Probabilistic Semi-supervised Learning via Sparse Graph Structure Learning

Li Wang, Raymond Chan, and Tieyong Zeng

***Abstract*—We present a probabilistic semi-supervised learning (SSL) framework based on sparse graph structure learning. Different from existing SSL methods with either a predefined weighted graph heuristically constructed from the input data or a learned graph based on the locally linear embedding assumption, the proposed SSL model is capable of learning a sparse weighted graph from the unlabeled high-dimensional data and a small amount of labeled data, as well as dealing with the noise of the input data. Our representation of the weighted graph is indirectly derived from a unified model of density estimation and pairwise distance preservation in terms of various distance measurements, where latent embeddings are assumed to be random variables following an unknown density function to be learned and pairwise distances are then calculated as the expectations over the density for the model robustness to the data noise. Moreover, the labeled data based on the same distance representations is leveraged to guide the estimated density for better class separation and sparse graph structure learning. A simple inference approach for the embeddings of unlabeled data based on point estimation and kernel representation is presented. Extensive experiments on various data sets show the promising results in the setting of SSL compared with many existing methods, and significant improvements on small amounts of labeled data.**

***Index Terms*—Semi-supervised learning, latent variable model, graph structure learning, kernel learning**

I. Introduction

Semi-supervised learning (SSL) [1], [2] aims to improve the learning problem in the case that small amounts of labeled data and relatively large amounts of unlabeled data are available. SSL has been widely used in many machine learning applications when annotating training data is time-consuming, costly and error-prone.

A plenty of SSL algorithms have been proposed in the literature. They are built on various assumptions of the given data, including generative models [3], density-region approaches [4], [5], graph-based methods [1], [6], [7], and embedding learning [8], [9]. Among these, graph-based methods have received much attention [1], [2]. The fundamental assumption of the graph-based methods is that the data is embedded in a low-dimensional manifold that may be reasonably expressed by a graph, where each vertex is associated with an input data point and the weight of each edge represents the similarity between two vertices, so that nearby vertices are more likely to have the same labels. Label propagation [6] and manifold regularization [10] are two popular graph-based SSL methods. Besides, generative SSL models have the advantage to model the posterior distribution of latent variables with priors [3].

In graph-based SSL methods, weighted graphs are often constructed directly from the input data. The neighborhood graphs are commonly used, e.g., the K-nearest neighbor (NN) graph. A similarity matrix (often very sparse) is constructed from the adjacency matrix of the given graph based on some prefixed similarity functions such as binary variable 0 and 1 for disconnection and connection respectively, and the heat kernel in terms of the Euclidean distances of two points and the neighborhood connectivities [10]. Although these methods have been successfully applied to many SSL problems, it could be very sensitive due to the high dependence on the ad hoc weighted graph, which becomes unreliable since the curvature of manifold and the density of data points may be varied in different regions of the manifold [11]. Moreover, most distance-based manifold learning methods suffer from the curse of dimensionality, i.e., there is little difference in the distances of pairs of data points [12]. Furthermore, for data with noise, a precomputed neighborhood graph to approximate the manifold of data is not reliable any more. Hence, it is less robust to directly construct a neighborhood graph in a high-dimensional space.

Learning a graph from data recently becomes popular for SSL. The graph is either pre-optimized [13], [14], [15], [16], [17], [18] , or jointly optimized with SSL prediction models [19], [20] based on criteria such as locally linear embedding (LLE) [21]. However, it is well known that LLE has some inherent drawbacks [22], so these SSL methods also inherits these drawbacks. Moreover, structure learning methods [11], [23], [24] have been proposed for unsupervised learning, but they are seldom explored in SSL.

The information we can leverage to achieve a better SSL model is not restricted to the data and its small amounts of labels. Various label priors have been explored such as class mass normalization and label bidding as a post-processing step [6], and class balance constraint [7]. The discriminative expectation constraints estimated with labeled data are also studied in [3]. In the setting of unsupervised learning, dimensionality reduction methods such as t-SNE [25] and maximum variance unfolding (MVU) [26] achieve great success by assuming to preserve certain information, including the clustering assignment in a low-dimensional space and pairwise distances in the reproducing kernel Hilbert space (RKHS), respectively, in order to better explore the characteristics of the input data. These learning criteria are proved to be effective for unsupervised learning, so it might also be useful for SSL.

Li Wang is with the Department of Mathematics and Department of Computer Science and Engineering, University of Texas at Arlington, Texas, 76019 USA. e-mail: li.wang@uta.edu.

Raymond Chan is with the Department of Mathematics, City University of Hong Kong, Hong Kong. e-mail: rchan.sci@cityu.edu.hk.

Tieyong Zeng is with the Department of Mathematics, The Chinese University of Hong Kong, Hong Kong. e-mail: zeng@math.cuhk.edu.hk.

Moreover, these unsupervised learning methods provide the capability of exploratory analysis via data visualization with the natural interpretation. Hence, it is worth exploring these criteria for traditional SSL, even though embeddings of the input data have recently been explored in the paradigm of deep learning [8], [9].

In this paper, we propose a novel probabilistic semi-supervised learning framework by simultaneously taking the following crucially important factors into SSL:

1) A sparse similarity matrix is learned from data;
2) Low-dimensional embeddings for data visualization are optimized for learning problem;
3) Various priors of the data can be naturally incorporated.

Specifically, we propose to model the density distribution of latent variables given the input data with small amounts of labels. Various distance measurements can be employed to characterize the relationships between any two data points specifically for the target domain. The expectation distance preservation criterion over the density leads to the robust learning of a sparse similarity matrix for capturing the intrinsic manifold structure of data. Priors related to the density are incorporated including data noise model based on the shrinkage effect of pairwise distances, and the prior of low-dimensional embeddings. Supervised labels guide the learning of density distribution by constraining their embeddings to be close if their data points are of the same classes, and otherwise to be distant. The optimized low-dimensional embeddings are then uncovered from the learned density for data visualization.

The main contributions of this paper are listed as follows:

- A novel probabilistic SSL framework is proposed by learning a density function over low-dimensional latent variables from the input data. This framework as a Bayesian model is flexible to integrate various priors for characterizing the data in the target domain and modeling data noise.
- The distance preservation criterion and the class separability from a small amount of labeled data as the supervised information are integrated into the proposed SSL framework. The resulting model shows that 1) a weighted graph is obtained from the data with an optimized sparse similarity matrix and the guidance of supervised information; 2) low-dimensional embeddings are uncovered from the weighted graph or a kernel matrix; and 3) the embeddings are used to infer labels of unlabeled data for semi-supervised classification and data visualization.
- We conduct extensive experiments on synthetic and benchmark data sets by comparing with a variety of the state-of-the-art methods in SSL. Our experimental results show that our proposed model not only achieves encouraging classification results for SSL, but also leads to an optimized kernel matrix for extracting embeddings, which are built on a learned sparse similarity matrix.

The rest of the paper is organized as follows. We first briefly introduce various existing methods in Section II. In Section III, we propose a unified probabilistic SSL framework with distance preservation criterion, class separability criterion, and various priors, and then present an optimization algorithm to solve the reformulated problem. Extensive experiments are conducted in Section IV. We conclude this work in Section V.

II. Related Work

We briefly discuss our SSL setting and several existing methods that are most related to this work by illustrating different perspectives of learning paradigms including SSL, kernel learning, and graph structure learning.

The problem of SSL aims to learn a classifier based on both the labeled and unlabeled data [1], [2]. There are two SSL paradigms: transductive learning [4], [7], [27], [28] and inductive learning [8], [10], where transductive learning applies the classifier to unlabeled data during the training stage and the classifier does not generalize to unseen data, while inductive learning learns a parametric function to explicitly represent the classifier so that it is applicable to unseen data. For graph-based SSL methods, the graph can either be derived from data [28], or known as the external domain knowledge such as a knowledge graph [29] or a citation network [30]. In this paper, we mainly focus on the transductive SSL and the graph is unknown.

Various learning criteria have been explored by existing SSL algorithms. Transductive SVM [4], [5] maximizes the margin of classifier based on low-density separation assumption so that the classifier lies in a sparse area of the feature space. Graph-based SSL methods [10], [28] assume that the nearby vertices on the graph are more likely to have the same labels. Learning a kernel matrix for supervised classification problem have been widely studied, e.g., the multiple kernel learning (MKL) [31], and it is also extended for the graph-based SSL [32], where the spectrum of the graph Laplacian matrix derived from data is optimized to achieve a ridge regression model for SSL classification problems. The kernel matrix is dense so it is not interpretable for exploring the manifold structure of the input data as usually represented by a sparse graph.

Since the graph is the key to the success of the graph-based SSL methods, improving the quality of the graph has become one of the hot topics. For example, manually crafted graphs are often used. In constrained large margin local projection methods [33], [34], graphs are pre-constructed from a K-NN graph of the labeled data and pairwise constraints of the class labels such as must-link (ML) and cannot-link (CL) constraints.

Inspired by learning the graphs from data for better quality, the sparse representation techniques, e.g., LLE, are widely used as a separate step for graph construction. Instead of manually crafting graphs as in label propagation [27], linear neighborhood propagation method [35] learns a sparse graph using LLE, which is then incorporated into the label propagation for SSL. For large-scale data, an anchor graph is constructed based on anchor points and the weights are then obtained by local anchor embedding (LAE) [13]. It is lately improved by modifying either the graph construction step, e.g., imposing the absolute operator on the weights of equality constraints [14], constructing a multiple layer anchors with a pyramid-style structure [15], or the SSL model using the flexible manifold embedding [16]. In [17], a graph precomputed via LLE for the locality information is later

TABLE I
NOTATION AND DEFINITIONS

Notation	Definition
c	the number of class labels
m	the dimension of the latent space
$\mathbb{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}$	A labeled data set consists of n_l data points $\mathbf{x}_i \in \mathbb{R}^d$ and its label $y_i \in \{1, 2, \ldots, c\}$
$\mathbb{D}_u = \{\mathbf{x}_i\}_{i=n_l+1}^{n}$	An unlabeled data set
$Z = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$	A matrix of random variables for embedded data with the ith column $\mathbf{z}_i \in \mathbb{R}^m$ corresponding to $\mathbf{x}_i$
$z_{r,i}$	A random variable for the rth row and the ith column of Z
$Z = [\mathbf{f}_1, \ldots, \mathbf{f}_m]^T$	$\mathbf{f}_r$ is a column random vector corresponds to the rth row of Z, $\forall r = 1, \ldots, m$
ϕ, φ	Pairwise distance functions over two points in the latent space and input space, respectively
p, π, $\mathbb{E}$, KL	Density function, prior density and the expectation function, and KullbackLeibler divergence of two density functions
α	The Lagrangian multipliers and also the matrix representation of the learned graph

incorporated into MMD-Isomap [18] by preserving pairwise geodesic distance for SSL. Instead of the conventional graphs, a hypergraph is constructed using ℓ_1 sparse representation, and then the hyperedge weights and predicted labels are jointly optimized [36]. The graphs obtained by the above methods highly rely on the LLE, so they may not work well in the case that LLE assumption does not hold [22].

The joint optimization of LLE-type graph learning and SSL models have also been explored. In [19], local manifold structure learning and constrained concept factorization are jointly optimized so as to improve the representation and discriminating abilities by imposing the consistency among the data reconstruction, the learned representation, and the predicted labels. The nonnegative ℓ_2 regularized graph learning is simultaneously solved with the objective of the positive and negative label propagation [20] in kernel space for the improvement of semi-supervised classification [37].

