Neurocomputing 157 (2015) 173–186

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Posterior Distribution Learning (PDL): A novel supervised learning framework using unlabeled samples to improve classification performance

Enmei Tu [a], Jie Yang [a,*], Nicola Kasabov [b], Yaqian Zhang [c]

[a] Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, China
[b] The Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, New Zealand
[c] Department of Electronic Engineering, Shanghai Jiao Tong University, China

ARTICLE INFO

Article history:
Received 22 September 2014
Received in revised form
29 December 2014
Accepted 10 January 2015
Communicated by D. Tao
Available online 24 January 2015

Keywords:
Supervised learning
Posterior Distribution Learning
Supervised manifold classification

ABSTRACT

Building a supervised model with good generalization ability in data space using a tiny number of labeled samples is always practically significant and attractive, because labeled samples are generally very expensive to obtain, as the labeling process usually takes much time and resource. In this paper we propose a new supervised two-step learning framework, Posterior Distribution Learning (PDL), to build a robust supervised model in data space by first describing a new constrained graph method to estimate the posterior probability of the unlabeled samples in learning set and then extending a real function regression model to a vector-valued function model to fit a nonlinear function for the posterior probability distribution in the input data space. Compared with traditional supervised learning method, PDL can significantly reduce the demand upon training samples and exhibits the potential to perform nonlinear manifold classification. Experimental results on both synthetic and real world data sets are presented to demonstrate the validity of the proposed supervised architecture.

1. Introduction

Supervised learning method is widely used in various areas, because once it is well trained, it builds a model in the whole input space and thus can classify any unseen sample easily. However, in order to obtain such a model with good generalization ability, one has to train the model with sufficient well labeled and uniformly distributed samples. But in many real applications, the cost of high-quality labeled samples is generally very expensive, as the labeling process usually consumes much time and resource. This poses an obstacle to those applications in which one needs to classify large amount of unlabeled samples with a very few labeled samples at hands, such as the applications of remote sensing data classification, gene and protein action prediction, image and language processing.

The problem of incorporating the distribution of unlabeled samples to remedy the insufficiency of labeled samples to improve the performance of supervised learning method has been studied for many years and previous works can be mainly casted into two categories. One is the co-training strategy [1–4]. The main idea is to train two or more different classifiers using the labeled samples first, and then each classifier tags some samples and adds them to other classifier training sets. This process repeats until all unlabeled samples are classified. These algorithms either require the data set that has two or more distinct views, each of which is sufficient to make good classification alone, or require different classifiers that can discover the diversity in the data set. But most of the real world problems do not meet these requirements. The other category is the partially supervised classification [5–7]. These algorithms are restricted to binary classification tasks, in which only labeled samples of one class (called positive samples) are known and labeled samples of the other class (called negative samples, possibly a mixture of several classes) are unknown. Theoretical studies of this strategy can be found in [7,8]. There are also some other methods to use unlabeled samples in the training process. In self-enhancement training [9], one supervised classifier is trained iteratively or sequentially, and each training epoch takes previous classification results into consideration. But it lacks a general rule to prevent gradual degeneration. Multiview learning [10–15] uses different distinct properties (or views)[1] of the (both labeled and unlabeled) data to improve results, but the requirement of different distinct properties is a rather strong precondition and most of the existing problems do not meet this requirement. Semi-supervised learning, such as [16–21], can also significantly enhance

* Corresponding author.
E-mail address: jieyang@sjtu.edu.cn (J. Yang).

[1] Therefore, some of them also belong to co-training category.

http://dx.doi.org/10.1016/j.neucom.2015.01.020
0925-2312/

classification accuracy by using unlabeled samples, but they are transductive learning and cannot build a supervised model in input data space and thus are very inconvenient while processing sequential data. (We will further discuss the relationship between our proposed method and semi-supervised learning method in the last section.) Therefore, how can we make use of the easily available unlabeled samples to build a reliable multiclass supervised learning model in input data space is still an interesting problem worth devotion and of great practical significance.

In this paper we propose a new supervised learning framework, Posterior Distribution Learning (PDL) which consists of posterior probability estimation component and posterior distributions regression component, to train a robust supervised model using a small number of labeled samples and a large number of unlabeled samples. The first component estimates the posterior probabilities of a learning set and the second component regresses a single multivariate posterior distribution function for all classes in the data space. When a new sample comes, the posterior distribution function can give directly its posterior probability of each class, so the class label can be obtained using the minimum Bayes error rule. Unlike previous works, the new supervised learning framework has the following characteristics: (1) it does not put constraints on the classification task, such as multi-view property or diversity between different classifiers; (2) it is more time-efficiency because it does not require training classifier (s) iteratively. Instead, it builds a single model for all classes in the input space.

The main contributions of this paper are as follows: (1) The LGC algorithm often leads to incorrect propagation while a large amount of ambiguous points appear between different classes. We improve it by using a locally adaptive learning rate instead of a globally fixed learning rate in the original algorithm and, meanwhile, incorporating pairwise constraint information in the algorithm to increase the robustness algorithm to ambiguous points during the propagation process; (2) The least square support vector machine (LS-SVM) is a powerful tool to regress a nonlinear function, but it is designed only for scalar function and thus only suitable for binary-class problem. In this paper we extend it to be able to regress any vector-valued nonlinear function once and thus make it applicable to handle multiclass problem by one model; (3) Most importantly, to resolve the contradiction between training a supervised model with good generalization ability and the insufficiency of a large number of well labeled training samples, we proposed a new two-step learning framework which can build a single model in the data space for multiclass classification using only a tiny number of training samples. We also implemented two versions (using multiclass LS-SVM and artificial neural networks as the density regressor) of the new learning framework. Experimental results on both synthetic and real world data sets are presented to demonstrate validity and effectiveness of the proposed framework and algorithm.

The remainder of this paper is organized as follows: Section 2 proposes the new PDL learning framework. Section 3 introduces a new posterior probability propagation algorithm and Section 4 describes a multivariate model for regression. Section 5 provides methods to determine propagation matrix and weight scheme and gives a theoretical study of the propagation algorithm. The simulations and comparisons results are presented in Section 6, followed by discussions and conclusions in Section 7.

Table 1
Symbols definition in this paper.

X	Whole data set
X_T	Labeled set
X_U	Unlabeled set
X_L	Learning set
n	Whole data set capacity
t	Labeled set capacity
l	Learning set capacity
ω	Class label
d	Sample feature length
C	Class number

2. Posterior Distribution Learning (PDL) framework

Let us consider a data set $X=\{x_1,x_2,\ldots,x_{t-1},x_t,x_{t+1},\ldots,x_n\}$, in which the first t samples are labeled by $\omega_j\in\{1,2,\ldots,C\}, j=1,\ldots,t$ and the last $n-t$ samples are unlabeled. C is the class number. Each $x_j\in R^d, j=1,2,\ldots,n$ is sampled independently according to a distribution function $p(x)$. We use $X_T=\{x_1,x_2,\ldots,x_{t-1},x_t\}$ and $X_U=\{x_{t+1},x_{t+2},\ldots,x_n\}$ to denote the labeled samples set (also referred as training set) and the unlabeled samples set, respectively. With a little abuse of notation, we also use X (similarly, X_T and X_U) to denote the data matrix $(x_1,x_2,\ldots,x_n)\in R^{d\times n}$. We define a learning set $X_L=\{x_1,x_2,\ldots,x_{t-1},x_t,x_{t+1},\ldots,x_l\}$, which contains all the labeled samples and $l-t$ ($l\le n$) unlabeled samples in X. For reading convenience, we also list the meaning of these symbols in Table 1.

The diagram of the proposed supervised learning framework is shown in Fig. 1. Note that the learning set X_L contains both labeled and unlabeled samples. The posterior estimator computes $P(\omega_j=i|x_j)$, the posterior probability of x_j coming from class i.

We require the output of the estimator to have the following properties:

- $\forall x_j\in X_L,\ P(\omega_j=i|x_j)\ge 0$.
- $\forall x_j\in X_L,\ \sum_{i=1}^{C}P(\omega_j=i|x_j)=1$.
- $x_j\in X_L,\ P(\omega_j=i|x_j)=1$ if x_j is labeled to class i.
- $x_j\in X_L,\ P(\omega_j=i|x_j)>P(\omega_j=k|x_j),\ k\ne i$ and $k=1,\ldots,C$, if x_j is unlabeled and more similar to samples from class i than class k.

Let $F(x_j)=(P(\omega_j=1|x_j),P(\omega_j=2|x_j),\ldots,P(\omega_j=C|x_j))\in R^C$, then $F(x_j)$, $j=1,\ldots,l$ can be regarded as points sampled from some continuous vector-valued function $F(x)$, the posterior function which gives the posterior probabilities of all classes at once. We try to build a multivariate model to fit the posterior function $F(x)$. After this, for a new sample x, its posterior probability can be obtained directly with $F(x)$ and the class label is determined by the minimum Bayes error rule.

It is worth mentioning that while the algorithms we present in Sections 3 and 4 are for the two components, the PDL framework is a very generic learning architecture and other algorithms can be adopted for the two components, such as the artificial neural network we used for regression in Section 6.

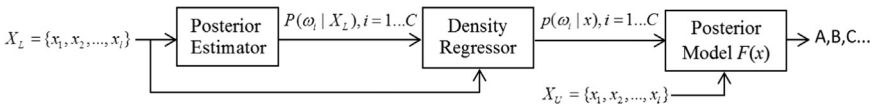


Fig. 1. Diagram of the proposed supervised learning framework that includes distribution of unlabeled samples into training stage.

3. Posterior estimator: propagate posterior probability from labeled samples to unlabeled samples

The posterior probabilities of a labeled sample are known, i.e. $P(\omega_j = i|x_j) = 1$ if x_j is from class i and $P(\omega_j = k|x_j) = 0$ if $k \neq i$. For unlabeled samples x_j, the posterior probabilities $P(\omega_j = i|x_j)$, $i =.1\ C$ are unknown and we attempt to compute these posterior probabilities of each sample.

Local and Global Consistency (LGC) algorithm [16] is one of the most popular semi-supervised learning algorithms. The core idea of LGC is to treat the labeled samples as the source of (posterior) information on graph and then to propagate the information on graph via iteration. Let us first define an initial posterior probability matrix $F_T \in R^{l\times C}$ as $(F_T)_{ij} = P(\omega_j = i|x_j) = 1$ if sample x_j is labeled to class i; otherwise $(F_T)_{ij} = 0$. The LGC algorithm is summarized in Table 2, where $A = \{A_{ij}|\, i,j = 1..l\}$ is the adjacency matrix and edges are weighted by Gaussian kernel and $\alpha \in (0,1)$ is the learning rate parameter. In the first iteration $\tilde{F} = F_T$.

The LGC algorithm is easy to be implemented and has achieved success in many applications. However, when the class distributions of different classes overlap or there are a large number of ambiguous samples filling the gap between different classes, the performance of LGC deteriorates rapidly. As a example, Fig. 2 shows a synthetic data and LGC result.

