Semi-Supervised Representation Learning based on Probabilistic Labeling

Ershad Banijamali[1] Ali Ghodsi[2]
[1]School of Computer Science, University of Waterloo
[2]Department of Statistics and Actuarial Science, University of Waterloo
{sbanijam , aghodsib}@uwaterloo.ca

***Abstract*—In this paper, we present a new algorithm for semi-supervised representation learning. In this algorithm, we first find a vector representation for the labels of the data points based on their local positions in the space. Then, we map the data to lower-dimensional space using a linear transformation such that the dependency between the transformed data and the assigned labels is maximized. In fact, we try to find a mapping that is as discriminative as possible. The approach will use Hilber-Schmidt Independence Criterion (HSIC) as the dependence measure. We also present a kernelized version of the algorithm, which allows non-linear transformations and provides more flexibility in finding the appropriate mapping. Use of unlabeled data for learning new representation is not always beneficial and there is no algorithm that can deterministically guarantee the improvement of the performance by exploiting unlabeled data. Therefore, we also propose a bound on the performance of the algorithm, which can be used to determine the effectiveness of using the unlabeled data in the algorithm. We demonstrate the ability of the algorithm in finding the transformation using both toy examples and real-world datasets.**

I. Introduction

As the amount of data grows rapidly, the process of extracting meaningful information becomes more and more challenging. Among these challenges, having no access to the data categories is very crucial. In the real world, the amount of labeled data compared to unlabeled data is almost negligible. On the other hand, determining data categories, or acquiring labels, is expensive for many reasons, e.g. it is extremely time-consuming for large datasets and usually needs human supervision. The importance of the methods that can benefit from this fast growing amount of unlabeled data has significantly increased.

Semi-supervised learning is the area of utilizing unlabeled data combined with, usually very smaller set of, labeled data to gain better data representation or classification accuracy. There are wide range of applications for semi-supervised learning including, but not limited to, text classification [1], [14], genetics and medical research [4], [21], and object detection [16].

A. Related Works

In recent years, semi-supervised learning has attracted attention from many researchers and several algorithms have been designed for semi-supervised learning that can relate to the present work.

Graph-based algorithms, which usually define a loss function for labeled data and use unlabeled as a regularizer, are important classes of semi-supervised learning methods. Example of this class are [5], [26] that try to convey the labels to the unlabeled data over the edges of the graph. Label propagation has been tried in many other articles including [22] which, inspired by the idea of locally linear embedding (LLE) [17], assumes the labels of data points can be linearly constructed by the labels of their adjacent samples in an sparse neighborhood and [25], which tries to propagate the labels over pairs of data points. Transductive support vector machines (TSVM) is another class of algorithms, used by [7], in which the goal is to maximize the margin for both unlabeled and labeled points.

However, an important point in semi-supervised learning is that there exists no guarantee that the use of unlabeled data will help us to achieve a better representation of the data. In an excellent work by Cozman et. al [8], the important question "Do Unlabeled Data Improve or Degrade Classification Performance?" was addressed and it was shown that, not only unlabeled data can be useless in learning a new representation, but also it can degrade the performance of the algorithm in many cases. To reduce the likelihood of having destructive unlabeled samples, there is a set of assumptions about the structure of the underlying distribution of data, including smoothness assumption, clustering assumption, and manifold assumption, that researchers usually make one of them.

B. Contribution

Most of the semi-supervised algorithms include two objective functions for labeled and unlabeled data points, which are optimized jointly. In this paper, we also start with deriving two separate objective functions. For the labeled points, we look for a mapping which maximizes the dependency of the transformed points and their labels, and for the unlabeled points we look for a mapping that keeps them near their labeled neighbors. However, by some manipulations, we then combine these two functions and solve the problem by optimizing a single objective function. Further investigations show that the objective function can also be obtained by a specific assignment of labels to the points. We call this probabilistic labeling. This labeling not only provides the objective function

of our problem much faster and easier, but also enables us to obtain a bound on the performance of the algorithm based on probability of classification error in the original space. This bound shows the maximum deviation of the objective function value from its optimal value, when we know the true label of all data points in our dataset.

We will also derive the kernelized version of the algorithm, which is very helpful when the linear transformation does not provide a good representation of data in the target space. The results of applying the algorithm on real and synthetic datasets will be presented.

II. Background: Hilber-Schmidt Independence Criterion (HSIC)

The Hilbert-Schmidt Independence Criterion (HSIC) is a very useful tool in statistics to measure the dependence between two random variables [11]. We use HSIC in our proposed method. Following is a short description about this measure.