Another approach for graph construction is that the coefficients from the low-rank representation are used to construct a graph for SSL [38]. In addition, metric learning is also used to learn the weights of a graph with the fixed connectivities, e.g., the weights parameterized by Gaussian kernel with Mahalanobis distance [39], [40]. These methods update weights of graphs instead of learning a sparse representation, so it is not easy to control the sparsity of the graph weights.

In unsupervised setting, kernel learning and graph structure learning have been studied to capture the intrinsic manifold structure of the input data. MVU [26] learns a kernel matrix by maximizing the variance of the kernel and simultaneously maintaining the pairwise distances over the set of neighbors. Maximum posterior manifold embedding (MPME) [41] learns a posterior distribution of low-dimensional latent variables by preserving the expectation distances over an unknown density distribution, so it is treated as a probabilistic version of MVU. MPME owns the advantages to easily incorporate prior information of data. Moreover, various graph structure learning algorithms [11], [23], [24] have been proposed to automatically derive a good weighted graph for dimensionality reduction and clustering. However, these learning criteria are seldom explored for SSL.

III. Probabilistic Semi-supervised Learning via Sparse Graph Structure Learning

In this section, we first present the motivation of the proposed work, and then give the detailed descriptions on model formulation, optimization method, and the inference of unlabeled data. For the ease of reference, we summarize some important notation and definitions in Table I, which will be used throughout the whole paper.

A. Motivation

As discussed in Section II, graphs are the key information extracted from the input data for graph-based SSL methods. Most commonly used graph construction approaches are considered as some variants of LLE, including 1) vertexes of graphs are defined in the original data space together with constraints, e.g., simplex constraints [13], [16] and linear constraints with absolute operator [14]; 2) vertexes of graphs are defined in some representation space [17], but the weights are learned from the input data; 3) vertexes of graphs are consistent in the input space, some representation space and label space [19], or kernel space and label space [37]. Hence, they inherit the assumption of LLE, that is, a manifold is formed by local patches which are nearly linear and overlap with one another. However, LLE has its intrinsic drawbacks [22]:

1) It is unavoidable to derive the non-uniform warps and folds if the sample density is low or the points are unevenly sampled.
2) It is very sensitive to noise.
3) The general metric is not easy to be incorporated except the inherent Euclidean distance.
4) It cannot guarantee two embedded points must be different if their corresponding input data points are different.
5) A single continuous manifold is assumed, so it is not proper for multi-class classification problems.

As a result, the graph-based SSL methods discussed in Section II face the similar drawbacks as LLE.

Inspired by overcoming the above drawbacks brought by LLE, we in this paper seek a completely different approach for learning graphs from the input data. We aim at estimating a density function for unknown embeddings given a dataset with certain priors. The drawback 1) will not be an issue if the manifold is built on a continuous density function instead of a set of drawn data points. Moreover, the probabilistic density estimation approach can easily incorporate priors such as noise model for drawback 2) and a general metric function for calculating proper pairwise distances between any two input data points for drawback 3). Furthermore, our approach is built on distance preservation criterion by simultaneously integrating labeled data so it does not assume any single continuous manifold. Hence, the drawbacks 4) and 5) will be resolved.

B. Model formulation

Let $\mathbb{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}$ be a labeled dataset, and $\mathbb{D}_u = \{\mathbf{x}_i\}_{i=n_l+1}^{n}$ be an unlabeled dataset, where n data points $\{\mathbf{x}_i\}_{i=1}^{n}$ and their associated labels $\{y_i\}_{i=1}^{n}$ are sampled from some unknown distribution. Every data point $\mathbf{x}_i, \forall i = 1, \ldots, n$ resides in $\mathbb{R}^d$ and its label y_i is assigned as one of the given set $\{1, 2, \ldots, c\}$ of c classes. There are only n_l known labels, although n data points are given. We aim to develop a novel probabilistic density estimation approach for SSL via graph structure learning based on $\mathbb{D}_l$ and $\mathbb{D}_u$ and then predict the labels of unlabeled data.

Our method is motivated from the idea of distance preservation [26], where pairwise distances between data points are preserved, so that the distances between two data points in the original space can be maintained for their corresponding embedded points. Based on this motivation, we assume that both labeled data and unlabeled data share some latent space in terms of distances. We further infer the class labels of unlabeled data through this latent space. Moreover, the given set of labeled data is leveraged to push away data points of different classes and pull data points of same classes as close as possible.

Denote the latent space by $\mathbb{R}^m$ where $m \leq d$. Let $\{\mathbf{z}_i\}_{i=1}^n$ with $\mathbf{z}_i \in \mathbb{R}^m, \forall i$, be the latent embeddings associated to the given data points $\{\mathbf{x}_i\}_{i=1}^n$. Now, the pairwise distance between two latent points is defined as

$$\phi(\mathbf{z}_i, \mathbf{z}_j) = ||\mathbf{z}_i - \mathbf{z}_j||_2^2, \forall i, j, \tag{1}$$

where the Euclidean distance is used in the latent space. As mentioned above, we assume that data points are sampled from some unknown distribution, so it is reasonable to assume there is a density function over the latent variables. To model the unknown distribution over latent variables, we treat $\mathbf{z}$ as a set of real values sampled from m random vectors of size n and denote its density function as

$$p(Z) = \prod_{r=1}^{m} p(\mathbf{f}_r), \tag{2}$$

where $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_n] = [\mathbf{f}_1, \ldots, \mathbf{f}_m]^T \in \mathbb{R}^{m \times n}$. In other words, the m random vectors $\{\mathbf{f}_r\}_{r=1}^m$ are independent and identically distributed by following $p(\mathbf{f}_r)$ with samples in $\mathbb{R}^n$. Since Z is assumed to be a matrix of $m \times n$ random variables, equation (1) now stands for a probability distribution instead of a distance metric. To model the distance metric in the probabilistic latent space, the expectation of (1) over the given density function of Z is used. As a result, we define the following function as the transformed distance in the probability space, $\forall i, j$,

$$\begin{aligned} \mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] &= \mathbb{E}\Big[||\mathbf{z}_i - \mathbf{z}_j||_2^2\Big] \\ &= \sum_{r=1}^{m} \int p(\mathbf{f}_r)(z_{r,i} - z_{r,j})^2 \mathrm{d}\mathbf{f}_r, \end{aligned} \tag{3}$$

where the second equality holds due to (2). It is clear that the Bayesian average of pairwise distances over the density of latent embeddings (3) can be more robust than the deterministic counterparts (1).

Given labeled data set $\mathbb{D}_l$, we aim to maximize the class separation by minimizing the within-class distances and maximizing the between-class distances. To achieve this goal, we propose to minimize the following function as

$$\mathcal{L}_{\mathbb{D}_l} = \frac{\nu}{|\mathcal{S}|} \sum_{(i,j)\in\mathcal{S}} \mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] - \frac{1-\nu}{|\mathcal{D}|} \sum_{(i,j)\in\mathcal{D}} \mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] \tag{4}$$

where $\mathcal{S} = \{(i,j)|y_i = y_j, \forall i,j = 1, \ldots, n_l\}$ and $\mathcal{D} = \{(i,j)|y_i \neq y_j, \forall i,j = 1, \ldots, n_l\}$ are pairs of labeled data points with same and different classes, respectively. $|\mathcal{S}|$ is the size of the set $\mathcal{S}$ (same notation for $|\mathcal{D}|$), and $\nu \in [0,1]$ is a parameter to regulate the strength of pushing and pulling operations on $\mathbb{D}_l$. We notice that similar criterion for leveraging supervised information can be found in the literature. In supervised learning, the learning criterion called class separability is widely used in linear discriminant analysis (LDA) [42]. In semi-supervised learning, the ML and CL constraints from labeled data are also explored [33], [34]. The key advantage of (4) is the robustness of the class separability criterion to the data noise because of the Bayesian average on the pairwise distances, which is not applicable for the above methods.

For all data points $\{\mathbf{x}_i\}_{i=1}^n$ from both $\mathbb{D}_l$ and $\mathbb{D}_u$, the pairwise distances need to be preserved in the latent space so as to build the bridge between $\mathbb{D}_l$ and $\mathbb{D}_u$, that is,

$$\mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] = \varphi(\mathbf{x}_i, \mathbf{x}_j), \forall i, j, \tag{5}$$

where φ is a task-specific distance function. The distance preservation provides a simple and natural way to incorporate various distance metrics. Some examples of various distance functions are shown as follows:

- the Euclidean distance:
$$\varphi(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||^2, \forall i, j, \tag{6}$$
- the cosine discrepancy:
$$\varphi(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{\mathbf{x}_i^T \mathbf{x}_j}{||\mathbf{x}_i||_2 ||\mathbf{x}_j||_2}, \forall i, j, \tag{7}$$
where cosine similarity is frequently used in document classification.
- the Gaussian kernel distance:
$$\varphi(\mathbf{x}_i, \mathbf{x}_j) = 2(1 - \kappa(\mathbf{x}_i, \mathbf{x}_j)), \forall i, j, \tag{8}$$
where κ is the Gaussian kernel function defined as $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\phi(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2})$ with bandwidth σ.

In some cases, it is known that the input data has some manifold structure, and the structure can be properly captured by a neighborhood graph such as the K-NN graph. Denote the neighbors of $\mathbf{x}_i$ by $\mathcal{E}_i$. We preserve these distances represented for the manifold given by

$$\mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] = \varphi(\mathbf{x}_i, \mathbf{x}_j), \forall i, j \in \mathcal{E}_i. \tag{9}$$

It is worth noting that the neighborhood graph here can be less sensitive to SSL than the prefixed graph used in existing graph-based methods since our method is able to impose the sparsity over a full connectivity graph. However, the neighbors of each data point can significantly reduce the computation complexity of the proposed method, which will be clarified in the following of this section.

To prevent $p(Z)$ from being arbitrary, we further constrain the unknown distribution to be close to a prior distribution

$$\pi(Z) = \prod_{r=1}^{m} \pi(\mathbf{f}_r) = \prod_{r=1}^{m} \mathcal{N}(\mathbf{f}_r | \mathbf{0}, \gamma I), \tag{10}$$

where I is the identity matrix of size $n \times n$ and $\gamma > 0$ is the bandwidth of the normal distribution with zero mean and γ variance. The noise of embeddings can be naturally modeled by density π. The above constraint can be achieved effectively by minimizing the KL-divergence between $p(Z)$ and $\pi(Z)$ given by

$$\begin{aligned} \mathrm{KL}(p(Z)||\pi(Z)) &= \int p(Z) \log \frac{p(Z)}{\pi(Z)} dZ \\ &= \sum_{r=1}^{m} \int p(\mathbf{f}_r) \log \frac{p(\mathbf{f}_r)}{\pi(\mathbf{f}_r)} \mathrm{d}\mathbf{f}_r. \end{aligned} \tag{11}$$

By combining the above three ingredients (4), (9), and (11), we propose the probabilistic semi-supervised learning by solving the following optimization problem

$$\begin{aligned} \min_{\{p(\mathbf{f}_r)_{r=1}^m\}} \; & \mathrm{KL}(p(Z)||\pi(Z)) + \lambda_2 \mathcal{L}_{\mathbb{D}_l} \\ \text{s.t. } & \mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] = \varphi(\mathbf{x}_i, \mathbf{x}_j), \forall i, j \in \mathcal{E}_i \\ & p(\mathbf{f}_r) \in \mathcal{P}_r \end{aligned} \tag{12}$$

where $\mathcal{P}_r$ is a feasible set of all density functions over $\mathbf{f}_r$ and $\lambda_2 > 0$ is the regularizaiton parameter.

