Multi-Manifold Positive and Unlabeled Learning for Visual Analysis

Chen Gong, *Member, IEEE*, Hong Shi, Jie Yang, and Jian Yang, *Member, IEEE*

***Abstract*— Positive and Unlabeled (PU) learning has attracted intensive research interests in recent years, which is capable of training a binary classifier solely based on positive and unlabeled examples when the negative data are absent or too are diverse. However, the existing PU learning methods largely overlook the relationship between the examples when handling the unlabeled data, leading to insufficient exploitation of data structure which actually contains useful distribution information. Therefore, by following the multi-manifold assumption which is observed in many real-world vision problems, this paper proposes a novel algorithm termed "Multi-Manifold PU learning" (MMPU), which assumes that the data belonging to different classes lie on different underlying manifolds. As such, the structural information revealed by the dataset is deployed, which is helpful in deciding the labels of unlabeled examples. Our MMPU contains two main steps, namely, multi-manifold exploration and positive confidence training, where the former is accomplished by computing the local similarity, structural similarity, and semantic similarity of pairwise data, and the latter establishes a binary classifier in reproducing kernel Hilbert space based on the real-valued confidence level of each example to be positive. Experimentally, we not only test the proposed MMPU on five highly nonlinear synthetic datasets but also apply MMPU to various typical computer vision tasks, including handwritten digit recognition, violent behavior detection, and hyperspectral image classification. The results demonstrate that MMPU can obtain a superior performance compared to the state-of-the-art PU learning methodologies.**

Manuscript received January 18, 2019; revised March 1, 2019; accepted March 3, 2019. Date of publication March 7, 2019; date of current version May 5, 2020. This work was supported by the NSF of China under Grant 61602246, Grant 61876107, Grant U1803261, and Grant U1713208, in part by the NSF of Jiangsu Province under Grant BK20171430, in part by the Fundamental Research Funds for the Central Universities under Grant 30918011319, in part by the Open Project of State Key Laboratory of Integrated Services Networks, Xidian University, under Grant ISN19-03, in part by the "Summit of the Six Top Talents" Program under Grant DZXX-027, in part by the "Innovative and Entrepreneurial Doctor" Program of Jiangsu Province, in part by the "Young Elite Scientists Sponsorship Program" by Jiangsu Province, in part by the "Young Elite Scientists Sponsorship Program" by CAST under Grant 2018QNRC001, in part by the Program for Changjiang Scholars, and in part by the 973 Plan, China, under Grant 2015CB856004. This paper was recommended by Associate Editor Y.-P. Tan. *(Corresponding authors: Chen Gong; Jian Yang.)*

C. Gong, H. Shi, and J. Yang are with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of the Ministry of Education, Jiangsu Key Laboratory of Image and Video Understanding for Social Security, and the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: chen.gong@njust.edu.cn; shihong@njust.edu.cn; csjyang@njust.edu.cn).

J. Yang is with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: jieyang@sjtu.edu.cn).

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

Digital Object Identifier 10.1109/TCSVT.2019.2903563

***Index Terms*— Positive and unlabeled learning, multi-manifold exploration, positive confidence training.**

I. Introduction

RECENT years have witnessed a dramatic increase of interest in the study of Positive and Unlabeled learning (PU learning) due to its usefulness and effectiveness [1]–[3]. The target of PU learning is to build a binary classifier solely based on positive and unlabeled training examples, so PU learning is very useful when the negative training data are absent or too diverse, such as

- *Image Retrieval [4]:* The query images constitute the positive examples, while the candidate image examples are treated as unlabeled as they contain both relevant and irrelevant images to the queries. In this application, we do not have explicit negative training examples, and thus PU learning can be adopted to identify the images of user's interest in the unlabeled set.
- *Automatic Face Tagging [2]:* A set of user's face images (i.e. positive examples) are firstly provided by himself/herself, and then PU learning can be employed to automatically tag the photos in the user's photo album by discriminating the user's face from others' faces.
- *Hyperspectral Image Classification [5]:* In remote sensing, sometimes the users are only interested in detecting one specific land cover type without considering other classes (*e.g.* the class of "tree" for studying forest expansion). In this case, it is easy to annotate some tree regions (i.e. positive data), but is difficult to exhaustively and representatively collect diverse non-tree regions (i.e. negative data), so PU learning can be utilized to conduct the detection of positive data.

Note that in above typical PU learning applications, every unlabeled example for building a PU classifier can be positive or negative, but the corresponding groundtruth label is unknown during the training stage. Therefore, it poses a great challenge for a PU learning algorithm to obtain the accurate two-class decision function. Although PU learning can be regarded as a special case of semi-supervised learning [6]–[8] when the labeled negative examples are not provided, their solutions are quite different. This is because the existing semi-supervised models cannot work properly without the existence of negative training data, so the algorithms that targets PU learning should be specifically designed.

Up to now, the existing PU learning methods can be attributed to three main categories according to how the unlabeled examples for training are dealt with. The methods of first category follow a two-step strategy which firstly identify a set of definite negative examples from unlabeled examples, and then adopt a traditional supervised learning algorithm to learn from these reliable negative examples as well as the original positive examples. Here the first step is critical to determining the final performance, and the inaccurate identification of definite negative examples may lead to disastrous outputs. The typical algorithms of this type include SEM [9], PEBL [10] and ROC-SVM [11], which adopt "spy technique", 1-DNF, and Rocchio algorithms [12] to accomplish the first step correspondingly. The second category regards the unlabeled data as highly noisy negative data, among which the original positive examples are deemed as mislabeled. Therefore, PU learning is transformed to the learning problem with one-sided label noise [13]–[15], as all known positive examples have clean labels. The representative works include biased SVM [16], weighted Logistic regression [17], and Loss Decomposition and Centroid Estimation [18]. The third category formulates PU learning as a cost-sensitive learning problem, in which each unlabeled example is viewed as a weighted positive example as well as a weighted negative example. A pioneering work is [19], which develops the weighted SVM and allocates different weights to the loss functions regarding positive and negative classes. After that, a non-convex ramp loss for PU learning is proposed in [20]. To fix the non-convexity issue, a convex unbiased double hinge loss is presented in [2]. To further address the overfitting problem inherited by [2], Kiryo *et al.* [3] suggest to force the loss value above zero, and the induced loss is called non-negative risk estimator. In the third category, every unlabeled example is injected to the loss functions regarding both positive and negative classes, which leads to the inappropriate penalties as the real label of an example should be unique. Consequently, their performances are still to be improved.

From above analyses, we see that different categories of existing methods have their own shortcomings. More critically, they isolatedly treat each of the unlabeled examples when computing the class probability or loss value. This largely ignores the relationship between data points, and thus the structural property carried by the training set has not been fully exploited. To this end, this paper relates the examples via manifold assumption and proposes a **M**ulti-**M**anifold **PU** learning (termed "MMPU") algorithm. To be specific, we assume that the data observations lie on multiple smooth low-dimensional manifolds, and each of the manifolds corresponds to a class. Such multi-manifold phenomenon has been widely observed in various computer vision applications, such as image set classification [21], gait analysis [22], [23], one-shot face recognition [24], motion segmentation [25], and object recognition [26]. Therefore, by discovering and utilizing the manifolds that support the entire dataset, our MMPU is able to obtain very encouraging performance in a variety of vision problems. The implementation of MMPU needs two steps. In the first step, a graph on the positive and unlabeled examples is built according to their similarities in the feature space [27], [28], where the local similarity, structural similarity and semantic similarity between pairs of examples are particularly considered. Based on the established graph, in the second step we treat the available positive examples as queries and rank the remaining unlabeled data along the manifolds, so that each of them will receive a confidence value to be positive. We further utilize the recent positive-confidence learning strategy [29] and formulate our problem as a risk minimization framework, which can be easily solved in the Repreducing Kernel Hilbert Space (RKHS). Due to the proper utilization of multi-manifold structural information, our MMPU yields superior performance to the state-of-the-art PU methods on a wide range of vision problems such as handwritten digit recognition, violent behavior detection, and hyperspectral image classification.

II. Related Work

In this section, we review the representative works on multi-manifold learning and PU learning, as these two learning frameworks are very relevant to the topic of this paper.

A. Multi-Manifold Learning

The early-staged manifold learning papers mainly work on single manifold, such as the well-known Isomap [30], LLE [31], and Laplacian Eigenmaps [32]. However, due to the complexity of practical data, single manifold cannot comprehensively describe the entire data distribution, so many researchers propose to use a mixture of manifolds to model the data observations [33]. Up to now, multi-manifold assumption has been intensively adopted in many learning tasks such as clustering, dimensionality reduction, and semi-supervised learning.

For clustering, it is assumed that each manifold represents a class and spectral theory is usually employed to achieve such clustering. For example, Wang *et al.* [34] employ spectral decomposition to divide the dataset into several sub-manifolds. Afterwards, they utilize the local geometric information of data points to handle the intersections and obtain the improved clustering results [35]. Besides, Gong *et al.* [36] estimate the local tangent space by weighted low-rank matrix factorization, and the estimated local structure is further used to assist the discovery of global structure. The local property is also employed by [37], in which local principal components analysis is performed in the selected neighborhoods so that the discrepancy between the principal subspaces of neighborhoods can be decided.

For dimensionality reduction, Li *et al.* [38] maximize the nonparametric manifold-to-manifold distances and meanwhile preserve the locality of manifolds to achieve the discriminant multi-manifold dimensionality reduction. Valencia-Aguirre *et al.* [39] extend the traditional Laplacian Eigenmaps to multi-manifold situations by computing the relationship among the examples of different classes based on an intra-manifold comparison. Apart from intra-manifold structure, the inter-manifold structure is also deployed by [22] to jointly learn the embedding results from different manifolds. Similarly, LLE is also adapted to multi-manifold cases based

on the manifold-to-manifold distance and point-to-manifold distance [40].

For semi-supervised learning, the first work considering multiple manifolds is [41]. This work adopts a "cluster-then-label" strategy and uses the Hellinger distance to depict the intuition that two points on different manifolds or in the regions with different density should be considered dissimilar. Differently, Xing *et al.* [42] propose a novel multi-manifold semi-supervised Gaussian mixture model by introducing a local tangent space based geometrical similarity.

Multi-manifold learning has been broadly applied to a wide range of visual applications. For example, Lu *et al.* [21] propose the multi-manifold deep metric learning to recognize an object of interest from a set of image instances captured from varying viewpoints or under varying illuminations. Lu *et al.* [24] also come up with a novel discriminative multi-manifold analysis method for face recognition from a single example by learning discriminative features from the non-overlapping image patches. Goh and Vidal [25] hypothesize that the point trajectories associated with different motions reside in different manifolds, and they cast motion segmentation as a clustering problem on multiple manifolds.