Definition 1. *Suppose $\mathcal{X}$ and $\mathcal{Y}$ are two domain sets. Let ϕ and ψ be two mappings that map $\mathcal{X}$ and $\mathcal{Y}$ to their corresponding Reproducing Kernel Hilbert Space (RKHS) $\mathcal{F}$ and $\mathcal{G}$. The Borel probability measure over $\mathcal{X} \times \mathcal{Y}$ is denoted by p_{xy}. Then HSIC is defined as the following:*

$$HSIC(p_{xy}, \mathcal{F}, \mathcal{G}) = \| \boldsymbol{E}_{x,y}[\phi(x) \otimes \psi(y)] - \mu_x \otimes \mu_y \|^2_{HS} \quad (1)$$

where μ_x and μ_y are mean of $\phi(x)$ and $\psi(y)$, respectively, and $\otimes$ is the tensor product. $\| \, . \, \|_{HS}$ is also the Hilbert-Schmidt norm.

The following theorem by [11] shows the relation between HSIC and independence of x and y, when (x, y) is drawn from p_{xy}.

Theorem 1. *Suppose k and l are reproducing kernels of RKHS's $\mathcal{F}$ and $\mathcal{G}$ on the compact domains $\mathcal{X}$ and $\mathcal{Y}$. Assume, without loss of generality, $\| f \|_\infty \leq 1$ and $\| g \|_\infty \leq 1$ for all $f \in \mathcal{F}$ and $g \in \mathcal{G}$. Then, HSIC$(p_{xy}, \mathcal{F}, \mathcal{G})$ is zero, if and only if, x and y are independent.*

1) Empirical HSIC: The empirical HSIC was also defined in [11] to show that HSIC is, in fact, a practical criterion.

Definition 2. *Let $Z = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\} \subseteq \mathcal{X} \times \mathcal{Y}$ be a series of m independent observation drawn from p_{xy}. An estimation of HSIC is given by:*

$$HSIC(Z, \mathcal{F}, \mathcal{G}) = \frac{1}{(m-1)^2} \boldsymbol{tr}(K H_m L H_m) \quad (2)$$

where K and L are matrices containing the evaluation of the reproducing kernel of $\mathcal{F}$ and $\mathcal{G}$ respectively, and H_m is the centering matrix of size m, $H_m = I - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$.

III. Algorithm SSRL-PL

Let $\mathcal{X}$ be a unit ball in d-dimensional space and X contain n observations from $\mathcal{X}$ in form of a $d \times n$ matrix, i.e. $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$ where each $\mathbf{x}_i \in \mathbb{R}^d$ is a column vector. According to this definition, $\| \, \mathbf{x}_i \, \|_2 \leq 1 \;\; \forall i = \{1, .., n\}$, where $\| \, . \, \|_2$ is the *L*-2 norm of the vector.

Suppose from n samples, l of them have labels and the rest $u = n - l$ are unlabeled. Let $d \times l$ matrix X_L and $d \times u$ matrix X_U contain the set of labeled and unlabeld samples, respectively. Without loss of generality, assume X is ordered such that the first l samples are labeled, i.e. $X = [X_L, X_U]$.

Suppose there are also C classes of data points $\{1, 2, ..., C\}$. Variable y_i denotes the label of data point $\mathbf{x}_i$ in X_L, which indicate the class to whixh $\mathbf{x}_i$ belongs. For data points $\mathbf{x}_j$ in X_U, y_j is unknown.

The goal of our algorithm is to map the data to a p-dimensional space by finding a linear transformation, that we denote it by V. V is a $d \times p$ matrix while d can be much larger than p. Let $\mathbf{z}_i$ be the low-dimensional representation of data point $\mathbf{x}_i$. Then: $\mathbf{z}_i = V^\top \mathbf{x}_i$, where $V^\top$ is the transposed of V. Accordingly, we can obtain the matrix Z, which contains the low-dimensional data points as follows:

$$Z = V^\top X \quad (3)$$

Considering the dataset as a graph with n nodes, the transformation V transforms labeled and unlabeled points as follows.

Labeled Data: For the labeled data, we try to find a mapping that maximizes the dependency between low-dimensional data points and the labels, based on the HSIC measure. Therefore, we will have the following objective:

$$\begin{aligned} &\arg\max_V \frac{1}{(l-1)^2} \mathbf{tr}(Z_L^\top Z_L H_l K_l H_l) \\ &= \arg\max_V \frac{1}{(l-1)^2} \mathbf{tr}(X_L^\top V V^\top X_L H_l K_l H_l) \end{aligned} \quad (4)$$

where we used linear kernel for the data points in p-dimensional space and K_l is a kernel over labels. A kernel commonly used for labels is the delta kernel. Entry (i, j) of a delta kernel is 1 if $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same label and is 0 otherwise. Delta kernel can be interpreted as the adjacency matrix of a graph whose nodes are labeled data points. Regardless of the relative position of the nodes in this graph, there is an edge between two nodes if they have the same label. All points have also a self-loop. We will use this kernel for labels throughout this paper.