In reality, data points are usually contaminated with noisy signals. Strictly preserving distances might not be the best choice. To tolerate the data noise, in this paper, we consider to learn a smooth skeleton structure of latent variables to represent the inherent manifold of the input data via the shrinkage approach [43] given by

$$\begin{aligned} \min_{\{p(\mathbf{f}_r)_{r=1}^m\}, \{\xi_{i,j}\}} \; & \mathrm{KL}(p(Z)||\pi(Z)) + \lambda_1 \sum_{i,j \in \mathcal{E}_i} \xi_{i,j} + \lambda_2 \mathcal{L}_{\mathbb{D}_l} \\ \text{s.t. } & \mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] \leq \varphi(\mathbf{x}_i, \mathbf{x}_j) + \xi_{i,j}, \forall i, j \in \mathcal{E}_i \\ & \xi_{i,j} \geq 0, \forall i, j \in \mathcal{E}_i \\ & p(\mathbf{f}_r) \in \mathcal{P}_r \end{aligned} \tag{13}$$

where λ_1 is the regularization parameter for shrinkage effect. Next, we transform (13) to its dual problem and present an optimization method to solve the dual problem.

C. Problem reformulation

As problem (13) involves the functional optimization variables, it is challenging to solve it directly. Fortunately, problem (13) is convex with respect to $\{p(\mathbf{f}_r)\}_{r=1}^m$ and $\{\xi_{i,j}\}$. Rather than solving (13), we consider to solve its dual problem by applying an equivalent transformation via the Lagrangian duality [44]. Specifically, we introduce multipliers $\{\alpha_{i,j} \geq 0\}$, $\{\beta_{i,j} \geq 0\}$. The Lagrangian function can be written as

$$\begin{aligned} & L(\{p(\mathbf{f}_r)_{r=1}^m\}, \{\xi_{i,j}\}, \{\alpha_{i,j}\}, \{\beta_{i,j}\}) \\ &= \sum_{r=1}^{m} \int p(\mathbf{f}_r) \log \frac{p(\mathbf{f}_r)}{\pi(\mathbf{f}_r)} \mathrm{d}\mathbf{f}_r + \lambda_1 \sum_{i,j \in \mathcal{E}_i} \xi_{i,j} - \sum_{i,j \in \mathcal{E}_i} \beta_{i,j} \xi_{i,j} \\ &+ \sum_{i,j \in \mathcal{E}_i} \alpha_{i,j} \left(\mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] - \varphi(\mathbf{x}_i, \mathbf{x}_j) - \xi_{i,j} \right) \\ &+ \lambda_2 \left(\frac{\nu}{|\mathcal{S}|} \sum_{(i,j) \in \mathcal{S}} \mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] - \frac{1-\nu}{|\mathcal{D}|} \sum_{(i,j) \in \mathcal{D}} \mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] \right). \end{aligned}$$

Here, we assume that the neighborhood graph is undirected, so the shrinkage constraints are symmetric. This leads to the symmetric multipliers, i.e., $\alpha_{i,j} = \alpha_{j,i}, \forall i, j$. Moreover, the distance function ϕ enforces the condition that the shrinkage constraint always holds, so $\alpha_{i,i} = 0, \forall i$ is optimal for (13). Let $\mathbf{1}$ be the vector of all ones. We derive the KKT conditions

$$\partial_{\xi_{i,j}} L = \lambda_1 - \alpha_{i,j} - \beta_{i,j} = 0, \forall i, j \in \mathcal{E}_i \tag{14}$$

$$\begin{aligned} & \partial_{p(\mathbf{f}_r)} L = \log p(\mathbf{f}_r) - \log \pi(\mathbf{f}_r) + 1 \\ & + \lambda_2 \left(\frac{\nu}{|\mathcal{S}|} \sum_{(i,j) \in \mathcal{S}} (z_{r,i} - z_{r,j})^2 - \frac{1-\nu}{|\mathcal{D}|} \sum_{(i,j) \in \mathcal{D}} (z_{r,i} - z_{r,j})^2 \right) \\ & + \sum_{i,j \in \mathcal{E}_i} \alpha_{i,j} (z_{r,i} - z_{r,j})^2 \\ & = \log p(\mathbf{f}_r) - \log \pi(\mathbf{f}_r) + 1 + 2\mathrm{Tr}(\mathbf{f}_r(\lambda_2 L_{SD} + L_A)\mathbf{f}_r^T) \\ & = 0, \forall r \end{aligned} \tag{15}$$

$$\int p(\mathbf{f}_r) \mathrm{d}\mathbf{f}_r = 1, p(\mathbf{f}_r) > 0, \forall r, \tag{16}$$

$$\alpha_{i,j} \left(\mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] - \varphi(\mathbf{x}_i, \mathbf{x}_j) - \xi_{i,j} \right) = 0, \forall i, j \in \mathcal{E}_i, \tag{17}$$

where $L_{SD} = \frac{\nu}{|\mathcal{S}|} L_S - \frac{1-\nu}{|\mathcal{D}|} L_D$, $L_S = \mathrm{diag}(S\mathbf{1}) - S$, $L_D = \mathrm{diag}(D\mathbf{1}) - D$, $L_A = \mathrm{diag}(A\mathbf{1}) - A$, with matrices S, D and A are defined as

$$S_{i,j} = \begin{cases} 1, & (i,j) \in \mathcal{S} \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

$$D_{i,j} = \begin{cases} 1, & (i,j) \in \mathcal{D} \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

$$A_{i,j} = \begin{cases} \alpha_{i,j}, & i, j \in \mathcal{E}_i \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

According to (15) and (16), we have the analytic solution

$$\begin{aligned} p(\mathbf{f}_r) & \propto \pi(\mathbf{f}_r) \exp(-2\mathrm{Tr}(\mathbf{f}_r(\lambda_2 L_{SD} + L_A)\mathbf{f}_r^T) \\ &= \frac{1}{\sqrt{2\pi}} \gamma^{\frac{n}{2}} \exp\left(-\frac{||\mathbf{f}_r||_2^2}{2\gamma} - 2\mathrm{Tr}(\mathbf{f}_r(\lambda_2 L_{SD} + L_A)\mathbf{f}_r^T \right) \\ &= (2\pi\gamma)^{-\frac{n}{2}} \exp\left(-\frac{1}{2} \mathrm{Tr}(\mathbf{f}_r(\frac{1}{\gamma} Q)\mathbf{f}_r^T) \right), \end{aligned} \tag{21}$$

where $Q = I + 4\gamma(\lambda_2 L_{SD} + L_A)$. According to (17), it is clear to see that $\alpha_{i,j} = 0$ if pairwise distance constraint is strictly unequal. As a result, the optimal solution of α should be sparser than the initial neighborhood graph. By substituting the above equations back to the Lagrangian function, we obtain the dual problem of (13) as

$$\max_{A\in\mathcal{A}} -\sum_{r=1}^{m}\Omega_r - \sum_{i,j\in\mathcal{E}_i}\alpha_{i,j}\varphi(\mathbf{x}_i,\mathbf{x}_j) \tag{22}$$

where the logarithm of partition term of density (21) is

$$\begin{aligned}\Omega_r &= \log\int(2\pi\gamma)^{-\frac{n}{2}}\exp\left(-\frac{1}{2}\mathrm{Tr}(\mathbf{f}_r(\frac{1}{\gamma}Q)\mathbf{f}_r^T)\right)d\mathbf{f}_r\\ &= -\frac{1}{2}\log\det(\frac{1}{\gamma}Q).\end{aligned} \tag{23}$$

Let $\mathcal{A}$ be the bounded feasible set for matrix A and matrix $A=[\alpha_{i,j}]$ with the (i,j)th element defined as

$$\alpha_{i,j}=\begin{cases}0, & j\notin\mathcal{E}_i \vee i=j\\ \alpha_{j,i}\in[0,\lambda_1], & \text{otherwise},\end{cases} \tag{24}$$

where the box constraints over $\alpha_{i,j}$ is obtained from (14) and the multipliers $\alpha_{i,j}\geq 0$ and $\beta_{i,j}\geq 0, \forall i,j\in\mathcal{E}_i$.

Finally, we obtain the simplified dual problem as

$$\begin{aligned}\min_{A\in\mathcal{A},Q\succ 0} \quad & -\frac{m}{2}\log\det(Q)+\sum_{i,j}\alpha_{i,j}\varphi(\mathbf{x}_i,\mathbf{x}_j)\\ \text{s.t.} \quad & Q=I+4\gamma(\lambda_2 L_{SD}+L_A).\end{aligned} \tag{25}$$

Problem (25) is the dual problem of (13) as a semidefinite programming (SDP) [44]. Next, we will present an efficient optimization method for solving (25).

D. Optimization algorithm

Due to the positive semidefinite constraint and sepcial structure of A, we propose to solve (25) using the alternating direction method of multipliers (ADMM) [45]. Specifically, we first formulate the proximal regularized Lagrangian function with multiplier R and parameter ρ given by

$$\begin{aligned}L_\rho(A,Q,\gamma,R)=&-\frac{m}{2}\log\det(Q)+\sum_{i,j}\alpha_{i,j}\varphi(\mathbf{x}_i,\mathbf{x}_j)\\ &-\langle R, Q-(I+\gamma(4\lambda_2 L_{SD}+4L_A))\rangle\\ &+\frac{\rho}{2}||Q-(I+\gamma(4\lambda_2 L_{SD}+4L_A))||_F^2\end{aligned}$$

According to [45], the following updates can be taken to solve (25) by iterating them until convergence:

$$\begin{aligned}A\leftarrow&\arg\min_{A\in\mathcal{A}}\sum_{i,j}\alpha_{i,j}\varphi(\mathbf{x}_i,\mathbf{x}_j)\\ &+\frac{\rho}{2}||Q-(I+\gamma(4\lambda_2 L_{SD}+4L_A))-\frac{1}{\rho}R||_F^2 && (26)\\ Q\leftarrow&\arg\min_{Q\succ 0}-\frac{m}{2}\log\det(Q)\\ &+\frac{\rho}{2}||Q-(I+\gamma(4\lambda_2 L_{SD}+4L_A))-\frac{1}{\rho}R||_F^2 && (27)\\ R\leftarrow& R-\rho(Q-(I+\gamma(4\lambda_2 L_{SD}+4L_A))). && (28)\end{aligned}$$

Below, we will show the method for solving each subproblem separately.

1) Solve problem (26): Define $4\gamma P=Q-I-4\lambda_2\gamma L_{SD}-\frac{1}{\rho}R$ and $\Psi=[\psi(\mathbf{x}_i,\mathbf{x}_j)]$. We have

$$\begin{aligned}f(A)&=\sum_{i,j}\alpha_{i,j}\varphi(\mathbf{x}_i,\mathbf{x}_j)+8\rho\gamma^2||L_A-P||_F^2\\ &=\langle A,\Psi\rangle+8\rho\gamma^2||\mathrm{diag}(A\mathbf{1})-A-P||_F^2.\end{aligned}$$

Let $U=\mathrm{diag}(A\mathbf{1})-A-P$. We obtain the first derivative with respect to $\alpha_{i,j}$ for $i<j$,

$$\begin{aligned}\frac{\partial f(A)}{\partial\alpha_{i,j}}&=8\rho\gamma^2\mathrm{Tr}\left(U^T\frac{\partial(\mathrm{diag}(A\mathbf{1})-A-P)}{\partial\alpha_{i,j}}\right)+2\varphi(\mathbf{x}_i,\mathbf{x}_j)\\ &=8\rho\gamma^2\mathrm{Tr}\left(U^T B_{i,j}\right)+2\varphi(\mathbf{x}_i,\mathbf{x}_j)\end{aligned} \tag{29}$$

due to the symmetric property of A where the (s,t) entry of matrix $B_{i,j}$ is given by

$$B_{i,j}^{s,t}=\begin{cases}1, & s=t=i \text{ or } s=t=j\\ -1, & s=i \text{ and } t=j \text{ or } s=j \text{ and } t=i\\ 0, & \text{otherwise}.\end{cases} \tag{30}$$

We can further simplify the computation of gradient for the upper triangular part of the symmetric matrix A as

$$\frac{\partial f(A)}{\partial\alpha_{i,j}}=8\rho\gamma^2(U_{i,i}+U_{j,j}-U_{i,j}-U_{j,i})+2\varphi(\mathbf{x}_i,\mathbf{x}_j).$$

With this reformulation, the total number of variables to be optimized in (26) is about $\frac{1}{2}\sum_{i=1}^n|\mathcal{E}_i|$. For a large n, the total number of variables will be a linear function of the total number of data points. Fortunately, problem (26) is convex with box constraints, so it can be solved efficiently for large-scale problems by existing methods such as L-BFGS-B [46].

