

[1] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 912–919.

[2] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1600–1615, Sep. 2009.

[3] Y. Huang, D. Xu, and F. Nie, "Semi-supervised dimension reduction using trace ratio criterion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 519–526, Mar. 2012.

[4] Y. Wang and S. Chen, "Safety-aware semi-supervised classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 11, pp. 1763–1772, Nov. 2013.

[5] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*. San Rafael, CA, USA: Morgan & Claypool Publishers, 2009.

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

[7] V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[8] H. Xue, S. Chen, and Q. Yang, "Structural regularized support vector machine: A framework for structural large margin classifier," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 573–587, Apr. 2011.

[9] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Dec. 2006.

[10] F. Morbidi, "The deformed consensus protocol," *Automatica*, vol. 49, no. 10, pp. 3049–3055, 2013.

[11] X. Li and Y. Guo, "Adaptive active learning for image classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 859–866.

[12] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2003, pp. 321–328.

[13] P. D. Hislop and I. M. Sigal, *Introduction to Spectral Theory: With Applications to Schrödinger Operators*. New York, NY, USA: Springer-Verlag, 1996.

[14] P. Lancaster and M. Tismenetsky, *Theory of Matrices*, vol. 2. New York, NY, USA: Academic, 1969.

[15] T. Hofmann, B. Schölkopf, and A. J. Smola, "A tutorial review of RKHS methods in machine learning," 2005.

[16] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.

[17] H. Xu and S. Mannor, "Robustness and generalization," *Mach. Learn.*, vol. 86, no. 3, pp. 391–423, 2012.

[18] M. Belkin, I. Matveeva, and P. Niyogi, "Tikhonov regularization and semi-supervised learning on large graphs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 3. May 2004, pp. 1000–1003.

[19] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 290–297.

[20] M. Wu and B. Schölkopf, "Transductive classification via local learning regularization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2007, pp. 628–635.

[21] F. Wang, T. Li, G. Wang, and C. Zhang, "Semi supervised classification using local and global regularization," in *Proc. 23rd AAAI Conf. Artif. Intell.*, 2008, pp. 726–731.

[22] H. Chang and D.-Y. Yeung, "Robust path-based clustering for the unsupervised and semi-supervised learning settings," Dept. Comput. Sci., Hong Kong Univ. Sci. Technol., Hong Kong, Tech. Rep. HKUST-CS04-04, 2004.

[23] Y. Li and Z. Zhou, "Towards making unlabeled data never hurt," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 1081–1088.

[24] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 679–686.

[25] J. Wang, T. Jebara, and S.-F. Chang, "Graph transduction via alternating minimization," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1144–1151.

[26] L. Chen, I. W. Tsang, and D. Xu, "Laplacian embedded regression for scalable manifold regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 6, pp. 902–915, Jun. 2012.

[27] Y. Wang, S. Chen, and Z.-H. Zhou, "New semi-supervised classification method based on modified cluster assumption," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 689–702, Apr. 2012.

[28] X. Zhu and J. Lafferty, "Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 1052–1059.

[29] M. Ji, T. Yang, B. Lin, R. Jin, and J. Han, "A simple algorithm for semi-supervised learning with improved generalization error bound," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1223–1230.

[30] M. H. Quang, L. Bazzani, and V. Murino, "A unifying framework for vector-valued manifold regularization and multi-view learning," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 100–108.

[31] C.-L. Chang, "Finding prototypes for nearest neighbor classifiers," *IEEE Trans. Comput.*, vol. 100, no. 11, pp. 1179–1184, Nov. 1974.

[32] K. Q. Weinberger and L. K. Saul, "An introduction to nonlinear dimensionality reduction by maximum variance unfolding," in *Proc. 21st Nat. Conf. Artif. Intell. (AAAI)*, 2006, pp. 1683–1686.

[33] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[34] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th Int. Conf. Mach. Learn.*, 1999, pp. 200–209.

[35] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster kernels for semi-supervised learning," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2002, pp. 601–608.

[36] M. Szummer and T. S. Jaakkola, "Information regularization with partially labeled data," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2002, pp. 1049–1056.