Note that there are many ambiguous points between the two classes and a large number of samples are misclassified by LGC due to the incorrect propagation through these ambiguous samples. We improve LGC from two aspects:

(1) The learning rate α in LGC is a global parameter which does not consider the sample distribution difference for different samples. For example in Fig. 2(a), samples B and C are in the inner-class region where the sample distribution is very dense and sample A lies in the inter-class region where the sample distribution is very sparse. The learning rate corresponding to sample A should be small and thus the information propagation over A is slow, so that the incorrect propagation can be effectively suppressed. In contrast, the learning rate corresponding to B and C should be large so that the information can be propagated rapidly and sufficiently for the inner-class region. To achieve this goal, we use a diagonal learning rate matrix $I_{rate} \in R^{n\times n}$ instead of a single parameter α. The diagonal element $(I_{rate})_{ii}$ is the learning rate corresponding to sample i. The method to determine I_{rate} and the convergence proof will be described in Section 5.

(2) While the labeled samples are generally difficult to obtain, the must-link and cannot-link constraint pairs are relatively easy to be retrieved by side information [22,23]. If we write the must-link constraint set as $\mathcal{M}$ and cannot link constraint set as $\mathcal{C}$, then we can use them as follows: if a sample pair $(x_i,x_j) \in \mathcal{M}$, then $W_{ij} = 1$ and if $(x_i,x_j) \in \mathcal{C}$, then $W_{ij} = 0$. Since the Gaussian kernel value is 1 only when the distance between two samples is 0, this is equivalent to merge x_i and x_j together if $(x_i,x_j) \in \mathcal{M}$. A similar argument can be made while $(x_i,x_j) \in \mathcal{C}$

Now the new improved LGC algorithm is summarized in Table 3.

The relationship between the posterior estimator and the graph-based transductive learning method is that both of them propagate information from labeled samples to unlabeled samples on the graph using a similar idea, but the estimator propagates the non-negative posterior probability while graph-based transductive learning propagates label information on the graph. It should be mentioned that these two kinds of information propagated on the graph can be transformed to each other. Note that if we set all diagonal elements of I_{rate} to α and ignore the constraints, the improved LGC in Table 3 is exactly the same LGC algorithm in Table 2, so the improved LGC is a generalized version of the standard LGC algorithm. Fig. 3 shows the comparison result of original LGC and improved LGC on two synthetic data sets, given one pair of constraints of each class.

Fig. 4 displays the mean and standard deviation of the error rate produced by standard LGC and improved LGC for different

Table 2
Algorithm procedure of LGC.

Steps	
1	Construct full graph $G(X_L,A)$ and compute A
2	Compute $S = D^{-1/2}AD^{-1/2}$
3	Iterate $F = \alpha SF + (1-\alpha)F_T$ until converges
4	Compute class label $\omega_i = \arg\max_{j=1..C} F_{ij}$, $i = 1..l$

Table 3
Algorithm procedure of improved LGC.

Steps	
1	Construct $G(V,A)$ and compute A
2	If $(i,j) \in \mathcal{T}, W_{ij} = 1$; if $(i,j) \in \mathcal{Y}, W_{ij} = 0$
3	Compute mean distance $\overline{d}_i$ from x_i to its K nearest neighbors
4	Compute $S = D^{-1/2}WD^{-1/2}$, $(I_{rate})_{ii} = \exp(-\overline{d}_i/2\sigma^2)$
5	Iterate $F_L = I_{rate}SF_L + (I - I_{rate})F_T$ until converges
6	Normalize $F = G^{-1}F_L$, where $G_{ii} = \sum_{j=1}^{C}(F_L)_{ij}$ and 0 elsewhere
7	Compute class label $\omega_i = \arg\max_{j=1..C} F_{ij}$, $i = 1..n$

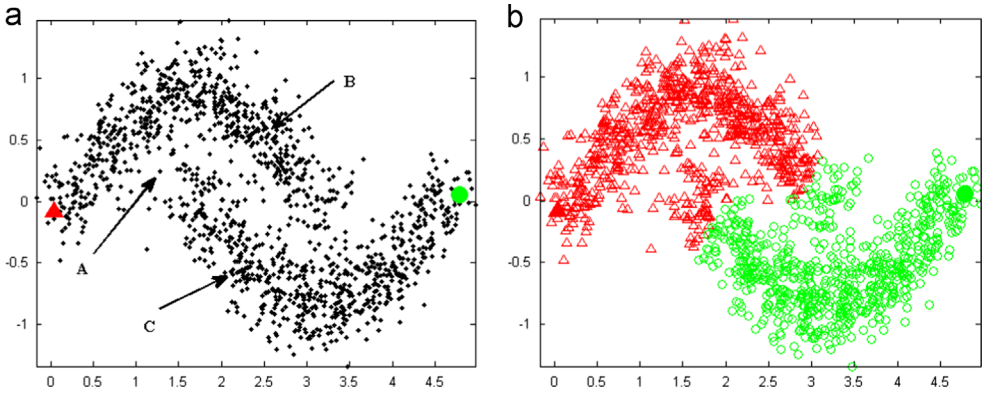


Fig. 2. A example data set with ambiguous samples. (a) the red triangle and green round are labeled samples for two class distributions, respectively. Black dots are unlabeled samples. (b) LGC propagation result.

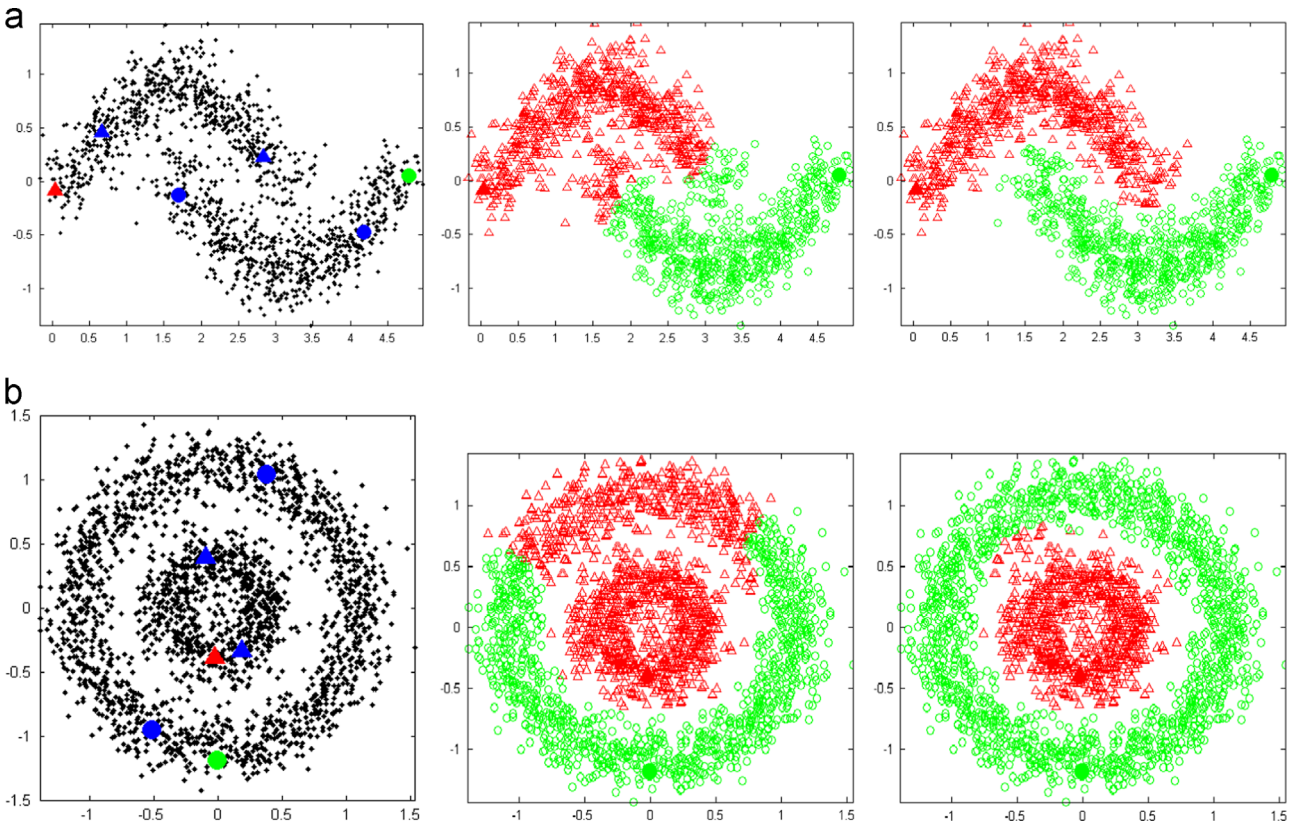


Fig. 3. Comparison of original LGC and improved LGC, Left:the red triangle and green round are labeled samples for two class distributions. Blue samples are must-link constraints pairs. Middle: Result of original LGC in Table 2; right: Result of improved LGC in Table 3. (a) Two moons data set. (b) Two circles data set.

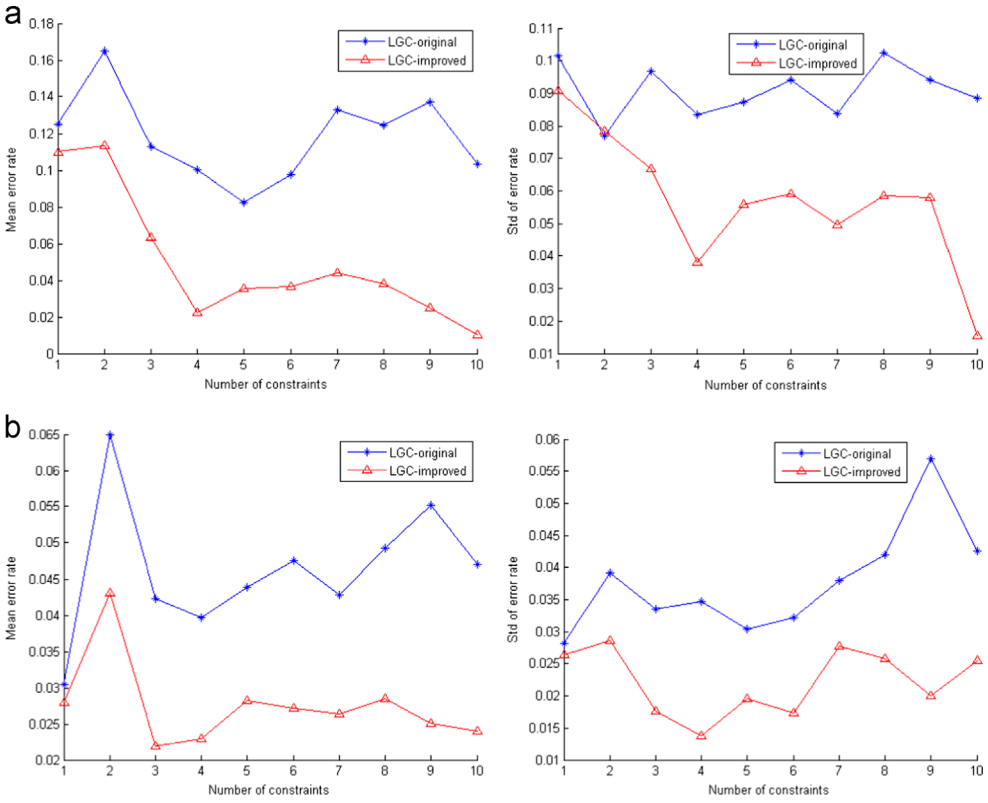


Fig. 4. Error rate comparison of original LGC and improved LGC: (a) The average (left) and standard deviation (right) of 20 runs on two moons data set. (b) The average (left) and standard deviation (right) of 20 runs on two circles data set. Note that there are many ambiguous points between the classes in both data sets.

pairs of constraints over 10 runs. The horizontal axis represents the number of constraints and the vertical axis represents the mean of the error rate (in the two figures on the left) or the standard deviation of the error rate (in the two figures on the right). From these results we can see that the improved LGC can achieve much better result, in terms of accuracy and stability, than standard LGC on data sets with much inter-class ambiguous samples.