2) Solve problem (27): Denote $C=I+4\gamma(\lambda_2 L_{SD}+L_A)+\frac{1}{\rho}R$ and eigen-decomposition $C=V\Sigma V^T$ with diagonal matrix $\Sigma_{i,i}=\sigma_i,\forall i$ and matrix V of orthonormal columns. The optimization problem (27) is reformulated as

$$\min_{Q\succ 0}-\frac{m}{2}\log\det(Q)+\frac{\rho}{2}||Q-C||_F^2, \tag{31}$$

which has the optimal solution [41]

$$Q=V\widehat{\Sigma}V^T:\widehat{\sigma}_i=\frac{\sigma_i}{2}+\sqrt{\frac{\sigma_i^2}{4}+\frac{\rho}{2m}},\forall i, \tag{32}$$

where $\widehat{\Sigma}$ is a diagonal matrix with the (i,i)th entry $\widehat{\sigma}_i$.

The adaptive update of ρ is adopted for fast convergence [45]. The convergence criteria of ADMM in [45] is employed.

E. Initialization

To solve problem (25) using ADMM, a good initializer can speed up the convergence. If we disable the supervised information in (13), i.e., $\lambda_2=0$, we have

$$\begin{aligned}\min_{\{p(\mathbf{f}_r)_{r=1}^m\},\{\xi_{i,j}\}} \quad & \mathrm{KL}(p(Z)||\pi(Z))+\lambda_1\sum_{i,j\in\mathcal{E}_i}\xi_{i,j}\\ \text{s.t.} \quad & \mathbb{E}\Big[\phi(\mathbf{z}_i,\mathbf{z}_j)\Big]\leq\phi(\mathbf{x}_i,\mathbf{x}_j)+\xi_{i,j},\forall i,j\in\mathcal{E}_i\\ & \xi_{i,j}\geq 0,\forall i,j\in\mathcal{E}_i\\ & p(\mathbf{f}_r)\in\mathcal{P}_r.\end{aligned} \tag{33}$$

Accordingly, we have its dual problem as

$$\min_{A\in\mathcal{A}} \quad -\frac{m}{2}\log\det(I+4\gamma L_A)+\sum_{i,j}\alpha_{i,j}\phi(\mathbf{x}_i,\mathbf{x}_j). \tag{34}$$

Algorithm 1 Structure Semi-supervised Learning (StructSSL)

```
1: Input: labeled data $\mathbb{D}_l$ and unlabeled data $\mathbb{D}_u$, neighbors $\mathcal{E}_i, \forall i$,
   reduced dimension $m$, parameters $\lambda_1$, $\lambda_2, \nu$, $\gamma$
2: compute distance measures using (6), or (7), or (8)
3: Initialize $A$ by solving (34)
4: repeat
5:     update $Q$ using (32)
6:     update $R$ using (28)
7:     obtain $A$ by solving (26) using L-BFGS-B
8: until Convergence
9: obtain $\widehat{Z}$ using KPCA
10: train a classifier on labeled embeddings and predict labels for
    unlabeled data
11: Output: embeddings $\widehat{Z}$, sparse graph matrix $A$, kernel matrix
    $Q^{-1}$ and the labels of unlabeled data
```

It is well-known that the graph Laplacian matrix L_A is positive semidefinite for $A \geq 0$. Hence, the semidefinite constraint in (34) is automatically satisfied for all $A \in \mathcal{A}$. As a result, (34) can also be solved by L-BFGS-B due to the similar formulation to subproblem (26). It is worth noting that problem (34) is the same as MPME [43], so our newly proposed model (25) is more general than MPME with the capability of semi-supervised learning and various distance measures.

F. Inferring the labels of unlabeled data via the learned graph

After Q is obtained, we recover the embeddings of the input points, and then conduct supervised classification based on the embedded points by training on labeled data only and predicting the labels of the unlabeled data.

First, we need to recover the latent variables $\{\mathbf{z}_i\}_{i=1}^n$ via point estimation based on (21), which is further written as a multivariate normal distribution

$$p(\mathbf{f}_r) = \mathcal{N}(\mathbf{0}, \gamma Q^{-1}). \tag{35}$$

The expectation of pairwise distance between two latent variables can be simplified as

$$\mathbb{E}\Big[\phi(\mathbf{z}_i, \mathbf{z}_j)\Big] = m\gamma\Big[Q^{-1}_{i,i} + Q^{-1}_{j,j} - 2Q^{-1}_{i,j}\Big], \forall i, j, \tag{36}$$

which leads to the point-based estimation $\{\widehat{\mathbf{z}}_i\}_{i=1}^n$ as

$$\widehat{\mathbf{z}}_i^T \widehat{\mathbf{z}}_j = m\gamma Q^{-1}_{i,j}, \forall i, j. \tag{37}$$

Interestingly, this is equivalent to the definition of a linear kernel over $\{\widehat{\mathbf{z}}_i\}_{i=1}^n$ according to the kernel trick [47]. For kernel-based classification method, this kernel can be directly used to measure the similarity between two input data points, such as support vector machines (SVMs) [47]. To further investigate the property of the latent variables such as the visualization for exploration analysis, it is natural to use KPCA [48] to uncover $\{\widehat{\mathbf{z}}_i\}_{i=1}^n$ from Q^{-1} by keeping the top m basis with the largest eigenvalues of the centralized matrix of Q^{-1}.

Given $\widehat{Z} = \{\widehat{\mathbf{z}}_i\}_{i=1}^n$, we can construct two new data sets including the labeled data $\widehat{\mathbb{D}}_l = \{(\widehat{\mathbf{z}}_i, y_i)\}_{i=1}^{n_l}$ and the unlabeled data $\widehat{\mathbb{D}}_u = \{\widehat{\mathbf{z}}_i\}_{i=n_l+1}^n$. Any classifiers such as support vector machines (SVM) [49] and K-NN classifier can be trained on the labeled data $\widehat{\mathbb{D}}_l$ and applied to make the prediction of unlabeled data $\widehat{\mathbb{D}}_u$. The proposed SSL algorithm is shown in Algorithm 1.

G. Computational Complexity Analysis

The computational cost of Algorithm 1 is mainly contributed by the following components. Denote by q the number of non-zeros in the lower triangular part of the initial neighborhood graph $\{\mathcal{E}_i\}_{i=1}^n$. In step 2, the computational cost for the distance measures of pairs of neighbors is $O(qd)$. In step 5, the eigenproblem is solved for Q, so the complexity is in $O(n^3)$. In step 6, updating R takes $O(n^2)$. In step 7, problem (26) is solved by L-BFGS-B, where the number of optimized variables is q. In L-BFGS-B, the computational complexity is contributed mostly by the following parts: 1) computing gradients with respect to A takes $4q$ since there is only 4 nonzeros in $B_{i,j}$; and 2) computing objective value takes $O(n^2)$. As discussed in [46], L-BFGS-B algorithm shares many features of quasi-Newton algorithms but it is very efficient for computing the approximate Newton's direction using efficient Hessian updates with limited memory footprint. In practice, it is very efficient for solving problem (26) with millions of variables. KPCA takes $O(n^3)$ to get the embeddings of the whole data in step 9. Except the cost of training classifier on the data of size $n_l \times m$ is negligible, the majority cost of Algorithm 1 is in the scale of $O(n^3)$. Due to the cubic computation complexity, Algorithm 1 is not scalable for large-scale data, but it works well for moderate-size datasets.

IV. Numerical Experiments

A. Experimental setting

We evaluate the performance of our proposed method in terms of three predefined distance measures, i.e., (6), (7) and (8), on one synthetic data set and five benchmark SSL data sets as shown in Table II, by comparing against various existing methods that are capable of conducting SSL and following the experimental setting used in [32], [50], [51]. The synthetic three-moon data is simulated with details shown in Section IV-B. Except Opt-digits from UCI machine learning repository[1], the other four datasets are available[2]. The descriptions of all benchmark data sets are given in Section IV-C. All compared methods in this paper are illustrated with their specific parameters (common parameters shared by various baselines will be discussed later for the concise representation), and are tuned in their own proper ranges in order to report their best results for fair comparisons:

- K-nearest neighbor classifier (K-NN): K is tuned in a wide range $\{1, 2, 4, 5, 10, 20, 30, 40, 50, 70, 90, 100, 120, 140, 150, 160, 180, 200\}$.
- Spectral Graph Transformation (SGT) [7]: the parameter c is searched in $\{10^3 a : a \in \{3, 3.2, 3.4, 3.8, 5, 100\}\}$.
- Laplacian Regularized Least Squares (LapRLS) [10]: two regularization parameters are tuned over $\gamma_A \in \{10^a : a \in \{-6, -4, -2, 0, 2\}\}$ and $\gamma_I \in \{10^a : a \in \{-\infty, -4, -2, 0, 2\}\}$.
- $\mathcal{P}_{SQ}$ using SQ-Loss-1 and Measure Propagation (MP) [50]: the trade-off parameters μ and ν are tuned over

[1] archive.ics.uci.edu/ml/datasets/optical+recognition+of+handwritten+digits

[2] olivier.chapelle.cc/ssl-book/benchmarks.html

$\{10^a : a \in \{-8,-6,-4,-2,0,1,2\}\}$ and $\{10^a : a \in \{-8,-6,-4,-2,0,1\}\}$, respectively.