[37] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. Int. Workshop Artif. Intell. Statist.*, 2004, pp. 57–64.
[38] C. J. C. Burges and J. C. Platt, "Semi-supervised learning with conditional harmonic mixing," in *Semi-Supervised Learning*, vol. 28. Cambridge, MA, USA: MIT Press, 2005.
[39] A. Frank and A. Asuncion. (2010). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
[40] E. Alpaydin, "Combined 5 × 2 cv F test for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 11, no. 8, pp. 1885–1892, 1999.
[41] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. 07-49, 2007.
[42] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2001, pp. I-511–I-518.
[43] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar, "Attribute and simile classifiers for face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Sep./Oct. 2009, pp. 365–372.
[44] E. B. Nievas, O. D. Suarez, G. B. García, and R. Sukthankar, "Violence detection in video using computer vision techniques," in *Computer Analysis of Images and Patterns*. New York, NY, USA: Springer-Verlag, 2011, pp. 332–339.

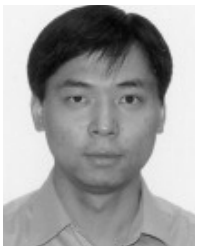
Chen Gong received the bachelor's degree from the East China University of Science and Technology, Shanghai, China, in 2010. He is currently pursuing the dual Ph.D. degree with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, and the Centre for Quantum Computation & Intelligent Systems, University of Technology, Sydney, Sydney, NSW, Australia, under the supervision of Prof. J. Yang and Prof. D. Tao.

His current research interests include machine learning, data mining, and learning-based vision problems.

Dr. Gong has authored 21 technical papers at prominent journals and conferences, such as the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CYBERNETICS, the Association for the Advancement of Artificial Intelligence Conference, and the IEEE International Conference on Multimedia and Expo.

Tongliang Liu received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2012. He is currently pursuing the Ph.D. degree in computer science with the University of Technology at Sydney, Sydney, NSW, Australia.

His current research interests include machine learning, computer vision, and optimization.

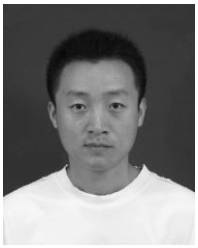


Dacheng Tao (M'07–SM'12–F'15) is currently a Professor of Computer Science with the Centre for Quantum Computation & Intelligent Systems, and the Faculty of Engineering and Information Technology, University of Technology, Sydney, Sydney, NSW, Australia. He mainly applies statistics and mathematics to data analytics. His research results have expounded in 1 monograph and over 100 publications at prestigious journals and prominent conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the *Journal of Machine Learning Research*, the *International Journal of Computer Vision*, the Neural Information Processing Systems Conference, the International Conference on Machine Learning, the IEEE Conference on Computer Vision and Pattern Recognition, the International Conference on Computer Vision, the European Conference on Computer Vision, the International Conference on Artificial Intelligence and Statistics, the IEEE International Conference on Data Mining (ICDM), and the ACM SIGKDD and Multimedia conferences. His current research interests spread across computer vision, data science, image processing, machine learning, neural networks, and video surveillance.

Dr. Tao was a recipient of several best paper awards, such as the Best Theory/Algorithm Paper Runner-Up Award in the IEEE ICDM'07, the best student paper award in the IEEE ICDM'13, and the 10-Year Highest-Impact Paper Award in the ICDM'14.

Keren Fu received the B.Sc. degree in automation from the Huazhong University of Science and Technology, Wuhan, China, in 2011. He is currently pursuing the Ph.D. degree with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China, under the supervision of Prof. J. Yang.

His current research interests include object detection, saliency detection, visual tracking, and machine learning.



Enmei Tu received the B.Sc. and M.Sc. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2007 and 2010, respectively. He is currently pursuing the Ph.D. degree with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China, under the supervision of Prof. J. Yang.

His current research interests include machine learning, computer vision, and remote sensing data processing.

Jie Yang received the Ph.D. degree from the Department of Computer Science, University of Hamburg, Hamburg, Germany, in 1994.

He is currently a Professor with the Institute of Image Processing and Pattern recognition, Shanghai Jiao Tong University, Shanghai, China. He has led many research projects (e.g., National Science Foundation, 863 National High Technology Plan), published one book in Germany, and authored over 200 journal papers. His current research interests include object detection and recognition, data fusion and data mining, and medical image processing.