4. Density regressor: a robust multivariate nonlinear model to regress the posterior distribution

We model the posterior probability density function of class i by $p(\omega=i|x)=w_i^T\varphi_i(x)+b_i$, where $\varphi_i(\cdot)$ is a unknown nonlinear mapping function which maps the input space to the so-called feature space, whose dimensionality are unknown and can be very large (possibly infinite). Thus $F(x)=(p(\omega=1|x), p(\omega=2|x), \ldots, p(\omega=C|x))$. Without loss of generality, we assume that the mapping functions of all classes are same, written as $\varphi(x)$. So $F(x)=W^T\varphi(x)+b$, and $W=(w_1, w_2, \ldots, w_C)^T$ is a matrix whose row dimension is same as that of the feature space and column dimension can be arbitrary large. Note that $F(x)$ is a multivariate vector-valued function and $F(x_j)=(P(\omega_j=1|x_j), P(\omega_j=2|x_j), \ldots, P(\omega_j=C|x_j))$ for $x_j\in X_L$.

The LS-SVM model is proposed in [24] and has been demonstrated to be a robust and efficient method for nonlinear function regression. But it was proposed to regress a real value function and thus suitable for binary classification problem only. Here we extend the LS-SVM to regress an arbitrary vector-valued nonlinear function and thus can be applied to multi-class classification problem. Specifically, we compute $F(x)$ by solving the following optimization problem:

$$\begin{aligned}&\min \quad J(W,b,E)=\frac{1}{2}\|W\|^2+\frac{1}{2}\gamma\sum_{j=1}^{l}v_j\|r_j\|^2\\&\text{s.t.} \quad F(x_j)=W^T\varphi(x_j)+b+r_j, \quad j=1..l\end{aligned} \tag{1}$$

where $||\cdot||$ is the Frobenius norm. $E=(r_1, r_2, \ldots, r_l)\in R^{C\times l}$ is the error matrix and γ is a regularization parameter. The upper part in Eq. (1) is the objective function and the lower part contains l constraint equations. The first term in the objective function is the regularization term which prevents the model from degeneration [25,26] and the second term is the fitting term which aims to minimize the residual value r_j between true function value $W^T\varphi(x)+b$ and estimated value $F(x)$, as described in the constraint equations. It is worth noting that because the mapping function is unknown and the dimensionality of the feature space can be arbitrary large, problem (1) is quite different from ridge regression (or linear regression or Tikhonov regularization) and cannot be solved directly in the primal form. $v=(v_1, v_2, \ldots, v_l)$ is a non-negative weight vector to reduce the sensitivity of the sum of squared error (SSE) to noise and outliers. A larger weight means that we are more confident that the estimated function value at this point is close to the true function value, because given a large v_j, even the residual value r_j is small but the product $v_j\|r_j\|$ will be large, which means that we put a stronger penalty on the deviation of estimated function value $F(x_j)$ from true function value $W^T\varphi(x_j)+b$ at x_j. Method to obtain v in our algorithm framework will be given in Section 5. In some tasks the weight might not be easy to determine and thus in these cases one can also simply set v_k to 1, which means that all the residual values are equally treated in the function model.

Table 4
Algorithm procedure of density regressor.

Steps	
1	Input X_L, F_L, $v=(v_1, v_2, \ldots, v_l)$, σ and the new point x
2	Compute K, $K_{ij}=\exp(-||x_i-x_j||^2/2\sigma^2)$, $i,j=1..l$
3	Construct $V=\mathrm{diag}(v)$ and $e=(1,1,\ldots,1)^T\in R^l$
4	Solve Eq. (6) for b and Λ
5	Compute $\hat{K}$, $\hat{K}_i=\exp(-||x_i-x||^2/2\sigma^2)$, $i=1..l$
6	Compute function value $F(x)$ at x by Eq. (7)

The corresponding Lagrange of (1) is

$$L(W,b,E,\Lambda)=\frac{1}{2}\|W\|^2+\frac{1}{2}\gamma\sum_{j=1}^{l}v_j\|r_j\|^2+\sum_{j=1}^{l}\lambda_j^T(F(x_j)-W^T\varphi(x_j)-b-r_j) \tag{2}$$

where $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_l)\in R^{c\times l}$ is the Lagrange multiplier matrix. It can be shown that problem (1) is a convex optimization problem [27]. According to the KKT conditions, the optimal solution is given by

$$\begin{cases}W=\sum_{j=1}^{l}\varphi(x_j)\lambda_j^T\\ \sum_{j=1}^{l}\lambda_l=0\\ \gamma v_j r_j-\lambda_j=0, \quad j=1..l\\ F(x_j)-W^T\varphi(x_j)-b-r_j=0, \quad j=1..l\end{cases} \tag{3}$$

With some algebra manipulations, Eq. (3) can be rewritten more concisely as

$$\begin{cases}W=\Psi\Lambda^T\\ \Lambda e=0\\ E=\frac{1}{\gamma}\Lambda V\\ Y=W^T\Psi+be^T+E\end{cases} \tag{4}$$

where $\Psi=(\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_l))$. Eliminating W and E, we have

$$\begin{cases}\Lambda e=0\\ Y=\Lambda\Psi^T\Psi+be^T+\frac{1}{\gamma}\Lambda V\end{cases} \tag{5}$$

where V is a diagonal matrix with vector v on the diagonal line. Eq. (5) is equivalent to solve the following linear systems:

$$\begin{bmatrix}0 & e^T\\ e & K+\eta V\end{bmatrix}\begin{bmatrix}b^T\\ \Lambda^T\end{bmatrix}=\begin{bmatrix}0\\ F_L^T\end{bmatrix} \tag{6}$$

where K is a kernel matrix $K_{ij}=\varphi(x_i)^T\varphi(x_j)$ and $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_l)\in R^{c\times l}$ is the Lagrange multiplier matrix. $e=(1,1,\ldots,1)^T\in R^l$. After obtaining b and Λ, we can evaluate $F(x)$ at any point x by

$$F(x)=\Lambda\hat{K}+b \tag{7}$$

where $\hat{K}=(\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_l))^T\varphi(x)$. Note that to compute $F(x)$ one only needs to know $\varphi(x)^T\varphi(x)$ even the concrete form of $\varphi(x)$ is unknown. We use Gaussian kernel $K_{ij}=\varphi(x_i)^T\varphi(x_j)=\exp(-||x_i-x_j||^2/2\sigma^2)$, where σ is the kernel width. The procedure of the algorithm is listed in Table 4. We will demonstrate the ability of the model to regress a nonlinear vector-valued function in the experimental section.

Steps 2–4 are the training process and the main computational cost lies in solving Eq. (6), but which is a linear equation and can be solved very efficiently using some well optimized linear algebra software library such as LAPACK.[2] Besides, since the training

[2] http://www.netlib.org/lapack/

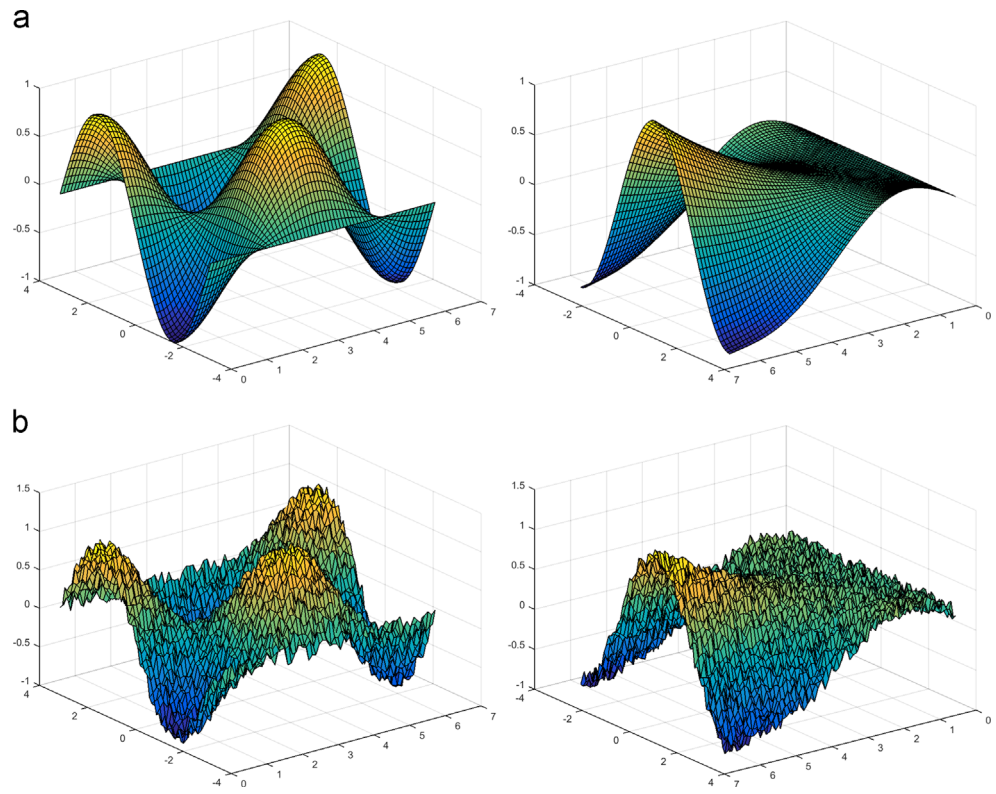


Fig. 5. Example of fitting a nonlinear multivariate vector-valued function. (a) Ground-truth function value; (b) noise contaminated function value (absolute value of the difference between ground truth value and regression value). Left: the first component function, right: the second component function (normalized to $[-1\ 1]$).