- Multiclass Ginzburg-Landau energy (Multiclass GL) and Multiclass graph-based Merriman-Bence-Osher (Multiclass MBO) [52]: the convexity parameter $C = \mu + 1/\epsilon$ is used in multiclass GL, and diffusion step $N_S = 3$ is used before any thresholding. As claimed in the paper, other parameters are specially tuned for each data set.
- Laplacian-based multiclass graph partitioning with a region-force (LapRF) and TV-based multiclass graph partitioning with a region-force (TVRF) [51]. $m = 1$ considers the direct neighbors of the labeled points and $m = 2$ uses the second neighbors. Other parameters are tuned as suggested by the authors.
- SimpleMKL [31] and Spectral kernel learning (SKL) [32]. Two kernel learning methods can be used for semi-supervised learning. SimpleMKL learns a convex combination of multiple base kernels including Gaussian kernels with bandwidth in $\{0.5,1,2,5,7,10,12,15,17,20\}$ and polynomial kernels with degrees in $\{1,2,3\}$. The graph Laplacian in SKL is constructed from K-NN using heat kernel weights and its degree in $\{2,5\}$. For both methods, parameter C is turned in $\in \{0.01,0.1,1,10,100\}$.
- AGR [13] and f/r-FME [16]. f/r-FME methods extend FME by taking the use of LAE graph for scalable SSL. The number of anchor points are the ratio of the number of data tuned in range $\{0.01, 0.1 : 0.1 : 1\}$, where large anchor points are used in order to achieve good performance on mediate datasets. The tradeoff parameters μ and γ in FME are tuned in $[10^{-9}, 10^9]$. The rest of other parameters follow the experimental settings in [16].
- KernelLP [37], a joint optimization method for positive and negative label propagation and adaptive weights learning in kernel space. In the experiments, the Gaussian kernel is used and tuned as explained below. Both tradeoff parameters α and β are tuned in range $\{0.01,0.1,1,10,100\}$. The weighting factors $u^+ = 10^{10}$ and $u^- = 1$ are used for labeled data, and 0 for unlabeled data, as stated in [37].
- SSLRR [38]. Low rank representation is used for graph construction by incorporating label information of the labeled data. The parameter λ for balancing the effects of nuclear norm of the coefficient matrix and the sample-specific corruptions and regularization parameter μ of label propagation are tuned in range $\{0.01,0.1,1,10,100\}$.
- MVU [26] and MPME [41]. Both methods are learning the embeddings of all data points in the unsupervised setting. Due to the high complexity of SDP solver used in MVU, we try two variants of MVU called Landmark MVU and Fast MVU and choose the best results for comparison [53].
- StructSSL, which is the proposed method as shown in Algorithm 1. Two base classifiers, SVM and K-NN classier, are used, and three distance functions are tested. We fix $\lambda_1 = 1.0$, and tune $m \in \{5,10,20,50\}$ for the dimensionality of latent points and $K \in \{5,10,20\}$ for K-NN graph. The label balance parameter ν is chosen in the grid $\{0.2,0.4,0.8\}$ and the trade-off parameter for supervised information is $\lambda_2 \in \{0.01,0.1,1\}$. The prior parameter γ is tuned in $\{0.1,1\}$.

TABLE II
DATA SETS USED IN THE EXPERIMENTS

Data Set	n	c	d
three-moon	1500	3	100
Digit1	1500	2	241
Text	1500	2	11960
USPS	1500	2	241
COIL6	1500	6	241
Opt-Digits	5620	10	64

In addition to the method-specific parameters, some parameters shared by the above-mentioned methods are discussed as follows. In the graph-based methods, K in the neighborhood graph is tuned over $\{2,5,10,50,100,200,n-1\}$. The bandwidth parameter of the Gaussian kernel in SQ-Loss-1, SGT and MP is determined over $\{g_a/3 : a \in \{2,3,\ldots,10\}\}$, where g_a is the average distance between each sample and its ath nearest neighbor over the entire data set. In LapRLS, the bandwidth parameter is tuned in a slightly different set $\{2^a\sigma : a \in \{-3,-2,-1,0,1,2,3\}\}$ where σ is the average norm of the feature vectors as recommended in [1]. In StructSSL with (8), the bandwidth parameters tuned for LapRLS are used to calculate the Gaussian kernel. For methods without inherent classification model, SVM for classification [49] with Gaussian kernel is used for evaluation. In the following results, we mark the unavailable results from corresponding methods as '-'. And, results taken from [50] do not report the standard deviation of each setting.

For binary classification methods such as SGT, LapRLS and SimpleMKL, the one vs. rest strategy is used to obtain the results for multiclass data sets. The average accuracies over 10 runs with randomly selected number of labeled samples are reported, where $n_l \in \{10,20,50,80,100,150\}$. For fair comparison, we compare the results of our proposed method StructSSL with the best results reported in baseline methods on the same data sets when the experiments are conducted under the same setting.

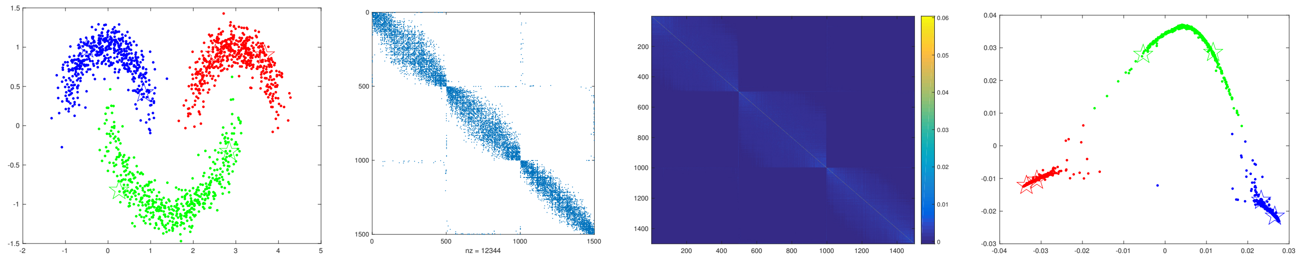
B. Synthetic data

The three-moon data is used to investigate the properties of the proposed method, which has also been used in the existing SSL methods [51], [52]. The three-moon data consists of three two-dimensional half circles with added Gaussian noise, i.e., $\mathcal{N}(0,0.14)$. The center locations of three points are $(0,0.5)$, $(3,0.5)$ and $(1.5,-0.5)$ with radius 1, 1 and 1.5, respectively. 500 points are uniformly sampled from each half circle. The data is expanded to the space $\mathbb{R}^{100}$ with only Gaussian noise for the rest 98 dimensions. The data is visualized in 2-D space using the two true features as shown in Fig. 1(a). The results are obtained by applying three different classifiers (1-NN, 3-NN and SVM) on the learned embeddings in 2-D space using the Euclidean distance (6) with prefixed parameters: $\lambda_2 = 10^{-3}$, $\nu = 0.9$, and $K = 5$.

The mean accuracies of StructSSL over 10 runs with randomly sampled labels are shown in Table III by comparing with other methods on the same data, where the results of these

TABLE III
MEAN ACCURACIES WITH STANDARD DEVIATIONS OF 8 COMPARED METHODS ON THREE-MOON DATA OVER 10 RUNS WITH RANDOMLY SELECTED NUMBER OF LABELS. THE BEST RESULTS ARE IN BOLD.

n_l	6 (0.4%)	15 (1%)	25 (1.67%)	30 (2%)	50 (3.3%)	75 (5%)
multicalss MBO	-	-	68.3	-	84.1	94.3
LapRF (m=1)	-	-	95.1	-	96.4	98.1
TVRF (m=1)	-	-	96.4	-	98.2	98.4
LapRF (m=2)	-	-	96.4	-	97.9	98.5
TVRF (m=2)	-	-	96.4	-	98.2	98.6
StructSSL (Euclidean, 1-NN)	99.26 ± 0.08	99.27 ± 0.06	99.14 ± 0.36	99.19 ± 0.23	99.19 ± 0.26	99.25 ± 0.18
StructSSL (Euclidean, 3-NN)	99.26 ± 0.03	99.25 ± 0.04	99.32 ± 0.07	99.26 ± 0.14	99.28 ± 0.07	99.31 ± 0.06
StructSSL (Euclidean, SVM)	**99.34** ± **0.06**	**99.33** ± **0.05**	**99.38** ± **0.08**	**99.37** ± **0.06**	**99.36** ± **0.09**	**99.39** ± **0.07**



(a) (b) (c) (d)

Fig. 1. The results obtained by StructSSL for three-moon data with 6 labels. (a) The ground truth data in 2-D space over the two true features where pentagram markers stands for the selected labeled data. (b) The adjacency matrix A learned by StructSSL (nz is the number of non-zero entries). (c) The kernel matrix Q^{-1} learned by StructSSL. (d) The visualization of latent points in 2-D space with the corresponding predicted labels.

TABLE IV
MEAN ACCURACIES WITH STANDARD DEVIATIONS OF 16 COMPARED METHODS ON DIGIT1 DATA OVER 10 RUNS WITH RANDOMLY SELECTED NUMBER OF LABELS. THE BEST RESULTS ARE IN BOLD.

n_l	10 (0.7%)	20 (1.3%)	50 (3.3%)	80 (5.3%)	100 (6.7%)	150 (10%)
k-NN	67.6	79.5	90.2	93.2	91.2	95.2
SGT	92.1	93.6	96.2	97.1	97.4	97.7
LapRLS	92.4	95.3	95.7	96.2	97.1	97.4
SQ-Loss-I	91.2	94.9	96.9	96.6	97.2	97.1
MP	92.1	95.4	96.1	97.4	97.4	97.8
AnchorGraph	93.97 ± 1.83	96.75 ± 1.03	97.11 ± 1.29	97.57 ± 0.60	98.02 ± 0.25	97.77 ± 0.59
f-FME	94.23 ± 2.00	96.71 ± 1.00	97.30 ± 0.55	97.61 ± 0.61	97.82 ± 0.31	97.70 ± 0.39
r-FME	94.22 ± 2.05	96.49 ± 0.89	97.07 ± 0.79	97.44 ± 0.53	97.49 ± 0.43	97.54 ± 0.43
KernelLP	93.08 ± 2.75	94.09 ± 1.86	95.74 ± 1.29	96.01 ± 1.17	95.56 ± 0.86	95.61 ± 0.66
SSLRR	83.02 ± 3.50	88.45 ± 2.47	93.04 ± 1.22	94.73 ± 1.64	95.49 ± 0.82	96.01 ± 0.66
SimpleMKL	78.28 ± 4.16	84.36 ± 4.32	91.41 ± 2.37	93.91 ± 1.90	94.67 ± 0.78	94.69 ± 1.10
SKL	93.76 ± 3.01	95.91 ± 1.50	97.32 ± 0.87	97.53 ± 0.67	97.78 ± 0.49	97.87 ± 0.35
MVU	91.78 ± 1.53	92.76 ± 1.33	93.83 ± 1.05	94.35 ± 0.94	94.35 ± 0.79	94.40 ± 0.58
MPME	90.65 ± 3.39	92.32 ± 3.60	96.45 ± 1.16	97.42 ± 1.27	97.83 ± 0.64	97.81 ± 0.64
StructSSL (Euclidean, SVM)	95.58 ± 2.48	96.32 ± 1.62	98.65 ± 0.39	**98.66** ± **0.50**	**98.91** ± **0.35**	98.81 ± 0.35
StructSSL (Gaussian, SVM)	**95.62** ± **2.09**	**96.84** ± **1.48**	**98.85** ± **0.40**	98.49 ± 0.54	98.86 ± 0.41	**98.90** ± **0.33**

methods are taken from [51]. From the results in Table III, we observe that 1) StructSSLs outperform compared methods over all the varied number of labels such as $n_l \in \{25, 50, 75\}$. 2) StructSSLs with the less labeled data, for example, $n_l = 6$, outperform baseline methods with more labeled data. 3) SVM as the classifier in StructSSL is marginally better than 1-NN and 3-NN. In the following experiments, we will report StructSSL based on the SVM classifier.

In addition, we show various intermediate results obtained by StructSSL including the sparse similarity matrix A (weighted graph), kernel matrix Q^{-1}, and latent embeddings $\widehat{Z}$ of the three-moon data with the predicted labels in Fig 1. First, we observe from Fig. 1(b) that the number of non-zero entries of the learned sparse similarity matrix is $12,344$, which is less than 0.6% of 1500×1500 matrix and also smaller than the 5-NN graph. Hence, StructSSL can reduce the initial non-zero entries of the K-NN graph to a sparser similarity matrix. The kernel matrix in Fig. 1(c) shows very clear blockwise diagonal structure, which corresponds to the fact that data points are sampled from three half-circles. Finally, from Fig. 1(d), it is easy to see that the embedded points in the 2-D space demonstrate three components where two smaller ones (blue and red) and a larger one (green). Each component corresponds to one half-circle of the ground truth data. Hence, StructSSL can recover the proper inherent smooth manifold structure of data regardless of the noise.