From above literature review, we see that multi-manifold learning is beneficial for many learning algorithms to boosting the performance, yet it has not been utilized by PU learning to properly identify the positive and negative classes in the unlabeled set.

B. Positive and Unlabeled Learning

PU learning is an emerging topic in weakly-supervised learning, which aims to train a binary classifier by simply harnessing positive examples and unlabeled examples. So far, the developed algorithms can be divided into the following three categories.

In the first category, some definite negative examples are detected in the first step, and then a conventional supervised classifier is applied to the detected negative examples as well as the original positive examples in the second step. To accomplish the first step, Liu *et al.* [9] introduce an interesting "spy" technique which sends "spy" examples from the positive set to the unlabeled set so that the probability of an unlabeled example belonging to the positive class can be estimated. Besides, Yu *et al.* [10] explore the feature frequencies in positive set and unlabeled set, and develop the 1-DNF technique to identify the definite negative examples. Moreover, Liu *et al.* [16] and Xiao *et al.* [43] respectively adopt the naive Bayesian classifier and K-means to determine the most likely negative examples and positive examples in the unlabeled set. For the second step, some traditional supervised algorithms such as SVM are implemented repeatedly or all at once to fulfill the binary classification [44]. Although the methods of first category are simple, they suffer from a critical drawback that the model output is strongly dependent on the quality of the first step. If the extracted negative examples are inaccurate in the first step, the classifier trained in the second step will be severely biased.

The second category formulates PU learning as a learning problem with one-sided noisy labels. That is to say, all unlabeled examples as treated as negative with incorrect labels, and the positive examples are "clean", i.e., without any label noises. Here the examples that are originally positive in the unlabeled set are regarded as mislabeled. Li *et al.* [16] propose the biased SVM which imposes different regularization parameters on the slack variables that control the tolerance of in-margin examples or even errors. Similarly, Lee and Liu [17] devise the weighted Logistic regression to handle the noisy labels, and the weights are selected on a validation set. However, the regularization weights for both biased SVM and weighted Logistic regression are tuned via some empirical or heuristic ways, so the model performance is very sensitive to their choices. To solve this problem, Shi *et al.* [18] propose an unbiased estimation of the true risk on PU datasets by utilizing the techniques of loss decomposition and centroid estimation. Considering that the positive examples are not uniformly sampled in many real-world applications, He *et al.* [45] propose an instance-dependent PU algorithm in which the probability of an example being positive is related to its feature representation. The methods of second category require the estimation of class prior in the training set, which could be difficult and error-prone. An inaccurate estimation may hurt the performance of PU algorithm.

The algorithms of third category transform PU learning to a cost-sensitive learning problem, in which each unlabeled example is viewed as a weighted positive example as well as a weighted negative example. So far, the algorithms belonging to this category have achieved the state-of-the-art performances. The first work of this category is arguably [19], which proposes the weighted SVM to assign corresponding weights to the class-specific losses. After that, Plessis *et al.* [20] favor to weight the per-class cost of every example according to the estimated class priors and propose a non-convex ramp loss to conduct PU classification. However, the non-convexity of the loss function in [20] may lead to the difficulty for the subsequent optimization, so a convex unbiased double hinge loss is presented in [2], which is composed of a weighted ordinary convex loss for unlabeled data and a weighted composite convex loss function for positive data. The scalability of this method is improved by a modified Sequential Minimal Optimization (SMO) approach with a significant reduction in memory and computation [4]. Unfortunately, [2] has a significant drawback that it may easily lead to the overfitting problem as the empirical risks of this method on training examples might be negative. Therefore, Kiryo *et al.* [3] amend the loss function in [2] by lower-bounding the loss value to be zero, and the designed loss is called non-negative risk estimator. Note that all the methods of this category simultaneously compute the losses of every unlabeled example on both positive and negative classes, so they inevitably introduce noises as the groundtruth label of an example is unique.

Other typical PU models include [46] based on positive margin, [47] based on generative adversarial learning, [48] for multi-label ranking, and [49] for semi-supervised learning. From above analyses, we see that there are no manifold-based PU algorithms that are specifically designed

for vision applications. Besides, most of the existing PU methods such as [2], [3], [5], and [18] directly learn the model parameters on the PU dataset without sufficiently exploiting the data distribution, so they are not likely to generate good performance when multi-manifold structure exists in the dataset. Due to the existence of multiple manifolds, the data distribution can be quite complicated, so the two-step paradigm employed by our MMPU could be better which firstly identifies the data distribution and then learns the model parameters.

III. Multi-Manifold Exploration

Let $X \in \mathbb{R}^d$ (d is data dimensionality) and $Y \in \{+1, -1\}$ be the input and output random variables, and $P(X, Y)$ be the joint distribution of (X, Y). Suppose we have a set of examples $\mathcal{X} = \{(\mathbf{x}_1, y_1), \cdots (\mathbf{x}_p, y_p), (\mathbf{x}_{p+1}, y_{p+1}), \cdots, (\mathbf{x}_n, y_n)\}$ identically and independently drawn from the distribution $P(X, Y)$, where $\{\mathbf{x}_i\}_{i=1}^n$ are features of the n examples, and y_i's are the labels of the corresponding examples. Note that the first p examples in $\mathcal{X}$ are assumed to be positive which are denoted as the positive set $\mathcal{P}$, and the remaining $u = n - p$ examples are unlabeled which are included in the unlabeled set $\mathcal{U}$. Therefore, given the hypothesis space as $\mathcal{F}$, the target of PU learning is to find a suitable decision function $f \in \mathcal{F} : \mathbb{R}^d \to \mathbb{R}$ based on $\mathcal{X} = \mathcal{P} \cup \mathcal{U}$ so that the unseen test example $\mathbf{x}$ can obtain the correct label assignment $sgn(f(\mathbf{x})) \in \{-1, 1\}$.

The proposed MMPU algorithm contains two steps, namely multi-manifold exploration and positive confidence training, and this section explains the details of the first step. The target of multi-manifold exploration is to discover the underlying multiple manifolds hidden in the dataset, so that they can aid the subsequent model training on PU dataset. Since PU learning is about binary classification, there are totally two manifolds which correspond to the positive class and negative class accordingly. To discover them, we build a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ over the training set $\mathcal{X}$ where $\mathcal{V}$ is the node set consisted of all n examples and $\mathcal{E}$ is the edge set depicting their similarities. Note that the traditional graph construction techniques are not suitable here as they are based on the assumption that there is only one manifold in the dataset. Consequently, they cannot be directly used for discovering the multi-manifold structure as the confusion will happen in the intersections of manifolds.

Inspired by [35], we exploit the geometric information of data to establish the graph such that different manifolds can be separated as much as possible. To be specific, in the intersecting regions, the data points on the same manifold will have similar local tangent spaces while the tangent spaces of examples belonging to different manifolds can be quite dissimilar. Therefore, it is necessary to find the local tangent space of a data point in advance.

A. Tangent Space Computation

Formally, the tangent space of $\mathbf{x}$ on a manifold is defined as $\Theta_{\mathbf{x}} = span(U_b)$ where U_b is the first b singular vectors of the covariance matrix formed by $\mathbf{x}$ and its neighbors in Euclidean space. However, in the intersecting regions of two manifolds, the two points of different manifolds can be quite close, so their associated covariance matrices will be very similar, which leads to the indistinguishable tangent spaces. Therefore, [35] favors to employ the Mixtures of Probabilistic Principal Component Analysis (MPPCA) to compute the local tangent space of $\mathbf{x}$, and such manipulation is based on two observations, i.e. 1) the global nonlinear manifold can be approximated by a series of local linear manifolds, and 2) the principal component analyzer can successfully travel through the intersecting regions of different manifolds.

Suppose the adopted MPPCA contains the mixture of M principal component analyzers $\theta_m = \{\boldsymbol{\mu}_m, \mathbf{V}_m, \sigma_m^2\}_{m=1}^M$, where $\boldsymbol{\mu}_m \in \mathbb{R}^d$, $\mathbf{V}_m \in \mathbb{R}^{d \times b}$, and σ_m^2 is a scalar. Therefore, for the m-th analyzer ($m = 1, 2, \cdots, M$), the original d-dimensional example $\mathbf{x}$ can be represented by a b-dimensional latent vector $\tilde{\mathbf{x}}$, namely

$$\mathbf{x} = \mathbf{V}_m \tilde{\mathbf{x}} + \boldsymbol{\mu}_m + \boldsymbol{\epsilon}_m, \tag{1}$$

where $\boldsymbol{\mu}_m$ represents the data mean, and the latent variable $\tilde{\mathbf{x}}$ and the noise $\boldsymbol{\epsilon}_m$ are Gaussians satisfying $\tilde{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\epsilon}_m \sim \mathcal{N}(\mathbf{0}, \sigma_m^2 \mathbf{I})$, respectively. Consequently, the marginal distribution of $\mathbf{x}$ is expressed as

$$P(\mathbf{x}|m) = \frac{1}{(2\pi)^{d/2}|\mathbf{C}_m|^{1/2}} \times \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_m)^\top \mathbf{C}_m^{-1}(\mathbf{x} - \boldsymbol{\mu}_m)\right\}, \tag{2}$$

where $\mathbf{C}_m$ is the covariance matrix that has the formation $\mathbf{C}_m = \sigma_m^2 \mathbf{I} + \mathbf{V}_m \mathbf{V}_m^\top$. As a result, the parameters $\boldsymbol{\mu}_m$, $\mathbf{V}_m$ and σ_m^2 can be learned via maximizing the log-likelihood of observing the totally n examples $\{\mathbf{x}_i\}_{i=1}^n$, namely

$$Likelihood = \sum_{i=1}^n \ln\left\{\sum_{m=1}^M \kappa_m p(\mathbf{x}_i|m)\right\}, \tag{3}$$

where κ_m ($m = 1, 2, \cdots, M$) are mixing proportions satisfying $\kappa_m \geq 0$ and $\sum_{m=1}^M \kappa_m = 1$. Specifically, the well-known Expectation Maximization (EM) algorithm is adopted to estimate the parameters, which is consisted of the following E-step and M-step.

1) E-Step: Given the parameters θ_m in the current iteration, we have

$$R_{im} = \frac{\kappa_m p(\mathbf{x}_i|m)}{\sum_{m=1}^M \kappa_m p(\mathbf{x}_i|m)}, \tag{4}$$

$$\kappa_m^{new} = \frac{1}{n}\sum\nolimits_{i=1}^n R_{im}, \tag{5}$$

$$\boldsymbol{\mu}_m^{new} = \frac{\sum_{i=1}^n R_{im}\mathbf{x}_i}{\sum_{i=1}^n R_{im}}, \tag{6}$$

where the superscript "new" denotes the updated variables.