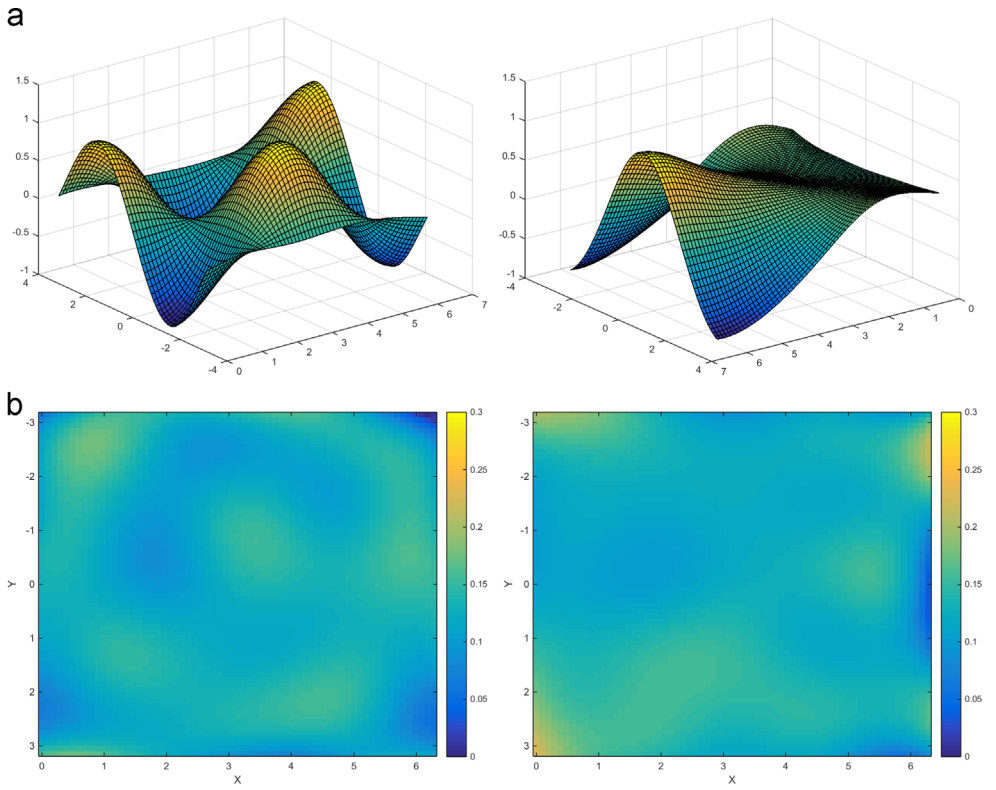


Fig. 6. Results of fitting a nonlinear multivariate vector-valued function. (a) The fitting function value; (b) fitting error image. Left: the first component function, right: the second component function.

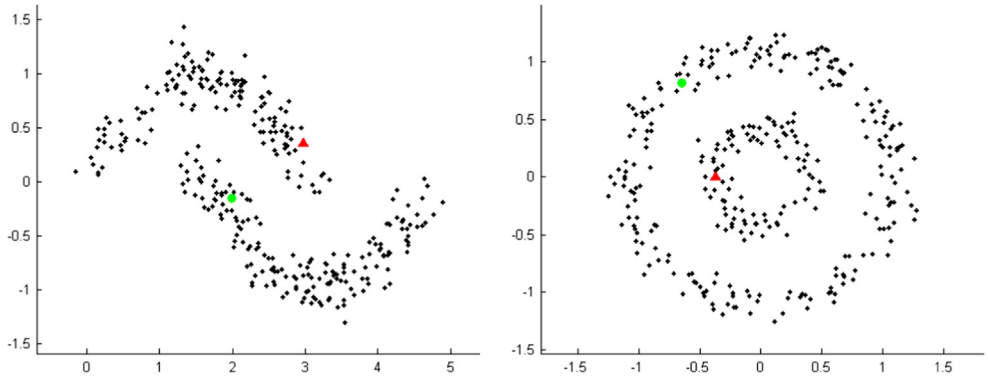


Fig. 7. Two synthetic data sets. The black dots are unlabeled samples. The green rounds and the red triangles are labeled samples, one for each class.

process only performed once, the overall computational burden is very limited. Steps 5 and 6 are used to predict the function value at a given point x and the matrix multiplication in Eq. (7) dominates the computational cost and this multiplication can also be computed very efficiently with the software library.

5. The propagation rate matrix I_{rate} and the weight vector v

The propagation rate matrix I_{rate} controls the speed of the posterior probability propagation on each sample and the convergence of the propagation algorithm. It has to be designed so that the propagation speed is large in the dense sample region while small in the sparse sample region, because sparse region usually means boundaries or overlapping region of classes and low speed can effectively suppress incorrect propagation. Here I_{rate} is computed by $(I_{rate})_{ii} = \exp\left(-\overline{d}_i^2/2\sigma^2\right)$, where $\overline{d}_i$ is the mean distances between x_i and its N nearest neighbors (N is empirically set to 20).

We now show that the propagation algorithm converges. After the ith iteration, we have

$$\tilde{F}^{(i)} = A_{sc}F_T + (I - I_{rate})A_{nb}F_T \tag{8}$$

where $A_{sc} = (I_{rate}S)^{i-1}$ and $A_{nb} = \sum_{k=0}^{i}(I_{rate}S)^k$, representing the posterior information amount acquired from sources and neighborhood, respectively. Let $P = D^{-1}A$ be the random walk matrix on the graph. Then for the spectral radius of P we have $\rho(P) \le \|P\|_\infty \le 1$. Because $S = D^{1/2}PD^{-1/2}$, S is similar to P, so $\rho(S) = \rho(P) \le 1$. Since the spectral radius $\rho(I_{rate}) < 1$, and $\rho(AB) \le \rho(A)\rho(B)$ (this results from the property of matrix norm and for symmetric matrix the spectral radius is equal to spectral norm), $\rho(I_{rate}S) < 1$. Therefore $\lim_{i\to\infty}A_{sc} = 0$, $\lim_{i\to\infty}A_{nb} = (I - I_{rate}S)^{-1}$ and

$$F_L = \lim_{i\to\infty}\tilde{F}^{(i)} = (I - I_{rate})(I - I_{rate}S)^{-1}F_T \tag{9}$$

For this optimal solution, we have the following proposition:

Proposition 1. *For a connected graph with nonnegative edge weights, all entries of the optimal solution in Eq. (9) are strictly positive.*

To prove it, we need the following lemma:

Lemma 1. *The multiplication of (1) a full row rank nonnegative matrix and a strictly positive matrix or (2) a strictly positive matrix and a full column rank nonnegative matrix is still a strictly positive matrix.*

Now let us prove them.

Proof of Lemma 1. We prove this using contradiction. Without loss of generality, we just prove the first situation. Let $A = BC$, where B is full row rank nonnegative and C is strictly positive. Assume that the (i,j) entry of A is 0, i.e. $A_{ij} = B_{i\cdot}C_{\cdot j} = 0$, where $B_{i\cdot}$ is the ith row of matrix B and $C_{\cdot j}$ is the jth column of matrix C, then all entries of $B_{i\cdot}$ should be 0, because every entry of B is nonnegative and every entry of C is positive. Then the matrix B is not a full row rank matrix, because its entries on the ith row are all 0. This contradicts with our assumption that B is full row rank. □

Proof of Proposition 1. Because I_{rate} is a full rank matrix and S is a nonnegative matrix, so from Lemma 1 we know that $A_{nb} = \sum_{k=0}^{i}(I_{rate}S)^k$ is a strictly positive matrix. By the construction method, all entries of F_T are nonnegative and F_T has full column rank, again from Lemma 1 we know that F^* is strictly positive. □

Because the strict positivity of F_L, after normalization, F has the posterior probability meaning, i.e. the entry of F_{ij} gives the posterior probability $P(\omega_j = i|x_j)$. We then apply the logarithmical transformation to it before density regressor. This logarithmical transformation is helpful because activation spreading on networks is scale free distributed [28,29], and thus $\log(F)$ is expected to be approximately piece-wise linear that can be fitted much easier and more accurately.

The weight vector of x_j is set to $v_j = \max_{i=1\ \ldots\ C}F(x_j) - \max_{k=1\ \ldots\ C,\ k\neq i}F(x_j)$, the difference between its largest and the second largest posterior probability. Because equal posterior probabilities indicate that the sample is ambiguous to all classes and thus the posterior probability is less informative and unreliable, so the weight should be small.

6. Experimental results

6.1. Regression experiments

To verify the regression ability of the proposed model, we construct arbitrarily the following two-component nonlinear function:

$$F(x,y) = \left(\sin(x)\cos(y),\ \sin(x)y^2 + \cos(y)x^2\right)$$

We sample 65 points for x from interval $(0, 2\pi)$ and 65 points for y from interval $(-\pi, \pi)$), so there are total 4225 points. Its ground truth two components are plotted in Fig. 5(a) and the noisy contaminated components are in Fig. 5(b) (Gaussian noise with standard deviate 0.5).

We randomly select 10% of total points to train the model in Eq. (1) and then predict all the points using Eq. (7). The regression results are shown in Fig. 6. From these results we can see that even each component of the function is highly nonlinear and noise-

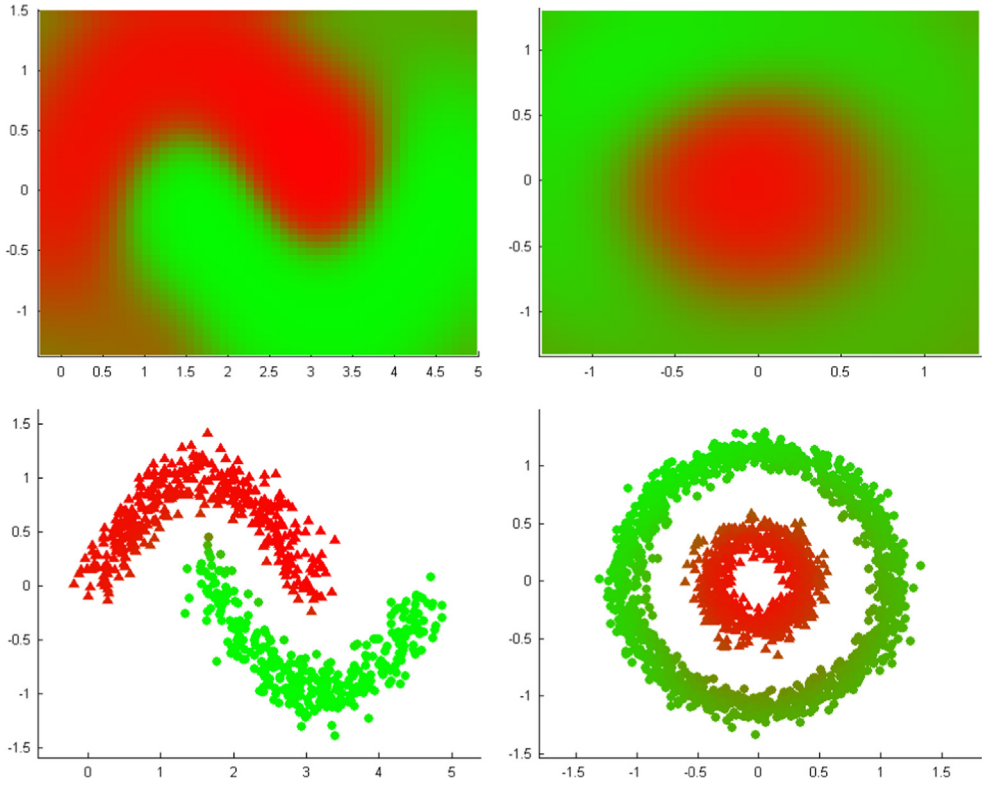


Fig. 8. Experimental results of PDL. Top: posterior function $F(x)$ learnt in the input space; bottom: classification result of out-of-sample data generated independently.