Since the classification results of six available labels can achieve much better results than compared methods over varied number of labels, this implies that StructSSL with distance preservation is effective and less sensitive to the number of labeled data due to the robustness of the embedding space over a smooth skeleton structure for SSL.

C. Benchmark data sets

We conduct extensive experiments by comparing StructSSL with various baseline methods for both binary classification and multiclass classification in terms of classification accuracy

TABLE V
MEAN ACCURACIES WITH STANDARD DEVIATIONS OF 15 COMPARED METHODS ON TEXT DATA OVER 10 RUNS WITH RANDOMLY SELECTED NUMBER OF LABELS. THE BEST RESULTS ARE IN BOLD.

n_l	10 (0.7%)	20 (1.3%)	50 (3.3%)	80 (5.3%)	100 (6.7%)	150 (10%)
k-NN	60.2	64.2	71.6	72.4	72.3	74.5
SGT	70.4	70.9	73.1	76.9	77.0	78.1
LapRLS	68.2	69.1	71.2	73.4	74.2	76.2
SQ-Loss-I	67.9	72.0	74.1	76.8	76.8	76.6
MP	70.3	72.6	73.0	75.9	75.4	77.9
AnchorGraph	62.11 ± 5.93	64.43 ± 1.39	68.55 ± 0.10	69.44 ± 1.79	73.61 ± 0.96	73.19 ± 0.73
f-FME	63.66 ± 0.52	69.05 ± 3.73	73.55 ± 1.12	74.08 ± 0.60	76.71 ± 0.40	78.52 ± 1.57
r-FME	69.40 ± 1.14	68.41 ± 3.30	72.21 ± 0.39	74.08 ± 1.79	75.21 ± 0.10	75.33 ± 0.73
KernelLP	65.91 ± 4.84	70.34 ± 3.43	74.94 ± 1.27	76.78 ± 0.94	78.08 ± 1.35	78.94 ± 1.09
SSLRR	61.66 ± 5.45	66.52 ± 3.08	72.17 ± 3.71	74.81 ± 1.36	76.69 ± 1.33	78.70 ± 1.15
SimpleMKL	62.40 ± 4.62	68.46 ± 3.26	73.86 ± 1.62	74.85 ± 1.02	76.86 ± 0.89	73.93 ± 2.81
SKL	62.38 ± 5.24	67.08 ± 3.92	71.78 ± 2.94	74.64 ± 1.83	75.29 ± 1.06	76.19 ± 1.24
MVU	62.88 ± 1.76	62.72 ± 2.75	63.83 ± 0.94	64.42 ± 0.52	64.46 ± 0.79	65.02 ± 0.38
MPME	67.30 ± 4.87	72.41 ± 2.10	74.78 ± 1.96	77.06 ± 1.00	76.84 ± 1.19	77.38 ± 0.85
StructSSL (Cosine, SVM)	**73.97 ± 4.27**	**76.69 ± 1.65**	**78.26 ± 0.71**	**78.78 ± 1.26**	**79.47 ± 0.97**	**79.55 ± 1.05**

TABLE VI
MEAN ACCURACIES WITH STANDARD DEVIATIONS OF 16 COMPARED METHODS ON USPS DATA OVER 10 RUNS WITH RANDOMLY SELECTED NUMBER OF LABELS. THE BEST RESULTS ARE IN BOLD.

n_l	10 (0.7%)	20 (1.3%)	50 (3.3%)	80 (5.3%)	100 (6.7%)	150 (10%)
k-NN	80.0	80.4	90.7	92.7	93.6	94.9
SGT	86.2	87.9	94.0	95.7	96.0	97.0
LapRLS	83.9	86.9	93.7	94.7	95.4	95.9
SQ-Loss-I	81.4	82.0	93.6	95.8	95.2	95.2
MP	88.1	90.4	93.9	95.0	96.2	96.8
AnchorGraph	72.90 ± 6.17	77.05 ± 6.60	82.25 ± 4.68	87.96 ± 3.62	86.46 ± 3.64	91.32 ± 2.17
f-FME	73.34 ± 6.63	80.70 ± 6.20	87.26 ± 3.63	90.52 ± 1.71	90.05 ± 3.48	92.38 ± 1.51
r-FME	72.89 ± 6.18	77.22 ± 4.86	82.24 ± 4.70	87.97 ± 3.60	86.44 ± 3.74	91.01 ± 3.07
KernelLP	69.68 ± 5.50	78.93 ± 6.28	86.15 ± 4.48	89.26 ± 3.08	91.39 ± 1.75	93.45 ± 1.48
SSLRR	79.66 ± 2.28	80.80 ± 0.91	80.88 ± 0.48	81.69 ± 0.00	82.14 ± 0.00	83.33 ± 0.00
SimpleMKL	62.48 ± 22.06	80.78 ± 5.74	81.26 ± 4.82	84.14 ± 6.32	84.53 ± 6.32	89.70 ± 2.36
SKL	66.83 ± 9.17	84.52 ± 8.56	90.94 ± 2.81	95.01 ± 1.13	92.84 ± 1.61	95.46 ± 0.69
MVU	78.08 ± 9.17	88.14 ± 2.78	90.33 ± 2.23	92.54 ± 1.29	91.21 ± 1.55	93.57 ± 0.73
MPME	86.68 ± 7.82	93.07 ± 2.62	95.66 ± 1.39	96.77 ± 0.86	96.81 ± 0.86	97.27 ± 0.36
StructSSL (Euclidean, SVM)	91.11 ± 3.37	94.22 ± 2.29	96.34 ± 0.41	96.83 ± 0.22	96.81 ± 0.33	97.03 ± 0.21
StructSSL (Gaussian, SVM)	**91.48 ± 2.95**	**95.32 ± 1.91**	**96.97 ± 0.62**	**97.41 ± 0.19**	**97.48 ± 0.24**	**97.59 ± 0.13**

over the unlabeled data and report the mean (± standard deviation) accuracies over 10 runs with randomly selected number of labels. Data descriptions for three binary classification and two multiclass classifcation, as well as the best experimental results of the compared methods with parameter tuning in the above-mentioned grids are shown as follows:

1) Digit1 data: This data consists of artificially generated writings (images) of the digit "1" developed by [54], which was designed to show that the low-dimensional manifold is not the cluster structure. Each image is the size of 16×16 and 1500 images are sampled. The class labels are assigned according to the tilt angle, with the boundary corresponding to an upright digit. A sequence of transformations are applied to the data for increasing the learning difficulty. Both Euclidean distance (6) and Gaussian kernel distance (8) are evaluated in StructSSL. The mean accuracies of 16 compared methods on Digit1 are reported in Table IV.

2) Text data: This is the 5 comp.* groups from the Newsgroups data set for classifying the IBM category versus the rest [55]. Each document is a sparse representation of a term frequency/inverse document frequency with $11,960$ dimensions. The cosine discrepancy (7) is used in StructSSL as the input distance since it generally works better in text classification. The mean accuracies of 15 compared methods on Text are reported in Table V.

3) USPS data: The benchmark USPS data is derived from the famous USPS set of handwritten digits. 150 images are randomly drawn for each of the ten digits. The digits "2" and "5" are assigned to the class $+1$ and the others are assigned to the class -1. The obscured data [1] is obtained in order to prevent researchers from exploiting the known spatial relationship of features in the image. Both Euclidean distance (6) and Gaussian kernel distance (8) are used in StructSSL. The mean accuracies of 16 compared methods on USPS are reported in Table VI.

4) COIL6 data: COIL6 [1] is created from the Columbia object image library (COIL-100), which is a set of color images of 100 different objects taken from different angles (in steps of 5 degrees) at a resolution of 128×128 [56]. The red channel of each image is downsampled to 16×16 by averaging over blocks of 8×8 pixels. 24 of the 100 objects are randomly selected and then are partitioned to six classes of four objects each. Both Euclidean distance (6) and Gaussian kernel distance (8) are used in StructSSL. The mean accuracies of 19 compared methods are reported in Table VII.

5) Opt-Digits data: This data consists of normalized bitmaps of handwritten digits from a preprinted form. 32×32 bitmaps are divided into nonoverlapping blocks of 4×4 and the number of on pixels are counted in each block. This generates an input matrix of 8×8 where each element is an integer in the range $[0, 16]$. Each bitmap has one label of 10 classes, i.e., $\{0, 1, \ldots, 9\}$. Both Euclidean distance (6) and Gaussian kernel distance (8) are used in StructSSL. The mean accuracies of 17 compared methods are reported in Table VIII.

TABLE VII
MEAN ACCURACIES WITH STANDARD DEVIATIONS OF 19 COMPARED METHODS ON COIL6 DATA OVER 10 RUNS WITH RANDOMLY SELECTED NUMBER OF LABELS. WE MARK THE UNAVAILABLE RESULTS FROM CORRESPONDING METHODS AS '-'. THE BEST RESULTS ARE IN BOLD.

n_l	10 (0.7%)	20 (1.3%)	50 (3.3%)	80 (5.3%)	100 (6.7%)	150 (10%)
k-NN	34.5	53.9	66.9	77.9	79.2	83.5
SGT	40.1	61.2	78.0	88.5	89.0	89.9
LapRLS	49.2	61.4	78.4	80.1	84.5	87.8
SQ-Loss-1	48.9	63.0	81.0	87.35	89.0	90.9
MP	47.7	65.7	78.5	**89.6**	90.2	91.1
LapRF (m=1)	-	-	71.7	-	87.0	91.0
TVRF (m=1)	-	-	80.3	-	90.0	91.7
multiclass GL	-	-	-	-	-	91.2
multiclass MBO	-	-	-	-	-	91.46
AnchorGraph	48.22 ± 4.39	58.33 ± 5.59	80.43 ± 5.33	87.67 ± 1.91	91.09 ± 2.12	91.53 ± 1.24
f-FME	47.07 ± 3.11	57.03 ± 5.86	81.52 ± 6.38	89.42 ± 2.68	91.48 ± 1.41	92.87 ± 0.66
r-FME	47.64 ± 4.11	57.19 ± 3.84	81.59 ± 6.27	89.40 ± 2.67	91.60 ± 1.51	92.87 ± 0.64
KernelLP	41.96 ± 2.86	47.36 ± 4.08	69.13 ± 4.28	78.51 ± 2.23	81.79 ± 1.85	86.10 ± 1.63
SSLRR	35.79 ± 4.69	38.08 ± 3.34	43.09 ± 4.30	48.93 ± 4.30	51.08 ± 2.60	53.41 ± 3.72
SimpleMKL	21.16 ± 2.54	21.13 ± 2.15	27.67 ± 3.67	31.05 ± 4.97	35.59 ± 4.86	41.04 ± 3.56
SKL	44.60 ± 5.73	54.89 ± 5.16	78.88 ± 6.95	86.48 ± 2.63	86.83 ± 1.82	89.85 ± 1.46
MPME	48.86 ± 6.73	56.00 ± 5.22	74.74 ± 2.71	80.39 ± 1.15	80.86 ± 1.39	82.41 ± 1.83
StructSSL (Euclidean, SVM)	52.76 ± 5.19	65.89 ± 4.78	82.27 ± 3.95	88.81 ± 1.58	90.60 ± 1.68	91.72 ± 0.55
StructSSL (Gaussian, SVM)	**54.91 ± 4.31**	**67.36 ± 4.90**	**85.09 ± 4.76**	89.54 ± 1.87	**91.88 ± 1.06**	**93.24 ± 0.84**

TABLE VIII
MEAN ACCURACIES WITH STANDARD DEVIATIONS OF 17 COMPARED METHODS ON OPT-DIGITS DATA OVER 10 RUNS WITH RANDOMLY SELECTED NUMBER OF LABELS. WE MARK THE UNAVAILABLE RESULTS FROM CORRESPONDING METHODS AS '-'. THE BEST RESULTS ARE IN BOLD.