2) M-Step: In this step, we update the parameters $\mathbf{V}_m$ and σ_m^2 as

$$\mathbf{V}_m^{new} = \mathbf{S}_m \mathbf{V}_m (\sigma_m^2 \mathbf{I} + \mathbf{T}_m^{-1} \mathbf{V}_m^\top \mathbf{S}_m \mathbf{V}_m)^{-1}, \tag{7}$$

$$(\sigma_m^2)^{new} = \frac{1}{b} tr\left(\mathbf{S}_m - \mathbf{S}_m \mathbf{V}_m \mathbf{T}_m^{-1} (\mathbf{V}_m^{new})^\top\right), \tag{8}$$

where

$$\mathbf{S}_m = \frac{1}{n\kappa_m^{new}} \sum\nolimits_{i=1}^n R_{im}(\mathbf{x}_i - \boldsymbol{\mu}_m^{new})(\mathbf{x}_i - \boldsymbol{\mu}_m^{new})^\top, \tag{9}$$

and

$$\mathbf{T}_m = \sigma_m^2 \mathbf{I} + \mathbf{V}_m^\top \mathbf{V}_m. \tag{10}$$

Therefore, the example $\mathbf{x}_i$ is assumed to come from the m'-th local analyzer if $P(\mathbf{x}_i|m') = \max_m P(\mathbf{x}_i|m)$, and the local tangent space of $\mathbf{x}_i$ (denoted as $\Theta_{\mathbf{x}_i}$) is then spanned by the row vectors of $\mathbf{V}_{m'}$, namely $\Theta_{\mathbf{x}_i} = span(\mathbf{V}_{m'})$.

B. Multi-Manifold Graph Construction

Benefiting from the method for computing the tangent space of an example introduced in Section III-A, we may build a graph $\mathcal{G}$ that captures the multi-manifold structure hidden in the entire dataset $\mathcal{X}$. In our paper, the similarity of two points in the graph are governed by the following three factors:

- *Local Similarity:* It is defined within a local area and is determined by the Euclidean distance between two examples.
- *Structural Similarity:* It is evaluated by the similarity of the tangent spaces of two examples, which is critical to distinguishing different manifolds in their intersections.
- *Semantic Similarity:* It is related to the high-level class information carried by the examples.

The local similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ is binarized as

$$s_{ij}^{local} = \begin{cases} 1, & \mathbf{x}_i \in KNN(\mathbf{x}_j) \ or \ \mathbf{x}_j \in KNN(\mathbf{x}_i) \\ 0, & otherwise \end{cases} \tag{11}$$

where $KNN(\mathbf{x})$ denotes the set of K nearest neighbors of $\mathbf{x}$ measured by the Euclidean distance. We adopt KNN graph as it is sparse which usually leads to satisfactory performance [50]–[52].

Given the tangent spaces of $\mathbf{x}_i$ and $\mathbf{x}_j$ as $\Theta_{\mathbf{x}_i}$ and $\Theta_{\mathbf{x}_j}$, the structural similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined by

$$s_{ij}^{structural} = similarity(\Theta_{\mathbf{x}_i}, \Theta_{\mathbf{x}_j}) = \left(\prod_{k=1}^{b} \cos(z_k) \right)^h, \tag{12}$$

where $h > 0$ is a tuning parameter, and $0 \le z_1 \le \cdots \le z_b \le \pi/2$ are a series of principal angles between the two tangent spaces $\Theta_{\mathbf{x}_i}$ and $\Theta_{\mathbf{x}_j}$ which is recursively defined as

$$\cos(z_1) = \max_{\mathbf{u}_1 \in \Theta_{\mathbf{x}_i}, \mathbf{v}_1 \in \Theta_{\mathbf{x}_j}, \|\mathbf{u}_1\| = \|\mathbf{v}_1\| = 1} \mathbf{u}_1^\top \mathbf{v}_1 \tag{13}$$

and

$$\cos(z_k) = \max_{\mathbf{u}_k \in \Theta_{\mathbf{x}_i}, \mathbf{v}_k \in \Theta_{\mathbf{x}_j}, \|\mathbf{u}_k\| = \|\mathbf{v}_k\| = 1} \mathbf{u}_k^\top \mathbf{v}_k, \quad k = 2, \cdots, b \tag{14}$$

where $\mathbf{u}_k^\top \mathbf{u}_i = 0$ and $\mathbf{v}_k^\top \mathbf{v}_i = 0$ for $i = 1, 2, \cdots, k-1$.

The semantic similarity of a pair of examples depends on whether the two examples have the same class label, and the value is 1 if both of them are in the positive set $\mathcal{P}$, namely

$$s_{ij}^{semantic} = \begin{cases} 1, & \mathbf{x}_i \in \mathcal{P} \ and \ \mathbf{x}_j \in \mathcal{P} \\ 0, & otherwise. \end{cases} \tag{15}$$

By taking above three factors into consideration, the integrated similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$ is formally defined as

$$w_{ij} = \max \left\{ s_{ij}^{local} s_{ij}^{structural}, \mathbb{I}[s_{ij}^{semantic} = 1] \right\} = \begin{cases} 1, & \mathbf{x}_i \in \mathcal{P} \text{ and } \mathbf{x}_j \in \mathcal{P} \\ \left(\prod_{k=1}^{b} \cos(z_k) \right)^h, & \mathbf{x}_i \in KNN(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in KNN(\mathbf{x}_i), \\ & \text{at most one of } \mathbf{x}_i, \mathbf{x}_j \in \mathcal{P} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where "$\mathbb{I}[\cdot]$" is an indicator function which returns 1 if $s_{ij}^{semantic} = 1$. Therefore, the adjacency matrix $\mathbf{W}$ of graph $\mathcal{G}$ is formed as $(\mathbf{W})_{ij} = w_{ij}$, and the (i,i)-th element of the induced diagonal degree matrix $\mathbf{D}$ is computed by $\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{W}_{ij}$.

IV. Positive Confidence Training

Given the graph $\mathcal{G}$ quantified by the adjacency matrix $\mathbf{W}$, we can get the confidence of every example being positive via the technique of manifold ranking [53], [54]. Specifically, we regard the initial positive examples in $\mathcal{P}$ as queries, and rank the rest unlabeled examples in $\mathcal{U}$ according to their similarities to the positive examples revealed by $\mathcal{G}$. To this end, an n-dimensional column vector $\mathbf{y} = (y_1, y_2, \cdots, y_n)^\top$ is employed where $y_i = 1$ if $\mathbf{x}_i \in \mathcal{P}$ and $y_i = 0$ if $\mathbf{x}_i \in \mathcal{U}$. Therefore, the ranking result can be obtained by

$$\mathbf{r} = \left(\mathbf{I} - \tilde{\alpha} \mathbf{D}^{-1} \mathbf{W} \right)^{-1} \mathbf{y}, \tag{17}$$

where $\mathbf{I}$ is the identity matrix throughout this paper, $\tilde{\alpha}$ is the free parameter set to 0.99 [53], [54], and r_i (i.e. the i-th element of $\mathbf{r}$) reflects the closeness of $\mathbf{x}_i$ to the positive examples. To make r_i have probabilistic interpretation, we normalize the vector $\mathbf{r}$ to the range $[0, 1]$, and set the i-th elements to 1 if $\mathbf{x}_i \in \mathcal{P}$. Consequently, the variable $\bar{\mathbf{r}}$, which is the normalized $\mathbf{r}$, reveals the confidence of $\mathbf{x}_i$ to be positive.

Here, the positive confidence value of every example encoded in $\bar{\mathbf{r}}$ is able to reflect the class information to some extent. However, the confidence values are continuous which brings about ambiguity. Moreover, our target is to obtain a well-trained PU classifier f that is generalizable to unseen test data and can accomplish the out-of-sample prediction. Therefore, in the following we aim to establish a binary classifier on the data with positive confidence values. According to [55], given a classifier $f(\mathbf{x})$ and some loss function $\ell(\cdot)$, we want to minimize the following classification risk:

$$\mathcal{R}(f) = \mathbb{E}_{P(X,Y)}[\ell(yf(\mathbf{x}))], \tag{18}$$

where $\mathbb{E}_{P(X,Y)}$ is the expectation over $P(X, Y)$. Empirically, given n training examples $\{\mathbf{x}_i\}_{i=1}^n$ and a decision function $f(\mathbf{x})$, above risk minimization process can be formulated as

$$\min_f \sum_{i=1}^{n} [\ell(f(\mathbf{x}_i)) + \frac{1 - \bar{r}_i}{\bar{r}_i} \ell(-f(\mathbf{x}_i))]. \tag{19}$$

where $\bar{r}_i$ is the i-th element of vector $\bar{\mathbf{r}}$.

To enable our MMPU to handle non-linear cases, we build our model in the Reproducing Kernel Hilbert Space (RKHS). An RKHS $\mathcal{H}_K$ is a Hilbert space $\mathcal{H}$ of functions on a set $\mathcal{D}$ with the property that for all $x \in \mathcal{D}$ and $f \in \mathcal{H}$, the point evaluations $f \to f(x)$ are continuous linear functionals [56]. Consequently, according to the Moore-Aronszajn theorem [57], there exists a unique positive definite kernel $K(\cdot,\cdot)$ on $\mathcal{D} \times \mathcal{D}$ which has an important property that $\forall x_1, x_2 \in \mathcal{D}$, $K(x_1, x_2) = \langle K(\cdot, x_1), K(\cdot, x_2)\rangle_{\mathcal{H}}$. Therefore, if we employ the hinge loss as our loss function (i.e., $\ell(f(\mathbf{x}), y) = \max(0, 1 - yf(\mathbf{x})) = [1 - yf(\mathbf{x})]_+$), and further introduce the regularizer $\|f\|^2_{\mathcal{H}}$ to (19) to prevent overfitting, the optimization model for MMPU is expressed as

$$\min_{f\in\mathcal{H}_k} \sum_{i=1}^{n} \left([1 - f(\mathbf{x}_i)]_+ + R_i[1 + f(\mathbf{x}_i)]_+\right) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}, \quad (20)$$

where $R_i = (1 - \bar{r}_i)/\bar{r}_i$ and λ is the nonnegative trade-off parameter.