If we do not impose any constraint on V, the function can be unbound. A good choice for the constraint which also guarantees the orthonormality of the basis of the p-dimensional space is $V^\top V = I$, where I is the identity matrix. By adding this constraint and also re-arranging the matrices inside the trace, we will have:

$$\begin{aligned} \arg\max_V \quad & \frac{1}{(l-1)^2} \mathbf{tr}(V^\top X_L H_l K_l H_l X_L^\top V) \\ \text{subject to} \quad & V^\top V = I \end{aligned} \quad (5)$$

For the sake of simplicity, we do not write the $V^\top V = I$ in the next expressions. However, we always consider this constraint in defining objective functions.

The objective function in (5) can be recast using X (all data points, labeled and unlabeled) and a kernel K_n defined over X. K_n is an $n \times n$ matrix with all zero entries except the first $l \times l$ block, which is equal to K_l. Then, we will have:

$$\arg\max_V \frac{1}{(n-1)^2} \mathbf{tr}(V^\top X H_n K_n H_n X^\top V) \tag{6}$$

The solution to (6) is the eigenvectors corresponding to the p largest eigenvalues of matrix $XH_nK_nH_nX^\top$.

Unlabeled Data: In the previous part we defined an objective function based on the relation of labeled points with each other. Now we want to define another objective function based the relation of the unlabeled points with the rest of the of the points. The goal here is to find a transformation that preserves the neighborhood between unlabeled data points and their labeled neighbors. We want the unlabeled points to have high similarity with their labeled neighbors in the p-dimensional space. This is a rational choice, as a common assumption in semi-supervised learning is that close points in original space are likely to have same labels.

If unlabeled data point $\mathbf{x}_i$ and labeled data point $\mathbf{x}_j$ are neighbors in d-dimensional space, then $\mathbf{z}_i$ and $\mathbf{z}_j$ should have high similarity. Similarity can be defined in different ways but we measure the similarity between two points $\mathbf{z}_i$ and $\mathbf{z}_j$ as $< \bar{\mathbf{z}}_i, \bar{\mathbf{z}}_j >$, where $< ., . >$ is the dot product and $\bar{\mathbf{z}}_i$ is the centered version of $\mathbf{z}_i$. Hence, we can define the following objective function, which measures the similarity of neighboring points:

$$\max \sum_{ij} w_{ij} < \bar{\mathbf{z}}_i, \bar{\mathbf{z}}_j >= \max \sum_{ij} w_{ij} \bar{\mathbf{z}}_i^\top \bar{\mathbf{z}}_j \tag{7}$$

where $0 \leq w_{ij} \leq 1$ determines the strength of neighborhood between $\mathbf{x}_i$ and $\mathbf{x}_j$. Note that if both of these points are labeled then $w_{ij} = 0$, as we have already taken care of labeled points in K_n. Clearly maximizing this objective function forces points with strong neighborhood (large w_{ij}) to have large similarity.

The value of w_{ij} between two unlabeled points depend on their similarity in term of their neighborhood. That is, their connections to the labeled nodes. For example, if two unlabeled points have strong neighborhood with labeled points from similar classs, then w_{ij} is high.

We define an $n \times n$ matrix W that contains w_{ij}'s. Based on our definitions here, the first $l \times l$ block of this matrix is all zeros. The objective function in (7) can be written in the following matrix form:

$$\begin{aligned} \sum_{ij} w_{ij} \bar{\mathbf{z}}_i^\top \bar{\mathbf{z}}_j &= \mathbf{tr}(\bar{Z}^\top \bar{Z} W) = \mathbf{tr}(H_n Z^\top Z H_n W) \\ &= \mathbf{tr}(V^\top X H_n W H_n X^\top V) \end{aligned} \tag{8}$$

Therefore, we can also write this objective function similar to (4) by multiplying the trace function to the normalization factor $1/(n-1)^2$ and adding a constraint on V.

$$\arg\max_V \frac{1}{(n-1)^2} \mathbf{tr}(V^\top X H_n W H_n X^\top V) \tag{9}$$

Combining (6) and (9), we should find mapping V such that the following objective is maximized.

$$\arg\max_V \frac{1}{(n-1)^2} \mathbf{tr}(V^\top X H_n (K_n + W) H_n X^\top V) \tag{10}$$

The inner matrix, $K_n + W$, is the matrix we needed. The elements in K_n as stated above, define the edge of the corresponding graph globally, i.e. the local position of the labeled nodes (being each other's neighbors) does not have effect on the edge between them. To be consistent with this global structure of the graph, we extend the edges defined in W, i.e. if unlabeled node $\mathbf{x}_i$ is connected to the labeled node $\mathbf{x}_j$ with an edge of weight w_{ij}, we also connect $\mathbf{x}_i$ to all other labeled nodes from the same class as $\mathbf{x}_j$ by edges of weight w_{ij}. This will clearly affect the weight between unlabeled points too. Therefore, the structure of the graph remains global, but connection between unlabeled data points and the rest of the graph is determined based on their local positions.