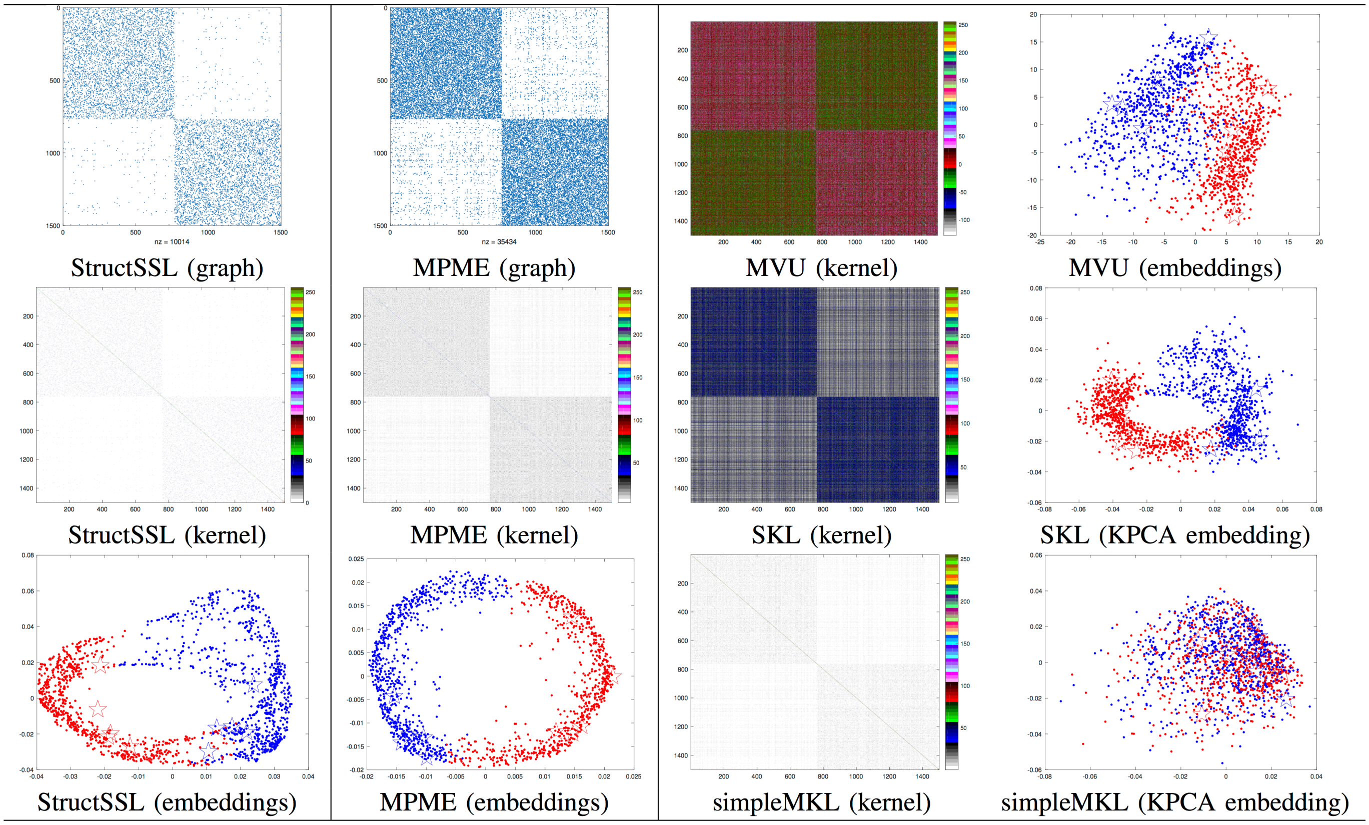
n_l	10 (0.18%)	20 (0.36%)	50 (0.89%)	80 (1.42%)	100 (1.78%)	150 (2.67%)
k-NN	79.6	83.9	85.5	90.5	92.0	93.8
SGT	90.4	90.6	91.4	94.7	97.4	97.4
LapRLS	89.7	91.2	92.3	96.1	97.6	97.3
SQ-Loss-1	92.2	90.2	95.9	97.2	97.3	97.7
MP	90.6	90.8	94.7	96.6	97.0	97.1
LapRF (m=1)	-	-	79.0	-	95.2	96.8
TVRF (m=1)	-	-	95.9	-	97.2	98.3
AnchorGraph	92.29 ± 3.02	93.29 ± 4.87	97.52 ± 0.59	97.88 ± 0.60	97.90 ± 0.55	98.01 ± 0.38
f-FME	92.22 ± 3.00	93.39 ± 4.76	97.31 ± 0.56	97.86 ± 0.60	97.90 ± 0.55	98.05 ± 0.61
r-FME	92.30 ± 3.02	93.31 ± 4.86	97.54 ± 0.60	97.91 ± 0.60	97.90 ± 0.55	98.03 ± 0.38
KernelLP	69.17 ± 8.81	72.09 ± 2.00	86.99 ± 0.67	89.17 ± 0.20	91.68 ± 1.83	93.78 ± 0.28
SSLRR	69.67 ± 9.01	69.81 ± 3.30	83.68 ± 0.48	86.71 ± 0.71	86.86 ± 3.27	89.25 ± 0.16
SimpleMKL	42.27 ± 23.00	75.56 ± 3.29	87.01 ± 2.32	90.63 ± 1.27	92.12 ± 0.82	93.24 ± 0.93
SKL	88.02 ± 3.89	91.34 ± 4.47	96.79 ± 0.93	97.21 ± 1.02	97.66 ± 0.80	97.98 ± 0.75
MPME	91.81 ± 4.54	95.36 ± 1.62	97.68 ± 0.38	97.92 ± 0.20	97.97 ± 0.26	98.18 ± 0.20
StructSSL (Euclidean, SVM)	92.39 ± 4.03	95.81 ± 1.81	97.71 ± 0.45	98.10 ± 0.15	98.20 ± 0.20	98.33 ± 0.15
StructSSL (Gaussian, SVM)	**96.30 ± 2.64**	**97.61 ± 1.50**	**98.35 ± 0.29**	**98.53 ± 0.09**	**98.59 ± 0.14**	**98.67 ± 0.10**

D. Discussions on experimental results

Experimental results in Tables IV-VIII show that the proposed method StructSSL in terms of three different distance measurements are very competitive to baseline methods including the state-of-the-art SSL methods, two unsupervised dimensionality reduction methods, and two kernel learning approaches on five datasets with three binary-class and two multiclass problems. By looking into the details, we have the following observations:

- StructSSLs show promising results compared with other SSL methods. The significant improvements can be observed with small amounts of labels over all the datasets used in the experiments. Moreover, StructSSL is flexible to take different distance functions as the input. This is crucially important for the success of StructSSL applied to Text data, where cosine similarity is known to be a good measurement. Distance metric derived from Gaussian kernel generally works better than the Euclidean distance. Although some methods are specifically designed for multiclass SSL such as multiclass MBO and TVRF, StructSSL can achieve competitive or even better results.
- Although the distance preservation criterion is used in MVU, MPME and StructSSL, our StructSSL demonstrates the better performance than MVU and MPME on SSL due to the integration of labels into the learning model.
- In regards to kernel learning, StructSSL shows significant better results than SimpleMKL which does not take the advantage of labels. SKL takes labels into learning a better kernel for unlabeled data. However, it is less robust when the number of available labels is small. In contrast, StructSSL shows good results in all levels of available labels. The key difference of our StructSSL from SimpleMKL and SKL is the learning of a sparse weighted graph, which is not able to be obtained by the compared kernel learning methods.

We further demonstrate the effectiveness of StructSSL in terms of sparse graph matrix, kernel matrix, and the embeddings with ground truth labels by comparing with baseline methods. For methods without naturally embeddings as the output such as SKL and simpleMKL, we take KPCA as the embedding approach. In addition, we also show the sparse weighted graphs learned by MPME and our method. Figs. 2 and 3 show these intermediate results on Digit1 and COIL6,



StructSSL (graph)

MPME (graph)

MVU (kernel)

MVU (embeddings)

StructSSL (kernel)

MPME (kernel)

SKL (kernel)

SKL (KPCA embedding)

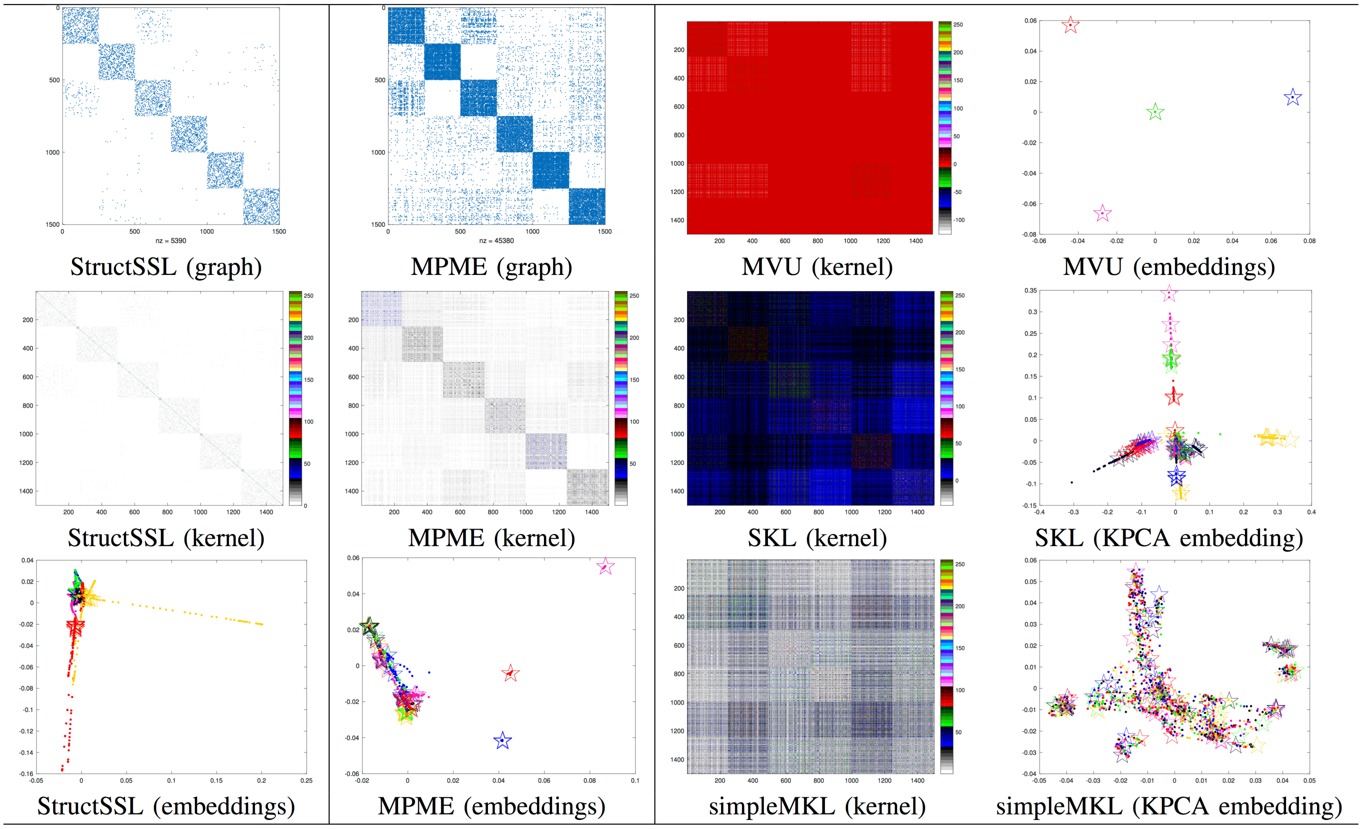
StructSSL (embeddings)

MPME (embeddings)

simpleMKL (kernel)

simpleMKL (KPCA embedding)

Fig. 2. Intermediate results obtained by various baseline methods on Digit1 data including kernels, weighted graph matrices, and embeddings. For embeddings, the pentagram markers stand for the labeled data, and each color represents one class.