According to the representer theorem, the minimizer of (20) can be written as the expansion of kernel functions on all n training examples, namely

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b, \quad (21)$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_n]^\top$ denotes coefficient vector and b is the biased term. In this work, we employ the graph diffusion kernel [58], [59] instead of the traditional Gaussian kernel so that the similarities of data reflected by the kernel are compatible with their relationship on the manifolds, namely $\mathbf{K} = (\mathbf{I} - \tilde{\alpha}\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})^{-1}$ where the (i, j)-th element of $\mathbf{K}$ refers to $K(\mathbf{x}_i, \mathbf{x}_j)$. By incorporating b into $\boldsymbol{\alpha}$ as $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^\top\ b)^\top$ and augmenting the kernel matrix $\mathbf{K}$ as $\mathbf{K} = (\mathbf{K}\ \mathbf{1})$ with $\mathbf{1}$ being the all-one column vector, (21) can be written in a concise form as $f(\mathbf{x}) = \mathbf{K}\boldsymbol{\alpha}$. As a result, by introducing the slack variables $\{\xi_i\}_{i=1}^n$ and $\{\zeta_i\}_{i=1}^n$, the primal problem equivalent to (20) is

$$\min_{\xi_i,\zeta_i,\boldsymbol{\alpha}} \sum_{i=1}^{n}(\xi_i + R_i\zeta_i) + \frac{\lambda}{2}\boldsymbol{\alpha}^\top\mathbf{K}^\top\mathbf{K}\boldsymbol{\alpha} \quad (22)$$

$$s.t. \quad \sum_{j=1}^{n} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b \geq 1 - \xi_i, \quad i = 1, \cdots, n \quad (23)$$

$$-\sum_{j=1}^{n} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b \geq 1 - \zeta_i, \quad i = 1, \cdots, n \quad (24)$$

$$\xi_i \geq 0, \quad \zeta_i \geq 0, \ i = 1, \cdots, n \quad (25)$$

where (23) and (24) correspond to the hinge losses on positive and negative classes, respectively.

To solve above constrained optimization problem, we introduce the Lagrangian variables $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_n)$, $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_n)$, $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_n)$ and $\boldsymbol{\psi} = (\psi_1, \cdots, \psi_n)$, and arrive at the following Lagrangian function:

$$\begin{aligned}\mathcal{J}(\boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\zeta}, b, \boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\tau}, \boldsymbol{\psi}) &= \frac{\lambda}{2}\boldsymbol{\alpha}^\top\mathbf{K}^\top\mathbf{K}\boldsymbol{\alpha} + \sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n} R_i\zeta_i \\ &+ \sum_{i=1}^{n}\beta_i(1 - \xi_i - \sum_{j=1}^{n}\alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - b) \\ &+ \sum_{i=1}^{n}\theta_i(1 - \zeta_i + \sum_{j=1}^{n}\alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b) \\ &- \sum_{i=1}^{n}\xi_i\tau_i - \sum_{i=1}^{n}\zeta_i\psi_i. \end{aligned} \quad (26)$$

To obtain the dual form, we compute the derivative of $\mathcal{J}$ to b, ξ_i and ζ_i, and then set the results to zero, namely

$$\begin{cases} \dfrac{\partial\mathcal{J}}{\partial b} &= -\sum_{i=1}^{n}\beta_i + \sum_{i=1}^{n}\theta_i = 0 \\ \dfrac{\partial\mathcal{J}}{\partial\xi_i} &= 1 - \beta_i - \tau_i = 0 \\ \dfrac{\partial\mathcal{J}}{\partial\zeta_i} &= R_i - \theta_i - \psi_i = 0 \end{cases} \quad (27)$$

which leads to

$$\begin{cases} \sum_{i=1}^{n}\beta_i &= \sum_{i=1}^{n}\theta_i \\ \beta_i + \tau_i &= 1 \quad \Rightarrow 0 \leq \beta_i \leq 1 \ (\because\ \tau_i \ is\ nonnegative) \\ \theta_i + \psi_i &= R_i \quad \Rightarrow 0 \leq \theta_i \leq R_i \ (\because\ \psi_i \ is\ nonnegative). \end{cases} \quad (28)$$

By substituting the above results into (26), we have the reduced Lagrangian function as

$$\mathcal{J}^R(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta}) = \frac{\lambda}{2}\boldsymbol{\alpha}^\top\mathbf{K}^\top\mathbf{K}\boldsymbol{\alpha} + \boldsymbol{\beta}^\top\mathbf{1} + \boldsymbol{\theta}^\top\mathbf{1} - \boldsymbol{\beta}^\top\mathbf{K}\boldsymbol{\alpha} + \boldsymbol{\theta}^\top\mathbf{K}\boldsymbol{\alpha}. \quad (29)$$

Taking derivative of the reduced Lagrangian function $\mathcal{J}^R(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\alpha}$, we obtain

$$\frac{\partial\mathcal{J}^R}{\partial\boldsymbol{\alpha}} = \lambda\mathbf{K}^\top\mathbf{K}\boldsymbol{\alpha} - \mathbf{K}^\top\boldsymbol{\beta} + \mathbf{K}^\top\boldsymbol{\theta}, \quad (30)$$

which implies

$$\boldsymbol{\alpha} = \frac{1}{\lambda}\left(\mathbf{K}^\top\mathbf{K}\right)^{-1}\mathbf{K}^\top(\boldsymbol{\beta} - \boldsymbol{\theta}), \quad (31)$$

where $\left(\mathbf{K}^\top\mathbf{K}\right)^{-1}$ is replaced by $\left(\mathbf{K}^\top\mathbf{K} + \epsilon\mathbf{I}\right)^{-1}$ in practical implementation with ϵ being a small positive number. By substituting (31) back to (29), we obtain the dual form of the original problem (22)$\sim$(25), which is

$$\begin{aligned} \max_{\boldsymbol{\beta},\boldsymbol{\theta}} \ & \sum_{i=1}^{n}\beta_i + \sum_{i=1}^{n}\theta_i - \frac{1}{2\lambda}(\boldsymbol{\beta} - \boldsymbol{\theta})^\top\tilde{\mathbf{K}}(\boldsymbol{\beta} - \boldsymbol{\theta}) \\ s.t. \ & \sum_{i=1}^{n}\beta_i = \sum_{i=1}^{n}\theta_i \\ & 0 \leq \beta_i \leq 1, \quad i = 1, \ldots, n \\ & 0 \leq \theta_i \leq R_i, \quad i = 1, \ldots, n \end{aligned}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ are dual variables and $\tilde{\mathbf{K}} = \mathbf{K}\left(\mathbf{K}^\top\mathbf{K}\right)^{-1}\mathbf{K}^\top$.

By merging $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ into a vector $\mathbf{h}$ as $\mathbf{h} = \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\theta} \end{pmatrix}$, and also denoting $\mathbf{E}_1 = \begin{pmatrix} \mathbf{I}_n \\ \mathbf{I}_n \end{pmatrix}$ and $\mathbf{E}_2 = \begin{pmatrix} \mathbf{I}_n \\ -\mathbf{I}_n \end{pmatrix}$ with $\mathbf{I}_n$ being the $n \times n$ identity matrix, (32) is equivalent to

$$\begin{aligned} \min_{\mathbf{h}} \quad & \frac{1}{2\lambda}\mathbf{h}^\top \mathbf{E}_2 \tilde{\mathbf{K}} E_2^\top \mathbf{h} - \mathbf{1}^\top \mathbf{E}_1^\top \mathbf{h} \\ s.t. \quad & \mathbf{1}^\top \mathbf{E}_2^\top \mathbf{h} = 0 \\ & \mathbf{0}_{2n} \leq \mathbf{h} \leq \begin{pmatrix} \mathbf{1} \\ \mathbf{R} \end{pmatrix} \end{aligned} \tag{32}$$

where $\mathbf{R} = (R_1, R_2, \cdots, R_n)^\top$ and $\mathbf{0}_{2n}$ represents the $2n$-dimensional all-zero column vector. Note that (32) is a standard quadratic programming problem, and it can be easily solved via many off-the-shelf toolboxes. In this paper, we use the "quadprog" command in Matlab to find its solution.

Algorithm 1: Pseudo-code of training the proposed MMPU

Input: Number of neighbors K, trade-off parameter λ, $h = 0.01$, $\tilde{\alpha} = 0.99$

Output: the optimal classifier parameter $\boldsymbol{\alpha}$

1 Construct the initial KNN graph $\mathcal{G}$;
2 Compute s_{ij}^{local}, $s_{ij}^{structural}$ and $s_{ij}^{semantic}$ via (11), (12) and (15), respectively;
3 Compute the multi-manifold similarity matrix $\mathbf{W}$ via (16) and the associated degree matrix $\mathbf{D}$;
4 Find ranking result $\mathbf{r}$ via (17) ;
5 Compute graph diffusion kernel matrix $\mathbf{K} = (\mathbf{I} - \tilde{\alpha}\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})^{-1}$;
6 Find the solution of problem (32) via existing QP solver;
7 Compute the classifier parameter via (31)

The entire process for training the proposed MMPU classifier is summarized in Alg. 1. In Alg. 1, the computational complexity for building the graph $\mathcal{G}$ and calculating the similarity matrix (i.e. Lines 1–3) is $\mathcal{O}(n^2)$. The complexity for obtaining $\mathbf{r}$ (i.e. Line 4) is at most $\mathcal{O}(n^3)$ as (17) can be transformed to solving a linear system. Line 5 takes $\mathcal{O}(n^3)$ complexity due to the inverse operation. Besides, the worst complexity for the QP solver in Line 6 is $\mathcal{O}(4n^2)$ by noting that the Hessian matrix $\frac{1}{\lambda}\mathbf{E}_2\tilde{\mathbf{K}}\mathbf{E}_2^\top$ is sparse, and the complexity of Line 7 is $\mathcal{O}(2n^3 + 2n^2)$. Therefore, the total complexity of Alg. 1 is at most $\mathcal{O}(4n^3 + 7n^2)$.

V. Experiments

In this section, we test the capability of our proposed MMPU algorithm on both synthetic and real-world datasets. The compared methodologies include:

- *One-class SVM (OCSVM) [60]:* In this method, only the examples in $\mathcal{P}$ are employed to obtain the one-class support vector machines classifier.
- *Weighted SVM (WSVM) [19]:* This is one of the traditional PU learning methods which assigns different weights to an example w.r.t. both positive and negative classes.
- *Double Hinge Loss (DH) [2]:* This convex estimator is proved to converge to the optimal solution for PU learning problem under the framework of empirical risk minimization.
- *Multi-Layer Perceptron With Non-Negative PU Risk Estimator (NNPU-MLP) [3]:* This method devises a novel non-negative risk estimator and then applies it to the multi-layer perceptron classifier.
- *Linear classifier With Non-Negative PU Risk Estimator (NNPU-Linear) [3]:* The results are generated by incorporating the above-mentioned non-negative risk estimator to the linear classifier.
- *Loss Decomposition and Centroid Estimation (LDCE) [18]:* This is a recently proposed PU method which decomposes the hinge loss to a label-dependent term and a label-independent that help to estimate the mean of the unlabeled data.