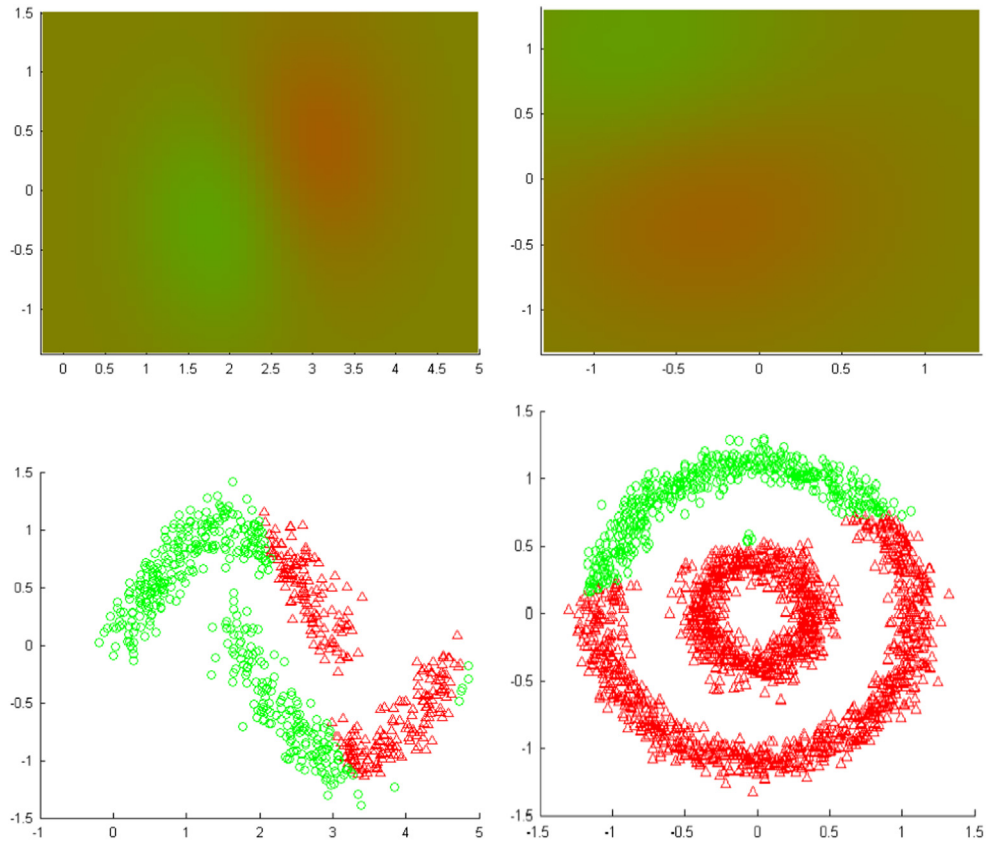


Fig. 9. Experimental results of SVM. Top: posterior function $F(x)$ learnt in the input space; bottom: classification result of out-of-sample data generated independently.

contaminated, the ground truth function can be well fitted by the model with only 10% of points.

6.2. Classification experiments

We have seen the propagation ability of improved LGC in Section 3 and the regression ability of the extended LS-SVM in Section 6.1. In this section we report the generalization ability of PDL on synthetic data set and real world data sets and the comparison results of the proposed PDL framework with traditional supervised learning method and the state-of-the-art algorithms.

6.2.1. Synthetic data sets

We conduct experiments on two synthetic data sets: two-moon data set and two-circle data set to compare the generalization ability of PDL with SVM, which is one of the most widely used traditional supervised learning algorithms.

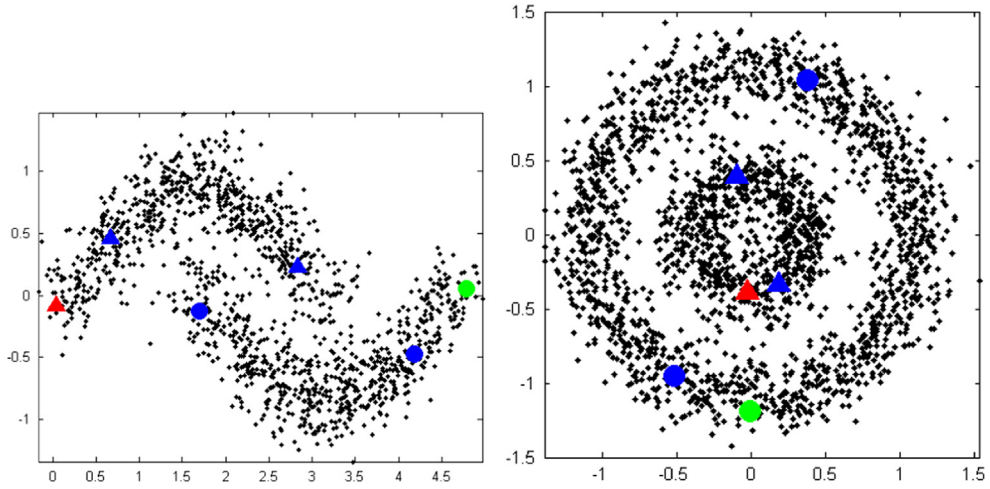


Fig. 10. Two synthetic data sets with inter-class noisy points. The black dots are unlabeled samples. The green rounds and the red triangles are labeled samples, one for each class. The two blue triangles and two blue rounds are pairs of must-link constraints.

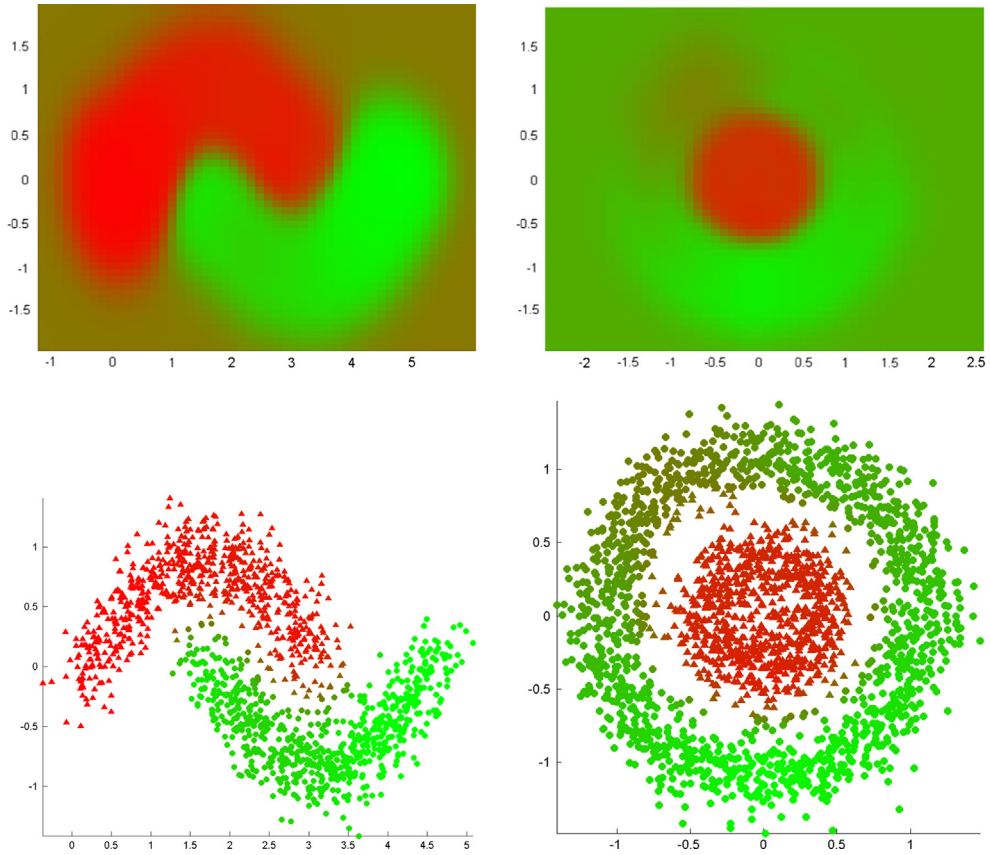


Fig. 11. Experimental results of PDL. Top: posterior function $F(x)$ learnt in the input space; bottom: classification result of out-of-sample data generated independently.

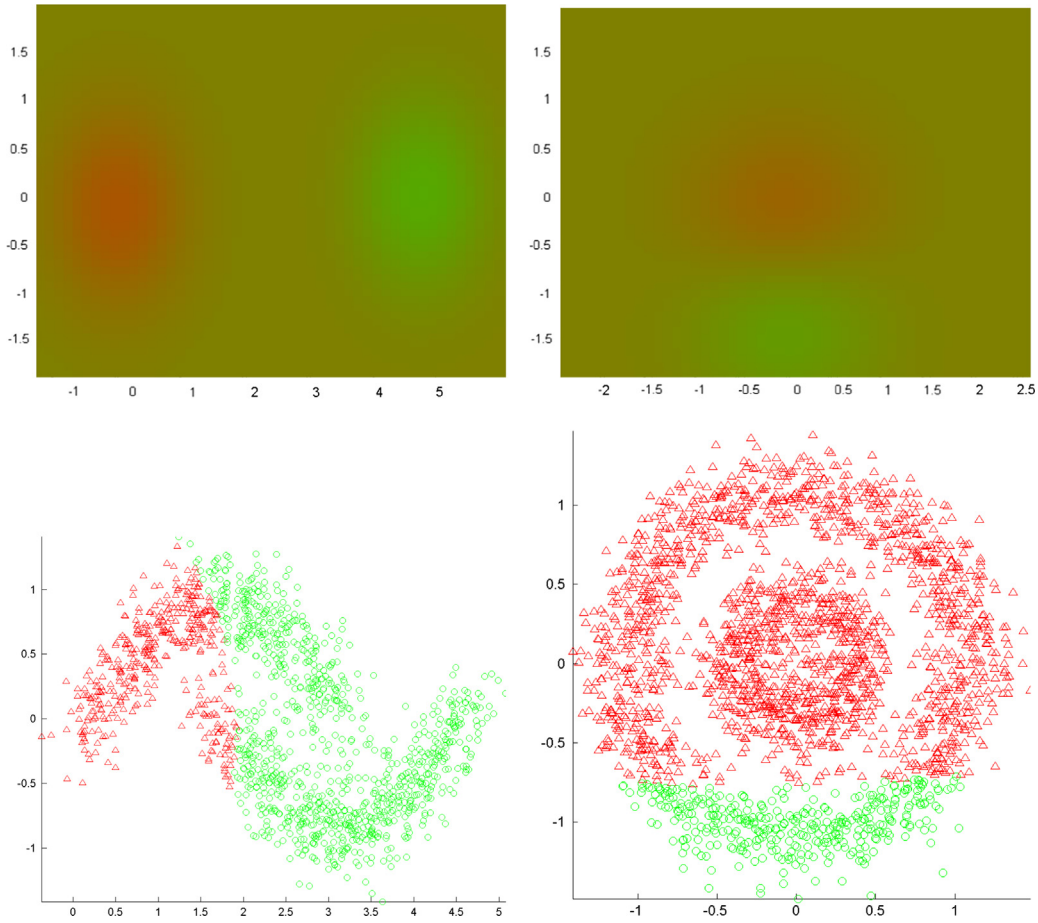


Fig. 12. Experimental results of SVM. Top: posterior function $F(x)$ learnt in the input space; bottom: classification result of out-of-sample data generated independently.