StructSSL (graph)

MPME (graph)

MVU (kernel)

MVU (embeddings)

StructSSL (kernel)

MPME (kernel)

SKL (kernel)

SKL (KPCA embedding)

StructSSL (embeddings)

MPME (embeddings)

simpleMKL (kernel)

simpleMKL (KPCA embedding)

Fig. 3. Intermediate results obtained by various baseline methods on COIL6 data including kernels, weighted graph matrices, and embeddings. For embeddings, the pentagram markers stand for the labeled data, and each color represents one class.

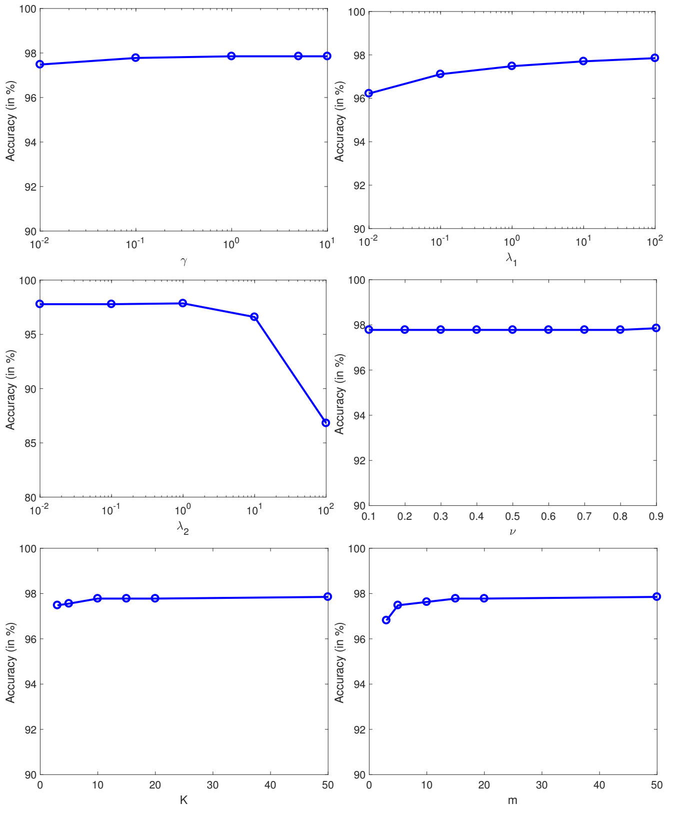


Fig. 4. Parameter sensitivity analysis of the proposed StructSSL (Euclidean SVM) method using USPS with 150 labels.

respectively. We have the following observations:

- The weighted graph learned by StructSSL is much sparser than that learned by MPME. This is because the labeled data as the guide contributes to the learned density function, and this guidance promotes the sparsity so that more inactive distance preservation constraints are obtained. As observed in Figs. 2 and 3, embeddings of StructSSL show better class separation than MPME.
- Both MVU and simpleMKL learn kernel matrices in an unsupervised setting. Their class separation patterns are not as good as StructSSL. This can be confirmed easily on COIL6. Although SKL takes both the manifold of the data and labels into account, the learned kernel does not properly capture the manifold structure, which is affected significantly by the initial weighted graph.

The above observations imply that both the distance preservation criterion and label information contribute to the success of our proposed method. It is worth noting that the weighted graph, kernel matrix and the embeddings learned by the proposed method can be useful for other purposes such as data visualization, which is not restricted to semi-supervised classification problems as studied in this paper.

E. Parameter sensitivity analysis

We conduct the sensitivity analysis of six parameters used in Algorithm 1. Parameters γ and λ_1 have impacts on the embeddings and the learned graph, while γ_2 and ν balance the influence from labels and the whole input data. Moreover, the neighborhood size K and the dimensionality of latent embeddings m improves the sparsity of the initial graph for learning and the performance of classification by using dimensionality reduction to remove data noise. As there are six parameters, we cannot visualize results of all varied parameters in one plot. We take the commonly used strategy by varying one parameter in a given range and fixing the others by reporting the best results over all the other parameters.

Fig. 4 shows the sensitivity analysis of one parameter by fixing the others. We have the following observations:

- Our proposed method is robust in terms of γ and λ_1 for constructing graph based on distance preservation criterion.
- The proposed method is a bit sensitive to the supervised information. We observe that the classification performance decreases when λ_2 becomes large on USPS. In other words, the balance between graph structure learning and the importance of labels is data-dependent.
- For both K and m, they demonstrate the better accuracies at the beginning and tune to be stable later if both parameter values increases. In general, the accuracies vary in a small interval.

From the above observations, StructSSL is a bit sensitive to λ_2, but robust to other parameters. More importantly, the embeddings and the learned graph are not too sensitive to their controlled parameters.

V. Conclusion

In this paper, we propose a probabilistic semi-supervised learning framework based on the assumption of distance preservation criterion, which has been successfully explored in unsupervised dimensionality reduction methods, and class separability criterion on labeled data. Moreover, our proposed method can naturally integrate different priors from either probability perspective or prior knowledge in the form of constraints. In addition to classification problems, our method can also provide the learned sparse weighted graph with the optimized similarities between data points, and also the embeddings for data visualization. Experiments on synthetic and benchmark datasets show promising results by comparing with the best results of a variety of existing methods, with the significant improvement on small amounts of labeled data.

References

[1] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[2] X. J. Zhu, "Semi-supervised learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2005.

[3] G. Druck and A. McCallum, "High-performance semi-supervised learning using discriminatively constrained generative models," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 319–326.

[4] T. Joachims, "Transductive inference for text classification using support vector machines," in *Icml*, vol. 99, 1999, pp. 200–209.

[5] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive svms," *Journal of Machine Learning Research*, vol. 7, no. Aug, pp. 1687–1712, 2006.

[6] Z. Xiaojin and G. Zoubin, "Learning from labeled and unlabeled data with label propagation," *Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University*, 2002.

[7] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 290–297.

[8] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 639–655.

[9] Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," *Proceedings of the 33rd International COnference on Machine Learning*, 2016.

[10] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of machine learning research*, vol. 7, no. Nov, pp. 2399–2434, 2006.

[11] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *Advances in neural information processing systems*, 2011, pp. 55–63.

[12] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is nearest neighbor meaningful?" in *International conference on database theory*. Springer, 1999, pp. 217–235.

[13] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 679–686.

[14] M. Wang, W. Fu, S. Hao, D. Tao, and X. Wu, "Scalable semi-supervised learning by efficient anchor graph regularization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1864–1877, 2016.

[15] M. Wang, W. Fu, S. Hao, H. Liu, and X. Wu, "Learning on big graph: Label inference and regularization with anchor hierarchy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 5, pp. 1101–1114, 2017.

[16] S. Qiu, F. Nie, X. Xu, C. Qing, and D. Xu, "Accelerating flexible manifold embedding for scalable semi-supervised learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, 2019.

[17] Y. Zhang, Z. Zhang, J. Qin, L. Zhang, B. Li, and F. Li, "Semi-supervised local multi-manifold isomap by linear embedding for feature extraction," *Pattern Recognition*, vol. 76, pp. 662–678, 2018.

[18] B. Yang, M. Xiang, and Y. Zhang, "Multi-manifold discriminant isomap for visualization and classification," *Pattern Recognition*, vol. 55, pp. 215–230, 2016.

[19] Z. Zhang, Y. Zhang, G. Liu, J. Tang, S. Yan, and M. Wang, "Joint label prediction based semi-supervised adaptive concept factorization for robust data representation," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[20] O. Zoidi, A. Tefas, N. Nikolaidis, and I. Pitas, "Positive and negative label propagations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 2, pp. 342–355, 2016.

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

[22] J. Chen and Y. Liu, "Locally linear embedding: a survey," *Artificial Intelligence Review*, vol. 36, no. 1, pp. 29–48, 2011.

[23] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with l1-graph for image analysis," *IEEE transactions on image processing*, vol. 19, no. 4, pp. 858–866, 2009.

[24] B. Lake and J. Tenenbaum, "Discovering structure by learning sparse graphs," 2010.

[25] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[26] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 106.

[27] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in neural information processing systems*, 2004, pp. 321–328.

[28] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.

[29] D. Wijaya, P. P. Talukdar, and T. Mitchell, "Pidgin: ontology alignment using web text as interlingua," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013, pp. 589–598.

[30] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao, "Graph regularized transductive classification on heterogeneous information networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 570–586.

[31] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2491–2521, 2008.

[32] Q. Mao and I. W. Tsang, "Parameter-free spectral kernel learning," in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010.

[33] Z. Zhang, M. Zhao, and T. W. Chow, "Marginal semi-supervised submanifold projections with informative constraints for dimensionality reduction and recognition," *Neural Networks*, vol. 36, pp. 97–111, 2012.

[34] D. Zhang, Z.-H. Zhou, and S. Chen, "Semi-supervised dimensionality reduction," in *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 2007, pp. 629–634.

[35] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, 2007.

[36] M. Wang, X. Liu, and X. Wu, "Visual classification by ℓ_1-hypergraph modeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2564–2574, 2015.

[37] Z. Zhang, L. Jia, M. Zhao, G. Liu, M. Wang, and S. Yan, "Kernel-induced label propagation by mapping for semi-supervised classification," *IEEE Transactions on Big Data*, vol. 5, no. 2, pp. 148–165, 2019.

[38] L. Zhuang, Z. Zhou, S. Gao, J. Yin, Z. Lin, and Y. Ma, "Label information guided graph construction for semi-supervised learning," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4182–4192, 2017.

[39] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for web image search," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4649–4661, 2012.

[40] B. Liu, M. Wang, R. Hong, Z. Zha, and X.-S. Hua, "Joint learning of labels and distance metric," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 973–978, 2009.

[41] L. Wang, Q. Mao, and I. W. Tsang, "Latent smooth skeleton embedding," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[42] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[43] Q. Mao, L. Wang, and I. W. Tsang, "A unified probabilistic framework for robust manifold learning and embedding," *Machine Learning*, vol. 106, no. 5, pp. 627–650, 2017.

[44] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[45] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[46] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[47] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[48] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International conference on artificial neural networks*. Springer, 1997, pp. 583–588.

[49] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[50] A. Subramanya and J. Bilmes, "Semi-supervised learning with measure propagation," *Journal of Machine Learning Research*, vol. 12, no. Nov, pp. 3311–3370, 2011.

[51] K. Yin and X.-C. Tai, "An effective region force for some variational models for learning and clustering," *Journal of Scientific Computing*, vol. 74, no. 1, pp. 175–196, 2018.

[52] C. Garcia-Cardona, E. Merkurjev, A. L. Bertozzi, A. Flenner, and A. G. Percus, "Multiclass data segmentation using diffuse interface methods on graphs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 8, pp. 1600–1613, 2014.

[53] K. Q. Weinberger, B. Packer, and L. K. Saul, "Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization." in *AISTATS*, vol. 2, no. 5. Citeseer, 2005, p. 6.

[54] M. Hein and J.-Y. Audibert, "Intrinsic dimensionality estimation of submanifolds in r d," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 289–296.

[55] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001.

[56] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (coil-20)," 1996.