A. Synthetic Data

To visually compare the classification results of MMPU and the baseline methods, we present their outputs on three 2D datasets (i.e. *TwoMoons*, *TwoLines*, and *TwoSpirals*) and two 3D datasets (i.e. *TwoKnots* and *Roll&Plane*).

All 2D datasets are presented in the "initial" subfigures of Fig. 1. The *TwoMoons* dataset consists of 640 examples, which are equally divided into two moons, and each moon forms a class. The entire dataset is contaminated by the Gaussian noise with standard deviation 0.15. Among the 320 positive examples, only 32 are included into $\mathcal{P}$ (see the red dots) and the remaining positive examples are regarded as unlabeled. The *TwoLines* dataset is made up of 402 examples distributed along two lines (i.e. two classes) with the standard deviation 1, and only 10 positive examples are explicitly labeled. More seriously, the two lines intersect at the point $(0, 0)$, which may confuse the compared algorithms. The *TwoSpirals* dataset contains two nonlinear spirals with each of them constituting a class. In this dataset, 100 out of 500 original positive examples are labeled, and the number of unlabeled examples is 900. The intersecting point of these two spirals is also located at $(0, 0)$.

All 3D datasets are illustrated in Fig. 2, in which the subfigures of "initial" show the initial states of the investigated datasets. From Fig. 2, we see that the *TwoKnots* dataset is shaped like a knot composed of two crossing rings with radius 0.8, and each ring represents a class. This dataset is contaminated by the Gaussian noise with standard deviation 0.3, and only $9/721 \approx 1.25\%$ positive examples have been labeled. The *Roll&Plane* dataset forms like a Swiss roll penetrated by a plane, among which the plane represents the positive class and the roll constitutes the negative class. Note that this dataset is quite challenging as the two manifolds intersects four times in the 3D space, and this poses a great difficulty for an algorithm to correctly discriminate the data points from different classes.

For OCSVM, the Gaussian kernel is adopted, and the kernel width is respectively tuned to 1, 0.001, 10, 0.01 and 0.01 on *TwoMoons*, *TwoLines*, *TwoSpirals*, *TwoKnots* and *Roll&Plane* datasets to achieve the optimal performance. The positive class prior for implementing DH is estimated via the ℓ_1-distance

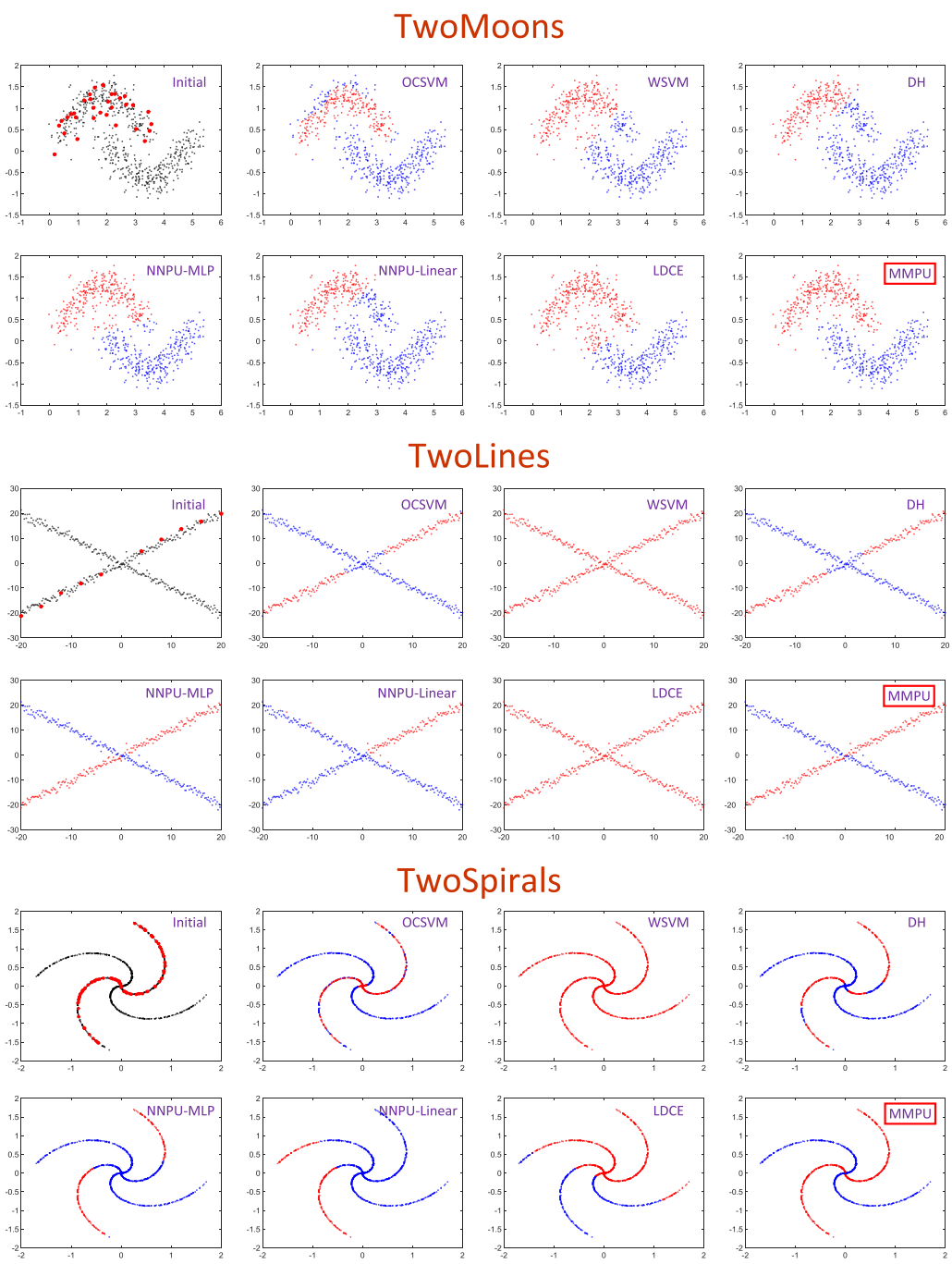

Fig. 1. The performance comparison of various methods on 2D datasets. In each dataset, the "initial" subfigure presents the initial annotation of the corresponding dataset, where the red dots represent positive examples and the black dots denote unlabeled examples. In the output of every compared method, the red dots and blue dots denote the determined positive examples and negative examples, respectively.

minimization proposed in [61]. In NNPU-MLP and NNPU-Linear, the γ for step size discount is set to the default value 1, and [3] has shown that the final performance is not sensitive to the choice of this parameter. In LDCE, we set the regularization parameter λ to 1, 1, 1, 10, 1, and the tolerance β to 0.01, 1, 10, 0.1, 0.01 on *TwoMoons*, *TwoLines*, *TwoSpirals*, *TwoKnots* and *Roll&Plane* correspondingly by searching the grid $\{0.001, 0.01, 0.1, 1, 10, 100\}$. For our proposed MMPU, the regularization parameter λ is adjusted to 10, 1, 1, 1, 1 on these five datasets, and the parameter K for graph construction is 10, 10, 5, 7, 5 correspondingly.

The qualitative comparison of all methods on 2D and 3D datasets are illustrated in Figs. 1 and 2. We can easily find that almost all baseline methods fail in handling the datasets with multiple nonlinear manifolds. The incorrect label transmission from one class to another often occur, such as the WSVM, DH, NNPU, LDCE on *TwoMoons* dataset, and the OCSVM, DH, NNPU-Linear, LDCE on *Roll&Plane*. OCSVM can only

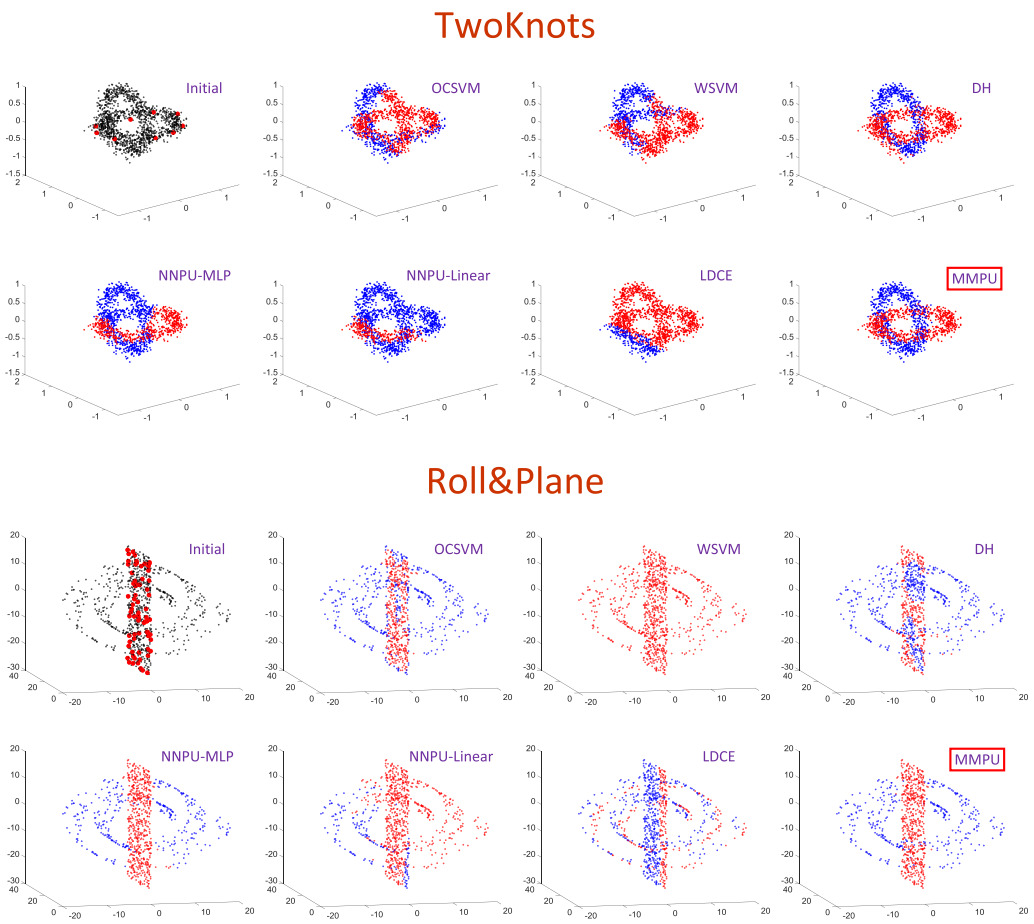

Fig. 2. The performance comparison of various methods on 3D datasets. In each dataset, the "initial" subfigure presents the initial annotation of the corresponding dataset, where the red dots represent positive examples and the black dots denote unlabeled examples. In the output of every compared method, the red dots and blue dots denote the determined positive examples and negative examples, respectively.

identifies a small number of positive examples which are enclosed by the regions supported by $\mathcal{P}$, therefore the massive positive examples in $\mathcal{U}$ are missing. Besides, the manifolds reflected by some of the compared methods are discontinues, such as OCSVM, DH and LDCE on *TwoSpirals*, where the data points along one continues manifold is often incorrectly interrupted by the examples of the other manifold. In addition, WSVM performs worse than any other approach on *TwoLines* and *TwoSpirals* datasets as it mistakenly classifies all data to the positive class on both datasets. In contrast, only our MMPU can discover the multiple manifolds and achieve perfect classification on both 2D and 3D datasets. Overall, we see that the existing methods cannot capture the underlying multi-manifold structure hidden in the dataset, so they are very likely to introduce classification errors along the manifolds or at the intersecting regions. Due to the capability of exploring multiple manifolds, our MMPU will not affected by the manifold intersections and is able to successfully distinguish the points belonging to different manifolds.