(a) *Clean data sets without ambiguous samples*: We first carry out experiments on two clean data sets, which are shown in Fig. 7 (the black dots are unlabeled samples and the color shapes are labeled samples, one labeled sample for each class).

In this experiment, we use only one labeled samples for each class and do not use any pairwise constraints. The experimental results of PDL are shown in Fig. 8 and the results of SVM in Fig. 9.

From these results we can see that given a very few labeled samples and many unlabeled samples, PDL can learn a distribution function well covering the whole data distribution and build a supervised model with a very good generalization ability. It is worth noting that PDL can also capture the underlying manifold characteristics of the data distribution and thus shows the potential to perform manifold classification. In contrast, traditional supervised learning method fails to obtain a reliable supervised model because the labeled samples contain very limited information to train the model with good generalization.

(b) *Noisy data set with ambiguous samples*: In this experiments, the two data sets contain many ambiguous samples, which could affect the classification results significantly, as shown in Fig. 10.

We use one pair of must-link constraint for each class. The result of PDL is shown in Fig. 11 and the result of SVM is shown in Fig. 12.

For SVM, we tune the kernel width parameter $\sigma = 0.1$ to produce best results. From these results we can see that even with one pair of constraints, the results can be improved considerably when ambiguous points are present.

Table 5
Information of the experimental data sets.

	usps	segmentation	banknote	pendigits	skin	miniBoo	statlog
n	9298	2086	1348	10 992	245 057	130 064	6435
d	256	19	4	16	3	50	36
C	10	6	2	10	2	2	6

6.2.2. Real world data sets

We conduct experiments on 7 real world data sets.[3] The information of the data sets is listed in Table 5 (n: number of samples; d: feature dimension; C: class number). Each data set is randomly split into X_L and X_U. The size of X_L and X_U are $0.1n$ and $0.9n$, respectively, for data set *skin* and *miniBoo*, and $0.7n$ and $0.3n$ for other data sets.

(a) *Comparison with traditional supervised learning method*: The baseline algorithms are support vector machine (SVM), k nearest neighbors (kNN), artificial neural networks (ANN), Nave Bayes (NB) and decision tree (DT). We use grid-search method to determine the optimal parameters of each algorithm on each data set. The configurations of the baseline algorithms are radial basis kernel for SVM; three-layer back-propagation networks with $\lceil d/2 \rceil_{10}$ neurons in the hidden layer for ANN, where $\lceil x \rceil_{10}$ is the smallest integer greater than or equal to the $\max(x, 10)$; Euclidean distance is utilized and k varies from 5 to 10 for kNN; For Nave

[3] http://archive.ics.uci.edu/ml/

Table 6
Classification error rate (%) of real world data set.

labelled samples per class		usps	segment	banknote	pendigits	skin	miniBoo	statlog
1	SVM	97.56 ± 0.9	86.68 ± 2.3	61.85 ± 9.3	96.82 ± 1.8	83.56 ± 4.4	62.54 ± 15.9	89.00 ± 0.4
	*k*NN	58.76 ± 3.9	59.09 ± 6.1	44.77 ± 2.0	70.72 ± 5.0	71.53 ± 0.1	71.85 ± 0.0	77.09 ± 4.4
	ANN	64.14 ± 4.2	**42.28 ± 4.1**	44.25 ± 18.5	44.69 ± 4.4	47.44 ± 7.2	42.53 ± 15.6	37.86 ± 10.4
	NB	45.85 ± 2.9	80.83 ± 3.9	38.75 ± 13.1	68.00 ± 4.0	83.40 ± 6.2	90.93 ± 0.1	42.04 ± 10.0
	DT	83.95 ± 0.6	71.21 ± 1.5	44.77 ± 2.0	87.90 ± 2.2	71.53 ± 0.1	71.85 ± 0.0	76.18 ± 0.8
	PDL	**37.76 ± 10.9**	43.29 ± 5.5	**14.32 ± 11.9**	**24.07 ± 6.2**	**30.99 ± 5.1**	**37.24 ± 5.5**	**31.77 ± 11.9**
3	SVM	80.5 ± 7.8	71.26 ± 8.2	62.22 ± 23.8	67.42 ± 14.9	55.43 ± 24.3	54.69 ± 22.65	57.20 ± 18.2
	*k*NN	40.11 ± 2.9	35.30 ± 5.0	25.83 ± 8.3	30.60 ± 4.0	40.47 ± 12.4	21.93 ± 4.9	33.50 ± 6.6
	ANN	38.83 ± 3.7	29.19 ± 4.7	13.75 ± 8.4	27.28 ± 3.2	25.18 ± 7.1	24.14 ± 3.7	27.44 ± 4.8
	NB	33.96 ± 3.7	48.08 ± 12.1	21.90 ± 5.8	48.87 ± 7.8	26.34 ± 13.0	83.98 ± 18.2	30.31 ± 4.1
	DT	71.30 ± 5.7	61.73 ± 5.5	45.04 ± 2.6	77.77 ± 11.5	71.49 ± 0.1	71.85 ± 0.0	74.35 ± 8.6
	PDL	**23.71 ± 6.7**	**24.33 ± 8.7**	**5.01 ± 4.3**	**12.34 ± 2.0**	**14.80 ± 8.6**	**18.64 ± 0.5**	**19.47 ± 1.7**
5	SVM	38.46 ± 9.2	45.78 ± 11.4	24.57 ± 24.8	19.33 ± 5.2	**6.71 ± 2.5**	45.22 ± 27.7	23.64 ± 3.9
	*k*NN	28.92 ± 1.9	28.56 ± 3.7	21.83 ± 8.7	23.17 ± 3.7	25.83 ± 6.5	20.61 ± 1.9	25.49 ± 5.5
	ANN	28.80 ± 2.2	29.20 ± 6.5	8.27 ± 4.6	21.52 ± 2.5	18.88 ± 8.5	23.89 ± 3.8	21.22 ± 3.2
	NB	27.47 ± 2.4	34.60 ± 6.9	21.14 ± 6.7	35.13 ± 5.4	29.45 ± 12.5	30.68 ± 11.5	24.80 ± 5.7
	DT	70.03 ± 9.8	42.43 ± 22.8	22.57 ± 8.0	63.81 ± 12.6	32.15 ± 10.2	26.57 ± 6.3	48.28 ± 18.9
	PDL	**19.53 ± 1.2**	**20.59 ± 7.3**	**4.59 ± 4.5**	**9.37 ± 2.6**	9.81 ± 3.4	**18.26 ± 0.2**	**16.77 ± 1.9**

Bold values means the minimal error rate.

Table 7
Comparison of PDL with the state-of-the-art algorithm for classification error rate (%) of real world data set: X_U.

labelled samples per class		usps	segment	banknote	pendigits	skin	miniBoo	statlog
1	VLR	50.94 ± 5.2	**31.68 ± 8.6**	22.47 ± 9.6	43.65 ± 6.2	35.34 ± 19.8	**32.33 ± 19.1**	41.75 ± 10.0
	TriT	76.62 ± 10.4	87.96 ± 10.0	43.06 ± 18.2	77.56 ± 6.7	32.65 ± 17.5	67.76 ± 12.9	47.79 ± 11.4
	MR	**22.88 ± 7.1**	33.23 ± 17.9	35.06 ± 7.5	28.05 ± 11.4	–	–	45.54 ± 5.3
	PDL-A	34.98 ± 3.2	41.98 ± 30.4	14.37 ± 11.7	38.90 ± 18.9	**30.83 ± 23.9**	33.30 ± 22.5	34.76 ± 12.4
	PDL-B	37.76 ± 10.9	43.29 ± 5.5	**14.32 ± 11.9**	**24.07 ± 6.2**	30.99 ± 5.1	37.24 ± 5.5	**31.77 ± 11.9**
3	VLR	41.30 ± 4.4	26.10 ± 3.2	5.21 ± 5.2	29.95 ± 2.8	22.03 ± 5.1	23.08 ± 8.4	28.03 ± 4.2
	TriT	52.57 ± 6.6	34.62 ± 8.2	27.31 ± 14.8	35.72 ± 5.0	32.29 ± 13.8	22.91 ± 6.0	40.91 ± 6.7
	MR	**10.93 ± 0.9**	31.90 ± 8.5	15.14 ± 4.5	12.96 ± 4.1	–	–	25.36 ± 1.9
	PDL-A	29.83 ± 5.9	29.07 ± 9.8	**4.79 ± 3.8**	23.43 ± 4.1	15.33 ± 8.5	18.78 ± 0.7	19.64 ± 3.0
	PDL-B	23.71 ± 6.7	**24.33 ± 8.7**	5.01 ± 4.3	**12.34 ± 2.0**	**14.80 ± 8.6**	**18.64 ± 0.5**	**19.47 ± 1.7**
5	VLR	35.94 ± 2.1	25.77 ± 4.6	**3.58 ± 2.5**	26.97 ± 2.6	18.81 ± 2.8	**17.35 ± 0.8**	27.50 ± 5.8
	TriT	41.15 ± 6.5	32.25 ± 4.9	17.06 ± 8.0	23.70 ± 4.7	28.09 ± 13.6	21.00 ± 11.0	23.27 ± 3.4
	MR	**9.25 ± 0.8**	**18.85 ± 2.3**	9.47 ± 9.8	**7.60 ± 2.9**	–	–	25.00 ± 3.5
	PDL-A	27.69 ± 2.4	29.79 ± 25.4	4.35 ± 4.4	18.54 ± 2.8	10.54 ± 3.4	18.25 ± 0.4	17.85 ± 2.4
	PDL-B	19.53 ± 1.2	20.59 ± 7.3	4.59 ± 4.5	9.37 ± 2.6	**9.81 ± 3.4**	18.26 ± 0.2	**16.77 ± 1.9**

Bold values means the minimal error rate.

Bayes we use kernel smoothing density estimation to model the features and the kernel width is automatically selected for each combination of feature and class, using a value that is optimal for a Gaussian distribution; For Decision Tree we construct a binary classification tree with 10 maximum category levels and a merging-pruning strategy is adopted according to validation error. At each time, we randomly select 1, 3 and 5 samples of each class from X_L as the labeled samples to form training set X_T. Because baseline algorithms cannot use any unlabeled sample in training, so they are trained with X_T. In contrast, PDL builds a model using X_L, which contains both the labeled and unlabeled samples. After all classifiers are trained, we use X_U as a out-of-sample data set to evaluate the generalization ability of each classifier and calculate the error rate by counting how many samples in X_U are misclassified. Each algorithm runs 10 times and the final result is the average error rate. We adopt the grid-search strategy for each data set to tune the parameters of each algorithm and the parameters producing lowest error rate is selected to be the optimal parameters of the data set used in the experiments.

Experimental results are shown in Table 6. The first column is the number of labeled samples from each class. Each table cell gives the corresponding mean and standard variance of the error rate.