Such advantage of MMPU is also statistically verified by the quantitative comparisons in Tab. I, from which we can clearly observe that MMPU is much better than other methods on all datasets, and the average accuracy of MMPU is almost 9% higher than the second best method (i.e. NNPU-MLP). NNPU-MLP can achieve comparable results with our MMPU on *TwoLines* and *Roll&Plane* datasets, but it is significantly inferior to MMPU on the remaining three synthetic datasets. Therefore, we see that proper exploitation of multi-manifold is critical for our method to reach high classification accuracy.

B. Handwritten Digit Recognition

It has been widely acknowledged that the representations of handwritten digits follow a concise manifold structure [62]. Therefore, in this section, we apply our MMPU method to handwritten digit recognition, and compare its performance with OCSVM, WSVM, DH, NNPU-MLP, NNPU-Linear and LDCE. Specifically, we adopt the *MNIST* dataset for our experiment. The *MNIST* dataset contains 70,000 digit images across ten classes (i.e. "0"∼"9"), and the resolution of every image is 28×28. We adopt the GIST descriptor to represent these images and thus each image example is characterized by a 512-dimensional feature vector.

Considering that two pairs of numbers including "2" vs. "7" and "6" vs. "9" are quite similar which easily lead to the confusion of various classifiers, we conduct two sets of experiments on *MNIST* by applying all compared methods to distinguish between "2" and "7", and "6" and "9" in

TABLE I

QUANTITATIVE EXPERIMENTAL RESULTS OF COMPARED METHODS ON FIVE SYNTHETIC DATASETS. THE AVERAGE RESULTS OVER THE ALGORITHM OUTPUTS ON THE FIVE DATASETS ARE ALSO REPORTED. THE BEST RECORD ON EACH DATASET IS HIGHLIGHTED IN BOLD

Datasets	*TwoMoons*	*TwoLines*	*TwoSpirals*	*TwoKnots*	*Roll&Plane*	Average
OCSVM [60]	89.38	90.05	83.50	69.35	82.89	83.03
WSVM [19]	88.91	50.00	49.70	64.70	55.56	61.77
DH [2]	85.47	82.09	87.00	94.17	71.56	84.06
NNPU-MLP [3]	95.47	97.76	72.00	86.20	92.00	88.69
NNPU-Linear [3]	85.16	72.64	58.60	68.93	64.89	70.04
LDCE [18]	91.09	50.00	54.70	51.39	44.33	58.30
MMPU (Ours)	**99.38**	**98.26**	**99.20**	**100.00**	**92.56**	**97.88**

Fig. 3. Some examples of *MNIST2&7* and *MNIST6&9* datasets.

TABLE II

THE ACCURACIES OF VARIOUS METHODS ON *MNIST2&7* DATASET WHEN 30% AND 60% POSITIVE EXAMPLES ARE LABELED. THE BEST RECORD UNDER EACH NOISE LEVEL IS MARKED IN **BOLD**

Methods \ t	30%	60%
OCSVM [60]	0.920 ± 0.001	0.921 ± 0.005
WSVM [19]	0.887 ± 0.019	0.962 ± 0.002
DH [2]	0.931 ± 0.040	0.977 ± 0.005
NNPU-MLP [3]	0.974 ± 0.009	0.961 ± 0.061
NNPU-Linear [3]	0.975 ± 0.024	0.972 ± 0.014
LDCE [18]	0.972 ± 0.010	0.976 ± 0.017
MMPU (ours)	**0.994 ± 0.001**	**0.995 ± 0.001**

the original *MNIST* dataset. The two sets of experiments are denoted as "*MNIST2&7*" and "*MNIST6&9*" which have totally 14283 and 13834 examples, respectively. Some example images of the two datasets are presented in Fig. 3. For each experiment, we randomly select $t = 30\%$ and $t = 60\%$ positive examples to form the positive set $\mathcal{P}$ and incorporate the remaining 70% and 40% positive examples as well as all negative examples to the unlabeled set $\mathcal{U}$. Therefore, it can be seen that the case of $t = 30\%$ is generally more challenging then $t = 60\%$ as a large proportion of positive examples are hidden in the unlabeled set $\mathcal{U}$. Under each labeling rate t, we conduct five-fold cross validation on all compared methods and report their average test accuracies and standard deviations over the five independent implementations. To achieve fair comparison, the selected positive examples and the dataset splits are kept identical for all the compared methodologies.

The parameters of all investigated models have been carefully tuned. The Gaussian kernel width for OCSVM is set to the optimal value 1 by searching the grid $\{10^{-3}, 10^{-2}, \cdots, 10^{3}\}$. In NNPU-MLP and NNPU-Linear, the parameter β for preventing overfitting is tuned to 0.5. In LDCE, the regularization parameter λ and tolerance β are set to 10 and 0.1, respectively. For the proposed MMPU, we build the 10-NN graph, and the regularization parameter λ is set to 1.

The average accuracies achieve by all compared methods on *MNIST2&7* and *MNIST6&9* are presented in Tab. II and Tab. III, respectively. We can see that our MMPU achieves very impressive results on these two datasets, and the obtained accuracies are above 99% when the labeling rate $t = 30\%$ and $t = 60\%$. In contrast, OCSVM generally performs worse than any other PU model as it ignores the unknown positive examples in $\mathcal{U}$ and only assumes that the positive class is revealed by the data in $\mathcal{P}$. DH and NNPU-MLP are superior to WSVM and NNPU-Linear because they are able to generate nonlinear decision boundaries which are close to the optimal solution. However, they are still inferior to our MMPU as they are not capable of discovering the multi-manifold structure hidden in the dataset. Another important observation from Tabs. II and III is that although only a small fraction of the original positive examples are revealed (*e.g.* 30%), our MMPU algorithm can still touch very high classification accuracy, which again demonstrates the importance of utilizing manifold property of dataset for PU learning.

C. Violent Behavior Detection

Recently, there is a surge of research interest in detecting the violent behavior in a video sequence as this technique is very useful for protecting public safety. Therefore, we adopt

TABLE III

THE ACCURACIES OF VARIOUS METHODS ON *MNIST6&9* DATASET WHEN 30% AND 60% POSITIVE EXAMPLES ARE LABELED. THE BEST RECORD UNDER EACH NOISE LEVEL IS MARKED IN **BOLD**

Methods \ t	30%	60%
OCSVM [60]	0.939 ± 0.003	0.942 ± 0.004
WSVM [19]	0.960 ± 0.010	0.983 ± 0.002
DH [2]	0.965 ± 0.007	0.972 ± 0.048
NNPU-MLP [3]	0.966 ± 0.026	0.981 ± 0.017
NNPU-Linear [3]	0.958 ± 0.002	0.977 ± 0.003
LDCE [18]	0.963 ± 0.004	0.963 ± 0.005
MMPU (ours)	**0.999 ± 0.001**	**0.999 ± 0.001**

Fig. 4. Some example video frames in the *HockeyFight* dataset.

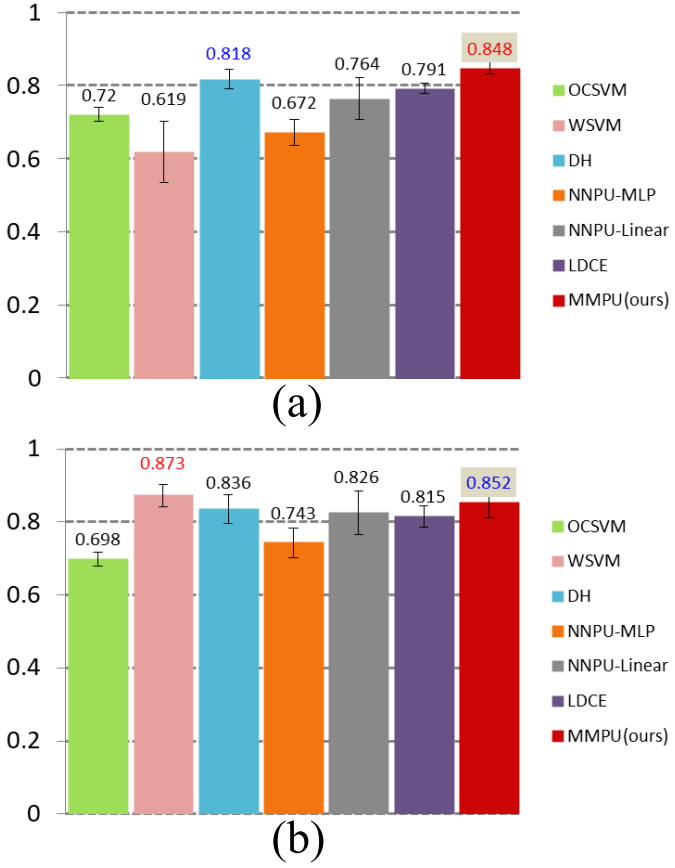

Fig. 5. The test accuracies of various methods on *HockeyFight* dataset. (a) and (b) show the results when $t = 30\%$ and $t = 60\%$, respectively. The best record under each t is highlighted in red, and the second best record is indicated in blue.

the *HockeyFight*[1] dataset to test the ability of all methods in recognizing the fight behavior. The adopted *HockeyFight* dataset contains 1000 video clips collected in ice hockey competitions, of which 500 contain fight behavior and 500 are normal non-fight sequences. The task of our experiment is to determine whether the fight behavior appears in given video clip of this dataset. From the examples provided in Fig. 4, we see that this dataset is quite challenging, as the crowded scenes with multiple interactive athletes may not represent the fighting behavior, while the scenes with only two athletes may contain the violent fighting activity. By following [50] and [63], we employ the space-time interest point (STIP) and motion SIFT (MoSIFT) as action descriptors, and thus each video clip of the dataset can be represented as a histogram over 100 visual words by further using the Bag-of-Words (BoW) quantization. Therefore, every video clip in this dataset is characterized by a 100-dimensional feature vector.