From these results we can see that the error rate of PDL can be up to 57% lower than SVM on data set *usps* given 3 labeled samples per class and up to 25% lower than SVM on data set *segmentation* given 5 labeled samples per class. Furthermore, for 1 training sample per class, SVM performs much worse than others. As the number of training samples increases, its error rate decreases sharply, especially when the number of training sample per class is 5, but the variance of the error rate is also very large. This indicates that not only the quantity of the labeled samples, but also the quality of the labeled samples affects the performance considerably. Because when randomly selected, the position of the labeled samples varies greatly in the data space. Sometimes they do not uniformly distribute and thus cannot well cover the whole data distribution. As a result, the model cannot estimate the true data distribution correctly during training and the performance can be very poor. This is also true for other supervised classifiers. In contrast, by taking the unlabeled samples into training stage, PDL significantly outperforms the traditional supervised learning algorithms given very limited training samples, because the posterior propagation algorithm performs a kind of graph diffusion and learns on manifold [30], so even a few and not well positioned labeled samples can be utilized correctly.

Table 8
Comparison of PDL with the state-of-the-art algorithm for classification error rate (%) of real world data set: X_L.

labelled samples per class		usps	segment	banknote	pendigits	skin	miniBoo	statlog
1	VLR	47.48 ± 5.1	**32.03 ± 5.3**	17.40 ± 11.3	43.77 ± 5.5	29.01 ± 8.0	27.10 ± 13.1	44.37 ± 8.1
	TriT	78.53 ± 12.1	81.13 ± 13.9	31.74 ± 11.7	73.95 ± 4.9	34.28 ± 14.0	67.21 ± 13.8	48.17 ± 12.2
	MR	**24.83 ± 7.8**	36.07 ± 8.7	37.26 ± 14.0	**15.21 ± 0.9**	70.62 ± 0.7	40.63 ± 29.5	51.52 ± 7.5
	PDL-A	35.54 ± 7.1	34.36 ± 24.6	**14.03 ± 12.2**	45.50 ± 28.5	21.11 ± 14.9	**23.33 ± 4.2**	41.22 ± 6.2
	PDL-B	36.76 ± 10.4	37.58 ± 4.4	14.09 ± 12.2	21.96 ± 4.8	**20.90 ± 5.2**	36.60 ± 4.4	**39.63 ± 7.0**
3	VLR	38.52 ± 2.9	**23.97 ± 3.0**	**4.11 ± 3.0**	28.11 ± 3.4	24.63 ± 11.6	23.19 ± 6.1	29.72 ± 6.4
	TriT	56.80 ± 8.4	36.82 ± 10.6	26.00 ± 11.6	35.83 ± 13.2	27.94 ± 0.5	**23.06 ± 3.0**	38.69 ± 7.7
	MR	**11.60 ± 2.5**	**20.80 ± 1.2**	19.51 ± 8.0	**8.33 ± 1.8**	67.42 ± 2.4	26.08 ± 5.4	30.31 ± 4.9
	PDL-A	27.28 ± 5.2	24.48 ± 10.7	5.62 ± 5.0	20.68 ± 3.5	13.37 ± 3.2	24.96 ± 16.6	28.26 ± 12.1
	PDL-B	23.50 ± 5.0	25.30 ± 4.2	5.64 ± 5.0	10.89 ± 2.1	**12.86 ± 3.4**	25.02 ± 6.3	**21.21 ± 6.2**
5	VLR	34.57 ± 3.0	24.91 ± 3.9	2.64 ± 1.9	25.95 ± 2.3	19.10 ± 5.7	21.14 ± 9.3	27.98 ± 5.2
	TriT	43.70 ± 7.6	27.89 ± 6.0	11.98 ± 3.5	23.22 ± 3.9	30.98 ± 14.8	**18.98 ± 5.4**	24.11 ± 2.7
	MR	**9.19 ± 1.6**	22.90 ± 5.4	7.07 ± 0.6	**6.94 ± 1.3**	45.24 ± 24.1	22.31 ± 2.9	24.19 ± 5.0
	PDL-A	22.29 ± 1.2	32.14 ± 24.5	0.94 ± 0.7	18.72 ± 2.6	11.29 ± 2.7	22.25 ± 9.1	18.02 ± 3.0
	PDL-B	16.20 ± 2.4	**19.65 ± 6.1**	**0.89 ± 0.7**	9.87 ± 1.9	**10.94 ± 2.9**	23.33 ± 2.8	**17.07 ± 2.4**

Bold values means the minimal error rate.

(b) *Comparison with the state-of-the-art algorithms*: We also compare the proposed PDL with two recently reported algorithms, the Tri-training (TriT) [4] and the virtual label regression (VLR) [31], which also train supervised learning classifiers using both labeled and unlabeled samples. For TriT, same as the paper, the three classifiers are artificial neural network, decision tree and naive Bayes, as they can well exploit the different properties of a data set [4], and their parameters are same as those used in previous subsection because they can generally achieve better results. For VLR, because it uses the linear regression method, so the parameter to be tuned is the jump probability α and the graph edge weight kernel width σ. We set α to its typical value 0.9 and σ is determined using same method described in the paper. Since the Manifold Regularization (MR) [32] semi-supervised learning algorithm can perform manifold classification and achieves state-of-the-art result in many cases, we also include it as a baseline algorithm. The parameter of MR is determined in using the method given in [31], i.e. $\sigma = \tau\sqrt{-\overline{d}/\ln(1/k)}$ where $\overline{d}$ is the overall mean distance of each sample to its K nearest neighbors. As we mentioned in Section 2, our PDL is a very generic learning framework which can build a multiclass supervised model utilizing both labeled and unlabeled samples. That means that both the two components can be replaced by other algorithms. Here we use regression artificial neural network (ANN) for posterior distribution regression and we denote this structure PDL-A. In contrast, we use PDL-B[4] to denote the learning algorithm described in this paper. The ANN in PDL-A has the same configuration as the ANN used in previous subsection.

We generate subsets X_T, X_L and X_U following the same procedure as in the former subsection. All the algorithms are first trained on X_L and then tested on X_U which serves as the out-of-sample data set. The error rate averaged over 10 runs on X_U is shown in Table 7.

We can see that PDL can achieve better results for most of the cases. For tri-training or co-training algorithms, it is easy to introduce noise labels as the training set grows and the impact of these noise labels can be very large. For VLR, it regresses a linear model using the discrete class-indicator vector of each sample, so it can hardly capture the nonlinearity and continuousness of the posterior density functions. Because MR has a large space complexity and due to memory restriction in an ordinary personal computer, the result of MR for data set skin and miniBoo is not given in the table. In contrast, PDL first propagates posterior information from labeled samples to unlabeled samples on graph and then regresses a nonlinear model in the input space. But graph has a tight relationship with the manifold structure [33], thus the underlying manifold information is thus also encoded in the distribution function. Therefore PDL shows a promising potential for classifying manifold-distributed data set. It is worth noting that PDL-B generally outperforms PDL-A. This indicates that in these experiments, the proposed kernel regression method performs better than regression neural network, which is deemed a powerful tool for multivariate regression.

Table 9
Comparison of prediction time on test set X_U (s).

labelled samples per class	VLR	TriT	MR	PDL-A	PDL-B
Time	0.01	2.4	45.83	0.5	8.1

Since all the algorithms are trained on learning set X_L, it is interesting to compare their performance on learning set. Because tri-training also gives the class label while training, so its error rate on X_L can be obtained directly after it is trained. For PDL and VLR, they first learn on X_L to build a model and then X_L is fed to the classifiers again to compute the classification error rate. The experimental results are shown in Table 8.

From Tables 7 and 8, we can draw some conclusions. (1) By including the unlabeled samples into training stage, these supervised learning algorithms which are trained with a very small number of labeled samples have a strong generalization ability. Their performance on learning set and on out-of-sample set are quite similar. This is because that unlabeled samples also contain useful information about the data distribution and this is well recognized in semi-supervised learning community [18,16,17]. So the supervised model that utilizes this information can have a better estimation of the underlying data distribution and, as a result, can predict unseen samples with a higher accuracy. (2) The two PDL algorithms can use the unlabeled samples in a "*more effective*" way. They attain mostly the best results on both learning set and out-of-sample set, especially on X_U data set.

We also compare the time cost of each algorithm on X_U of USPS and the result is shown in Table 9.

Because VLR constructs a linear model in input data space, it is very efficient and its time cost is the lowest one. MR consumes the most time among all algorithms, because its time complexity is polynomial and it builds a binary classifier. For multi-class case, it has to build a more complex classifier under one-against-one or one-against-all architecture, and thus the time cost will also

[4] Softwares of PDL-A and PDL-B are available upon request.

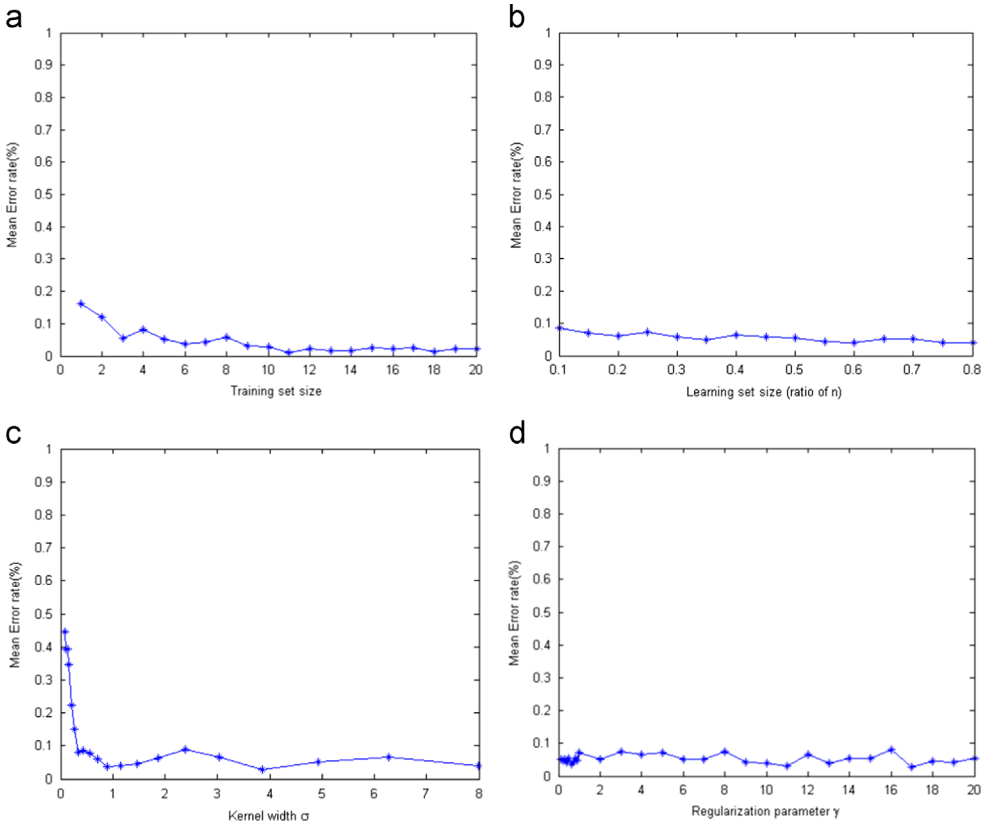


Fig. 13. Experimental results of parameter effect upon classification performance.

increase much. In contrast, the two variants of PDL algorithms achieve a better accuracy with relatively lower time consumption.