Similar to the experiment in Section V-B, here we also investigate the performances of all compared methods on two different labeling rates such as $t = 30\%$ and $t = 60\%$, and conduct five-fold cross validation on these methods so that they are trained on 80% video examples and tested on the remaining 20% examples. The dataset splits are kept identical for all methods including OCSVM, WSVM, DH, NNPU-MLP, NNPU-Linear, LDCE and our MMPU. The average test accuracies achieved by them are compared in Fig. 5. From Fig. 5(a), we notice that MMPU touches the highest recognition accuracy 84.8% when $t = 30\%$, which leads the second best DH with a margin of 3%. This indicates that MMPU is effective when the labeled positive examples are scarce. When the number of positive examples increases so that the labeling rate $t = 60\%$, the situation becomes simpler than the setting of $t = 30\%$, as more positive examples are observed. As a result, the accuracy of WSVM can be boosted to 87.3%, which is slightly better than MMPU as revealed by Fig. 5(b). The reason for the good performance of WSVM under $t = 60\%$ is that the weights of an example regarding positive class and negative class can be precisely estimated if sufficient positive examples are disclosed, and thus a substantial performance gain can be observed. Overall, our proposed MMPU is among the top two methods on the *HockeyFight* dataset, which is a very impressive result.

Furthermore, we also investigate the parametric sensitivity of our method to the two key tuning parameters K and λ, where K governs the number of neighbors for graph construction and λ determines the weight of model complexity regularizer in (20). Specifically, we change one of them from small to large while keeping the other one to a constant value, and then examine the average test accuracy output by MMPU. Both situations of $t = 30\%$ and $t = 60\%$ are studied, and the results are presented in Fig. 6. From this figure, it can be clearly observed that the performance of MMPU is not sensitive to the variations of these two parameters within a wide range, so these parameters can be easily tuned for practical applications.

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

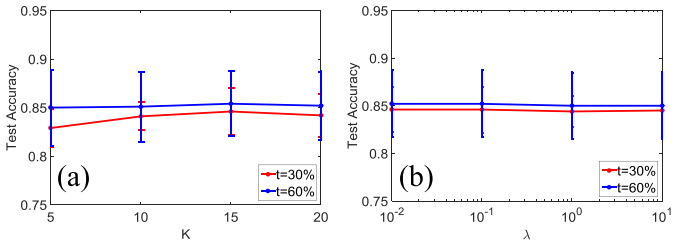

Fig. 6. The parametric sensitivity of (a) K and (b) λ of our MMPU under the labeling rates $t = 30\%$ and $t = 60\%$.

TABLE IV

THE ACCURACIES OF VARIOUS METHODS ON *Japserridge* DATASET WHEN 30% POSITIVE EXAMPLES ARE LABELED. THE BEST AND SECOND BEST RECORDS FOR EACH LAND TYPE IS MARKED IN RED AND BLUE, RESPECTIVELY

Methods \ Land type	soil	water	tree	road	average
OCSVM [60]	0.910	0.835	0.875	0.940	0.889
WSVM [19]	0.826	0.804	0.664	0.877	0.793
DH [2]	0.896	0.995	0.809	0.761	0.866
NNPU-MLP [3]	0.982	0.997	0.757	0.924	0.915
NNPU-Linear [3]	0.975	0.997	0.757	0.925	0.913
LDCE [18]	0.619	0.937	0.648	0.865	0.767
MMPU (ours)	0.963	0.996	0.920	0.958	0.959

TABLE V

THE ACCURACIES OF VARIOUS METHODS ON *JapserRidge* DATASET WHEN 60% POSITIVE EXAMPLES ARE LABELED. THE BEST AND SECOND BEST RECORDS FOR EACH LAND TYPE IS MARKED IN RED AND BLUE, RESPECTIVELY

Methods \ Land type	soil	water	tree	road	average
OCSVM [60]	0.790	0.770	0.829	0.943	0.833
WSVM [19]	0.887	0.992	0.856	0.983	0.929
DH [2]	0.879	0.993	0.795	0.785	0.863
NNPU-MLP [3]	0.961	0.996	0.757	0.925	0.910
NNPU-Linear [3]	0.959	0.995	0.757	0.924	0.909
LDCE [18]	0.677	0.985	0.701	0.924	0.822
MMPU (ours)	0.967	0.996	0.923	0.956	0.960

D. Hyperspectral Image Classification

Hyperspectral image classification is an important task in remote sensing area, of which the target is to classify every pixel into one of several pre-defined land types (i.e. classes) according to its spectral feature. Most existing works aim to identify all appeared land types in an hyperspectral image, however in some cases we only need to figure out one specific land category for certain task-driven purpose. As mentioned in the introduction, at this time we may simply treat the class of interest as positive and take the remaining pixels (i.e. examples) as unlabeled.

For our experiment, we adopt a typical hyperspectral dataset *JapserRidge* for model evaluation. This dataset is formed by a

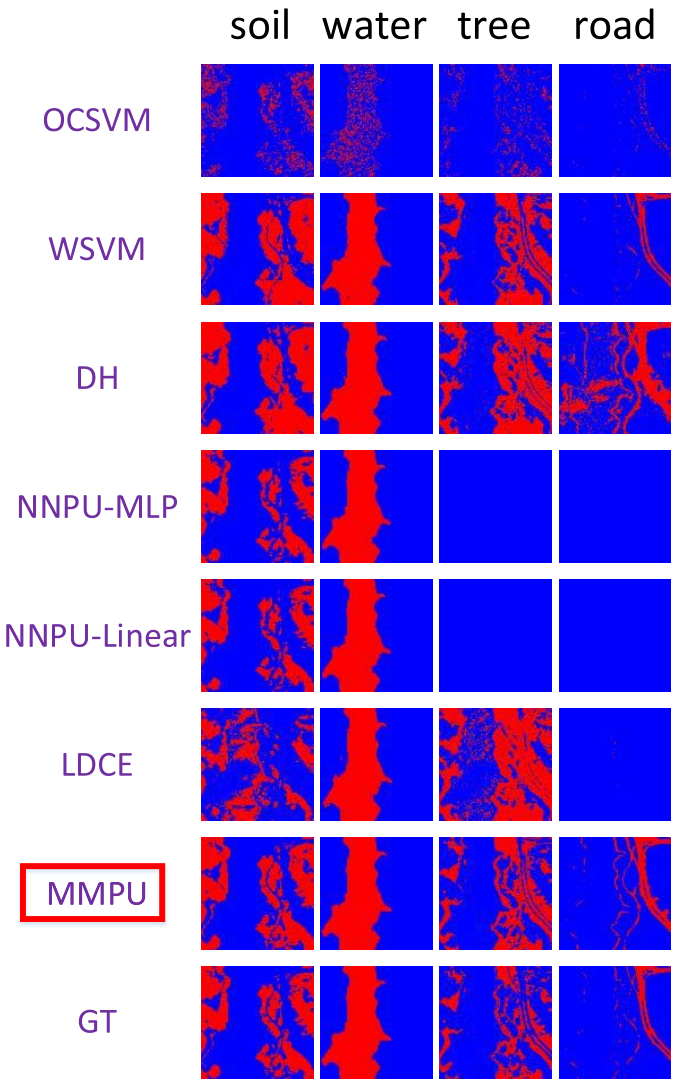

Fig. 7. The experimental results of compared methods on *JasperRidge* dataset in terms of four classes such as "soil", "water", "tree", and "road" with $t = 60\%$. The determined positive pixels and negative pixels are represented in red and blue, respectively. "GT" is short for groundtruth.

remotely sensed image with the resolution of 100×100, and each pixel is recorded by totally 198 spectral channels ranging from 380 nm to 2500 nm. This scenery contains four categories including "tree", "soil", "water", and "road", therefore every method should run four times by taking each of the four classes as positive. Similar to above experiments, we also investigate two cases where the labeling rate $t = 30\%$ and $t = 60\%$.

The accuracies of all methods on classifying the four categories and their averages are presented in Tabs. IV and V. We can see that our MMPU generally lies in the first or second place among the compared methods regarding different classes, therefore its averaged accuracy on all classes is higher than any other comparator. Although NNPU-MLP and WSVM can obtain the top level results on some classes, their outputs are not stable as they cannot consistently produce high accuracy on all categories. Consequently, they are worse than MMPU in terms of average accuracy. Furthermore, the classification results generated by different approaches when $t = 60\%$

are visualized in Fig. 7. We see that OCSVM cannot yield complete regions as it assumes that the positive class is only represented by the examples in $\mathcal{P}$ and thus largely ignoring the potential positive pixels in $\mathcal{U}$. NNPU-MLP and NNPU-Linear cannot identify tree and road regions as all pixels are classified as negative by them. The road region is also neglected by LDCE as the road pixels only account for a small fraction of total image pixels. In contrast, the outputs of our MMPU are very close to the groundtruth on all classes, and the small regions in each class are also precisely detected.

VI. Conclusion

This paper proposes a novel PU learning algorithm termed MMPU based on the multi-manifold assumption which is ubiquitous in many typical computer vision problems. By assuming that the data points of different classes lie on different manifolds, MMPU is quite effective in discovering the underlying structure of datasets which helps to boost the model discriminability. By comparing MMPU with representative state-of-the-art PU methods on both synthetic and practical datasets, the superiority of the developed MMPU can be easily observed. In the future, we may devise an acceleration technique to efficiently find the multi-manifold structure as the involved EM step for tangent space computation is relatively slow. Besides, since the multi-manifold property can often be found in multi-view learning [64] and multi-label learning [65], we may further discover their relationship and apply our method to more learning paradigms.

References

[1] G. Niu, M. C. D. Plessis, T. Sakai, Y. Ma, and M. Sugiyama, "Theoretical comparisons of positive-unlabeled learning against positive-negative learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1199–1207.

[2] M. C. D. Plessis, G. Niu, and M. Sugiyama, "Convex formulation for learning from positive and unlabeled data," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1386–1394.

[3] R. Kiryo, G. Niu, M. C. D. Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1674–1684.

[4] E. Sansone, F. G. B. De Natale, and Z.-H. Zhou, "Efficient training for positive unlabeled learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018. doi: 10.1109/TPAMI.2018.2860995.

[5] W. Li, Q. Guo, and C. Elkan, "A positive and unlabeled learning algorithm for one-class classification of remote-sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 2, pp. 717–725, Feb. 2011.

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

[7] C. Gong, D. Tao, X. Chang, and J. Yang, "Ensemble teaching for hybrid label propagation," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 388–402, Feb. 2019.

[8] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3249–3260, Jul. 2016.

[9] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 2, 2002, pp. 387–394.

[10] H. Yu, J. Han, and K. C.-C. Chang, "PEBL: Positive example based learning for web page classification using SVM," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: ACM, 2002, pp. 239–248.

[11] X. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 3, 2003, pp. 587–592.

[12] J. J. Rocchio, "Relevance feedback in information retrieval," in *The SMART Retrieval System: Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1971, pp. 313–323.

[13] B. Frënay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.

[14] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2013, pp. 1196–1204.

[15] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5552–5560.

[16] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabeled examples," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2003, pp. 179–186.

[17] W. Lee and B. Liu, "Learning with positive and unlabeled examples using weighted logistic regression," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 3, 2003, pp. 448–455.

[18] H. Shi, S. Pan, J. Yang, and C. Gong, "Positive and unlabeled learning via loss decomposition and centroid estimation," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 2689–2695.

[19] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM, 2008, pp. 213–220.

[20] M. C. D. Plessis, G. Niu, and M. Sugiyama, "Analysis of learning from positive and unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 703–711.

[21] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, "Multi-manifold deep metric learning for image set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1137–1145.

[22] C.-S. Lee, A. Elgammal, and M. Torki, "Learning representations from multiple manifolds," *Pattern Recognit.*, vol. 50, pp. 74–87, Feb. 2016.

[23] X. Zhang, M. Ding, and G. Fan, "Video-based human walking estimation using joint gait and pose manifolds," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 7, pp. 1540–1554, Jul. 2017.

[24] J. Lu, Y.-P. Tan, and G. Wang, "Discriminative multimanifold analysis for face recognition from a single training sample per person," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 39–51, Jan. 2013.

[25] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–6.

[26] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Combining multiple manifold-valued descriptors for improved object recognition," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl.*, Nov. 2013, pp. 1–6.

[27] S. Qiu, F. Nie, X. Xu, C. Qing, and D. Xu, "Accelerating flexible manifold embedding for scalable semi-supervised learning," *IEEE Trans. Circuits Syst. Video Technol.*, 2018. doi: 10.1109/TCSVT.2018.2869875.

[28] J. Zhang and Y. Peng, "SSDH: Semi-supervised deep hashing for large scale image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 212–225, Jan. 2019.

[29] T. Ishida, G. Niu, and M. Sugiyama, "Binary classification from positive-confidence data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5919–5930.

[30] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

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

[32] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[33] H. Hu, "Sparse discriminative multimanifold grassmannian analysis for face recognition with image sets," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 10, pp. 1599–1611, Oct. 2015.

[34] Y. Wang, Y. Jiang, Y. Wu, and Z.-H. Zhou, "Multi-manifold clustering," in *Proc. Pacific Rim Int. Conf. Artif. Intell. (PRICAI)*. Daegu, South Korea: Springer, 2010, pp. 280–291.

[35] Y. Wang, Y. Jiang, Y. Wu, and Z.-H. Zhou, "Spectral clustering on multiple manifolds," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1149–1161, Jul. 2011.

[36] D. Gong, X. Zhao, and G. Medioni. (2012)."Robust multiple manifolds structure learning." [Online]. Available: https://arxiv.org/abs/1206.4624

[37] E. Arias-Castro, G. Lerman, and T. Zhang, "Spectral clustering based on local PCA," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 253–309, 2017.

[38] B. Li, J. Li, and X.-P. Zhang, "Nonparametric discriminant multi-manifold learning for dimensionality reduction," *Neurocomputing*, vol. 152, pp. 121–126, Mar. 2015.
[39] J. Valencia-Aguirre and A. Álvarez-Meza, G. Daza-Santacoloma, C. Acosta-Medina, and C. G. Castellanos-Domínguez, "Multiple manifold learning by nonlinear dimensionality reduction," in *Proc. Iberoamerican Congr. Pattern Recognit.* Pucón, Chile: Springer, 2011, pp. 206–213.
[40] R. Hettiarachchi and J. F. Peters, "Multi-manifold lle learning in pattern recognition," *Pattern Recognit.*, vol. 48, no. 9, pp. 2947–2960, 2015.
[41] A. Goldberg, X. Zhu, A. Singh, Z. Xu, and R. Nowak, "Multi-manifold semi-supervised learning," in *Proc. Artif. Intell. Statist.*, Clearwater Beach, FL, USA, 2009, pp. 169–176.
[42] X. Xing, Y. Yu, H. Jiang, and S. Du, "A multi-manifold semi-supervised Gaussian mixture model for pattern classification," *Pattern Recognit. Lett.*, vol. 34, no. 16, pp. 2118–2125, 2013.
[43] Y. Xiao, B. Liu, J. Yin, L. Cao, C. Zhang, and Z. Hao, "Similarity-based approach for positive and unlabeled learning," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2011, vol. 22, no. 1, pp. 1577–1582.
[44] G. Li, "A survey on postive and unlabelled learning," Tech. Rep., 2013. [Online]. Available: https://www.eecis.udel.edu/ vijay/fall13/snlp/lit-survey/PositiveLearning.pdf
[45] F. He, T. Liu, G. I. Webb, and D. Tao. (2018). "Instance-dependent PU learning by bayesian optimal relabeling." [Online]. Available: https://arxiv.org/abs/1808.02180
[46] T. Gong, G. Wang, J. Ye, Z. Xu, and M. Lin, "Margin based PU learning," in *Proc. 22nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 1–8.
[47] M. Hou, B. Chaib-draa, C. Li, and Q. Zhao, "Generative adversarial positive-unlabelled learning," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*. Menlo Park, CA, USA: AAAI Press, 2018, pp. 2255–2261.
[48] A. Kanehira and T. Harada, "Multi-label ranking from positive and unlabeled data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5138–5146.
[49] T. Sakai, M. C. D. Plessis, G. Niu, and M. Sugiyama. (2016). "Semi-supervised classification based on classification from positive and unlabeled data." [Online]. Available: https://arxiv.org/abs/1605.06955
[50] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph Laplacian for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2261–2274, Oct. 2015.
[51] C. Gong, T. Liu, Y. Tang, J. Yang, J. Yang, and D. Tao, "A regularization approach for instance-based superset label learning," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 967–978, Mar. 2018.
[52] C. Gong, K. Fu, A. Loza, Q. Wu, J. Liu, and J. Yang, "PageRank tracker: From ranking to tracking," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 882–893, Jun. 2014.
[53] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 321–328.
[54] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang, "Label propagation via teaching-to-learn and learning-to-teach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1452–1465, Jun. 2017.
[55] T. Ishida, G. Niu, and M. Sugiyama, "Binary classification from positive-confidence data," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 1–12.
[56] T. Hofmann, B. Schölkopf, and A. J. Smola. (2005). *A Tutorial Review of RKHS Methods in Machine Learning*. [Online]. Available: http://alex.smola.org/papers/2005/unpubHofSchSmo05.pdf
[57] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.
[58] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1320–1327.
[59] C. Gong, D. Tao, K. Fu, and J. Yang, "Fick's law assisted propagation for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2148–2162, Sep. 2015.
[60] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, Dec. 2001.
[61] M. C. D. Plessis, G. Niu, and M. Sugiyama, "Class-prior estimation for learning from positive and unlabeled data," in *Proc. Asian Conf. Mach. Learn. (ACML)*, 2015, pp. 221–236.
[62] G. E. Hinton, P. Dayan, and M. Revow, "Modeling the manifolds of images of handwritten digits," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 65–74, Jan. 1997.
[63] L. Xu, C. Gong, J. Yang, Q. Wu, and L. Yao, "Violent video detection based on mosift feature and sparse coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2014, pp. 3538–3542.
[64] J. Li, C. Xu, W. Yang, C. Sun, and D. Tao, "Discriminative multi-view interactive image re-ranking," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3113–3127, Jul. 2017.
[65] S. You, C. Xu, Y. Wang, C. Xu, and D. Tao, "Privileged multi-label learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 3336–3342.

Chen Gong received the B.E. degree from the East China University of Science and Technology (ECUST) in 2010, and a dual Ph.D. degree from Shanghai Jiao Tong University (SJTU) and the University of Technology Sydney (UTS) in 2016 and 2017, respectively. He is currently a Full Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. He has published more than 50 technical papers in prominent journals and conferences, such as the IEEE TNNLS, the IEEE TIP, the IEEE TCYB, the IEEE TCSVT, the IEEE TMM, the IEEE TITS, CVPR, AAAI, IJCAI, and ICDM. His research interests mainly include machine learning, data mining, and learning-based vision problems. He serves as a Reviewer for more than 20 international journals, such as AIJ, the IEEE TNNLS, and the IEEE TIP, and also as a PC member of several top-tier conferences, such as ICML, AAAI, IJCAI, ICDM, and AISTATS. He received the "Excellent Doctoral Dissertation" awarded by SJTU and the Chinese Association for Artificial Intelligence (CAAI). He was also enrolled on the "Summit of the Six Top Talents Program" of Jiangsu Province, China, and the "Young Elite Scientists Sponsorship Program" of the China Association for Science and Technology.

Hong Shi received the B.E. degree in computer science and technology from Liaoning University, Shenyang, China. She is currently pursuing the master's degree with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. Her current research interests include machine learning and learning-based vision problem.

Jie Yang received the Ph.D. degree from the Department of Computer Science, Hamburg University, Germany, in 1994. He is currently a Professor with the Institute of Image Processing and Pattern recognition, Shanghai Jiao Tong University, China. He has led many research projects, including that of the National Science Foundation and the 863 National High Tech. Plan, had one book published in Germany, and has authored more than 200 journal papers. His major research interests are object detection and recognition, data fusion and data mining, and medical image processing.

Jian Yang received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST) in 2002. In 2003, he was a Post-Doctoral Researcher with the University of Zaragoza. From 2004 to 2006, he was a Post-Doctoral Fellow with the Biometrics Centre, The Hong Kong Polytechnic University. From 2006 to 2007, he was a Post-Doctoral Fellow with the Department of Computer Science, New Jersey Institute of Technology. He is currently a Chang-Jiang Professor with the School of Computer Science and Technology, NUST. He has authored more than 200 scientific papers on pattern recognition and computer vision. His papers have been cited more than 5 000 times in the Web of Science and 13 000 times in the Google Scholar. His research interests include pattern recognition, computer vision, and machine learning. He is a fellow of IAPR. He is/was an Associate Editor of *Pattern Recognition*, *Pattern Recognition Letters*, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Neurocomputing*.