6.2.3. Parameter sensitivity

In this section we conduct four experiments on data set *banknote* to investigate the effects of four parameters upon the classification performance: number of labeled samples per class t_c, learning data size l, kernel width σ and regularization parameter γ. In each experiment, we change one parameter and keep other parameter unchanged, i.e., experiment 1: t_c is from 1 to 20 and $l=0.7n$, $\sigma = 0.8$, $\gamma = 5$; experiment 2: l is from $0.1n$ to $0.8n$ and $t_c=5$, $\sigma = 0.8$, $\gamma = 5$; experiment 3: σ is from $0.1\overline{d}$ to $10\overline{d}$ and $t_c=5$, $l=0.7n$, $\gamma = 5$; experiment 4: γ is from 0.1 to 20 and $t_c=5$, $l=0.7n$, $\sigma = 0.8$. Experimental results averaged over 10 runs on out-of-sample data set X_U are shown in Fig. 13.

The experimental results show that (1) PDL can be a stable supervised model with very good generalization ability given a small training set (e.g. 5 training samples per class for this data set). (2) The performance of PDL is rather stable against learning set size, which means that one can use a small part of samples to form the learning set to build the PDL model while the whole data set is large. (3) While small σ has a significant influence upon the performance, large σ generally yields stable and accurate classification results. (4) PDL is rather insensitive to changes in the regularization parameter γ.

7. Discussions and conclusions

In this paper we developed a novel two-step framework, Posterior Distribution Learning (PDL), to combine the advantage of supervised learning and semi-supervised learning to build a robust model in data space using a very few training samples and plenty of unlabeled samples. The framework first propagates posterior information from labeled samples to unlabeled samples and then regresses a multivariate posterior function in the data space. The relationships between PDL and previous learning methods are the following:

(1) *Relationship between PDL and traditional supervised learning method*: Both PDL and traditional supervised learning method build a model in the input data space. Once the model is well trained, it can be applied to any unseen sample to predict its class label. The difference is that while the traditional supervised learning method is a one-stage learning method and it does not consider the information contained in plenty of unlabeled sample, PDL is a two-stage learning method and it takes the unlabeled sample distribution into consideration. The utilization of unlabeled samples has two major benefits: (a) It reduces the quantity and quality demand upon labeled samples to build a robust supervised model and thus reduces human burden to obtain labeled samples, because the traditional supervised model has to be trained with sufficient well labeled samples while PDL can use a small number of labeled samples plus plenty of unlabeled samples to build a robust supervised model. (b) It can provide more data distribution information to train a better supervised model with good generalization ability. The effect of unlabeled samples can be understood in this way: the marginal distribution $p(x)$ also provides useful information to the estimation of the conditional distribution $p(\omega|x)$, because if two points x_1 and x_2 are close for the marginal distribution $p(x)$, then $p(\omega_1|x_1)$ and $p(\omega_2|x_2)$ should be similar. So by taking the unlabeled samples into consideration, the information of $p(x)$ can be exploited by PDL.

(2) *Relationship between PDL and semi-supervised learning (SSL) method*: Both PDL and SSL can use a tiny number of labeled samples plus abundant unlabeled sample to predict the class labels of unlabeled samples. However, PDL is a inductive learning method, i.e. it builds a multiclass model in the input data space and thus it can handle any unseen samples directly. In contrast, most SSL algorithms are the transductive learning method, i.e. they only build

a model in the target data set, so it cannot be applied to any unseen samples [18]. Although there are SSL algorithms, such as [32,34], can perform inductive learning, they are binary classifiers and have cubic computational and spatial complexity in terms of data set size. For multiclass case, one needs to build a more complex multiclass model using one-against-one or one-against-all architecture, so the computational burden and memory requirements to handle the kernel matrix will further increase. This limits its application to large data sets, e.g. millions of samples.

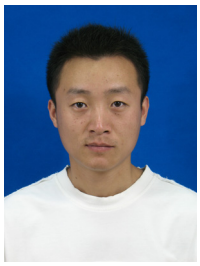
In the following work we will focus on theoretical analysis of the architecture and extending the framework by including other supervised learning techniques.

Acknowledgments

This research is partly supported by NSFC, China (Nos. 61273258, 61403247), 973 Plan, China (No. 2015CB856004).

References

[1] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, ACM, New York, United States, 1998, pp. 92–100.
[2] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using em, Mach. Learn. 39 (2–3) (2000) 103–134.
[3] M.-R. Amini, P. Gallinari, The use of unlabeled data to improve supervised learning for text summarization, in: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, United States, 2002, pp. 105–112.
[4] Z.-H. Zhou, M. Li, Tri-training: exploiting unlabeled data using three classifiers, IEEE Trans. Knowl. Data Eng. 17 (11) (2005) 1529–1541.
[5] F. Denis, PAC learning from positive statistical queries, in: Algorithmic Learning Theory, Springer, New York, United States, 1998, pp. 112–126.
[6] X. Li, B. Liu, Learning to classify texts using positive and unlabeled data, in: IJCAI, vol. 3, 2003, pp. 587–592.
[7] B. Liu, W.S. Lee, P.S. Yu, X. Li, Partially supervised classification of text documents, in: ICML, vol. 2, Citeseer, 2002, pp. 387–394.
[8] W.S. Lee, B. Liu, Learning with positive and unlabeled examples using weighted logistic regression, in: ICML, vol. 3, 2003, pp. 448–455.
[9] M. Chi, L. Bruzzone, A semilabeled-sample-driven bagging technique for ill-posed classification problems, IEEE Geosci. Remote Sens. Lett. 2 (1) (2005) 69–73.
[10] S. Szedmak, J. Shawe-Taylor, Synthesis of maximum margin and multiview learning using unlabeled data, Neurocomputing 70 (7) (2007) 1254–1264.
[11] T. Xia, D. Tao, T. Mei, Y. Zhang, Multiview spectral embedding, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 40 (6) (2010) 1438–1446.
[12] I. Muslea, S. Minton, C.A. Knoblock, Active+ semi-supervised learning=robust multi-view learning, in: ICML, vol. 2, 2002, pp. 435–442.
[13] P. Dhillon, D.P. Foster, L.H. Ungar, Multi-view learning of word embeddings via CCA, in: Advances in Neural Information Processing Systems, 2011, pp. 199–207.
[14] J. Yu, M. Wang, D. Tao, Semisupervised multiview distance metric learning for cartoon synthesis, IEEE Trans. Image Process. 21 (11) (2012) 4636–4648.
[15] V. Sindhwani, D.S. Rosenberg, An RKHS for multi-view learning and manifold co-regularization, in: Proceedings of the 25th International Conference on Machine Learning, ACM, New York, United States, 2008, pp. 976–983.
[16] D. Zhou, O. Bousquet, T.N. Lal, J.Weston, B. Schölkopf, Learning with local and global consistency, in: Advances in Neural Information Processing Systems, vol. 16, 2004, pp. 321–328.
[17] X. Zhu, Z. Ghahramani, J. Lafferty, et al., Semi-supervised learning using Gaussian fields and harmonic functions, in: ICML, vol. 3, 2003, pp. 912–919.
[18] O. Chapelle, B. Schölkopf, A. Zien, et al., Semi-supervised Learning, vol. 2, MIT Press, Cambridge, 2006.
[19] X. Zhu, Semi-supervised Learning Literature Survey, 2005.
[20] K. Bennett, A. Demiriz, et al., Semi-supervised support vector machines, in: Advances in Neural Information Processing Systems, 1999, pp. 368–374.
[21] T. Joachims, Transductive inference for text classification using support vector machines, in: ICML, vol. 99, 1999, pp. 200–209.
[22] E. Xing, A. Ng, M. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, Advances in Neural Information Processing Systems 15 (2002) 505–512.
[23] Z. Li, J. Liu, X. Tang, Constrained clustering via spectral regularization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE, Cambridge, United Kingdom, pp. 421–428.
[24] J.A. Suykens, J. De Brabanter, L. Lukas, J. Vandewalle, Weighted least squares support vector machines: robustness and sparse approximation, Neurocomputing 48 (1) (2002) 85–105.
[25] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67.
[26] T. Hastie, R. Tibshirani, J. Friedman, J. Franklin, The elements of statistical learning: data mining, inference and prediction, Math. Intell. 27 (2) (2005) 83–85.
[27] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, United Kingdom, 2009.
[28] A.-L. Barabási, Emergence of Scaling in Complex Networks, Handbook of Graphs and Networks: From the Genome to the Internet, 2002, pp. 69–84.
[29] R. Pastor-Satorras, A. Vespignani, Epidemic spreading in scale-free networks, Phys. Rev. Lett. 86 (14) (2001) 3200.
[30] S.S. Lafon, Diffusion maps and geometric harmonics (Ph.D. thesis), Yale University, 2004.
[31] F. Nie, D. Xu, X. Li, S. Xiang, Semisupervised dimensionality reduction and classification through virtual label regression, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 41 (3) (2011) 675–685.
[32] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[33] F.R. Chung, Spectral Graph Theory, vol. 92, American Mathematical Society, 1997.
[34] P. Kumar Mallapragada, R. Jin, A.K. Jain, Y. Liu, Semiboost: boosting for semi-supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2009) 2000–2014.



Enmei Tu was born in Anhui, China. He received his B. Sc. degree and M.Sc. degree from University of Electronic Science and Technology of China (UESTC) in 2007 and 2010, respectively. He got his Ph.D. degree from the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, China, in 2014. He is now working in the Knowledge Engineering and Discovery Research Centre (KEDRI), New Zealand, as a Research Fellow. His research interests are machine learning, computer vision and neural information processing.

Jie Yang was born in Shanghai, China, in August 1964. He received a Bachelor's degree and a Master's degree in Automatic Control in Shanghai Jiao Tong University in 1985 and 1988, respectively. In 1994, he received Ph. D. at Department of Computer Science, University of Hamburg, Germany. Now he is the Professor and Director of Institute of Image Processing and Pattern recognition in Shanghai Jiao Tong University. He is the Principal Investigator of more than 30 nation and ministry scientific research projects in image processing, pattern recognition, data mining, and artificial intelligence, including two national 973 research plan projects, three national 863 research plan projects, three national nature foundation projects, five international cooperative projects with France, Korea, Japan, New Zealand. He has published more than 500 of articles in national or international academic journals and conferences.

Nikola Kasabov is a Fellow of the Royal Society of New Zealand, the New Zealand Computer Society and the Institute of Electrical and Electronic Engineers (IEEE). He is the founding Director and the Chief Scientist of the Knowledge Engineering and Discovery Research Centre (KEDRI) and Personal Chair of Knowledge Engineering in the School of Computing and Mathematical Sciences at AUT. His main interests are in the areas of computational intelligence, neuro-computing, bioinformatics, neuroinformatics, speech and image processing, novel methods for data mining and knowledge discovery. He has published over 450 works, among them journal papers, text books, edited research books and monographs, conference papers, book chapters, edited conference proceedings, patents and authorship certificates in the area of intelligent systems, connectionist and hybrid connectionist systems, fuzzy systems, expert systems, speech recognition, bioinformatics, neurocomputing and neural networks.

Yaqian Zhang was born in Harbin, China, in 1992. She is now a Senior Student at Department of Electronic Engineering, Shanghai Jiao Tong University and will get her Bachelor Degree in Engineering, in 2015. Her research interests are face recognition and deep learning.