Elements of $K_n + W$ show our certainty in connecting different nodes in the graph. For labeled nodes, we have 0 and 1 which indicates absolute certainty. For unlabeled nodes, we have $0 \leq w_{ij} \leq 1$, which is an indicator of our uncertainty. To capture these properties, we define a C-dimensional label vector for each data point. For the data point $\mathbf{x}_i$, the label vector is denoted by $\mathbf{y}_i$. If $\mathbf{x}_i$, is labeled then $\mathbf{y}_i$ is an all zero vector except in position y_i, which gets value 1 and it determines the class of $\mathbf{x}_i$. If $\mathbf{x}_i$ is unlabeled, then the c^{th} element of $\mathbf{y}_i$, which we denote it by y_i^c, is the probability that $\mathbf{x}_i$ belongs to class c, and $\sum_{c=1}^{C} y_i^c = 1$. To assign this label probabilities, we look at the set of the k nearest labeled neighbors of the unlabeled points $\mathbf{x}_i$. Let us denote this set by $\mathcal{L}_{i,k}$. Then:

$$y_i^c = \frac{f_i^c}{\sum_{c=1}^{C} f_i^c} \qquad \text{where} \qquad f_i^c = \sum_{\substack{\mathbf{x}_j \in \mathcal{L}_{i,k} \\ y_j^c = 1}} \mathcal{S}(\mathbf{x}_i, \mathbf{x}_j) \tag{11}$$

where $\mathcal{S}(.,.)$ is a measure of similarity. As nearby unlabeled points are sharing similar labeled points, they are more likely to have similar label probability vectors as well.

Now lets look at the dot product of label probability vectors of two points $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e. $\mathbf{y}_i \mathbf{y}_j^\top$ ($\mathbf{y}_i$'s are defined as row vectors). If $\mathbf{x}_i$ and $\mathbf{x}_i$ are labeled, this dot product builds elements of Delta kernel matrix, and if one of the points is unlabeled, the dot product builds elements of W. Therefore, we can build $K_n + W$ simply by $YY^\top$ where Y is an $n \times C$ label matrix. The i^{th} row of Y is $\mathbf{y}_i$, the label vector of $\mathbf{x}_i$. A graph with an adjacency matrix of $YY^\top$ will satisfy all the conditions we wanted for $K_n + W$. Based on the ordering,

we defined for the data points, the first l rows of Y will be corresponding to the labeled points and rest of the rows will be corresponding to the unlabeled data.

Based on the above descriptions, the objective function in (12), is equal to:

$$\arg\max_{V} \frac{1}{(n-1)^2} \mathbf{tr}(V^\top X H_n Y Y^\top H_n X^\top V) \qquad (12)$$

This is the objective we use to find the $d \times p$ mapping matrix V. Similar to (6), the columns of the mapping matrix are the eigenvectors corresponding to the top p eigenvalues of $X H_n Y Y^\top H_n X^\top$.

Suppose X_{ts} is a $d \times n_{ts}$ matrix that contains n_{ts} test samples. It is clear that the test points can be mapped to low-dimensional space simply by: $Z_{ts} = V^\top X_{ts}$

IV. Kernelized Version

The advantage of a linear transformation is that it explicitly states the basis of new space as a linear combination of the basis of original space. However, in many applications, a linear transformation is not capable of yielding a good representation of the data in the new space. Kernel trick is a useful method in these situations, by which, we first implicitly take the data points to a high dimensional RKHS using a non-linear function and then find the low-dimensional representation. An important aspect of our algorithm is its ability to be stated in the kernelized form.

Based on the representer theorem, the matrix V, which we find from (12) can be constructed by a linear combination of functions of data points in the Hilbert space. Let ϕ be the function in the Hilbert space. Then $V = \phi(X)\beta$. By plugging this in (12) and replacing $\phi(X)^\top \phi(X)$ by the kernel matrix K_X, we will have:

$$\begin{aligned} \arg\max_{\beta} \quad & \frac{1}{(n-1)^2} \mathbf{tr}(\beta^\top K_X H_n Y Y^\top H_n K_X \beta) \\ \text{subject to} \quad & \beta^\top K_X \beta = I \end{aligned} \qquad (13)$$

ehere β is a $n \times p$ transformation matrix. Again, suppose $Q = K_X H_n Y Y^\top H_n K_X$. The solution to (13) that determines β is the eigenvectors corresponding to the top p eigenvalues of the generalized eigenvalue problem: $Q\beta = \lambda K_X \beta$. The p-dimensional representation of the data is obtained by: $Z = \beta^\top K_X$. A popular kernel, which also works very well in our experiments, is the RBF kernel.

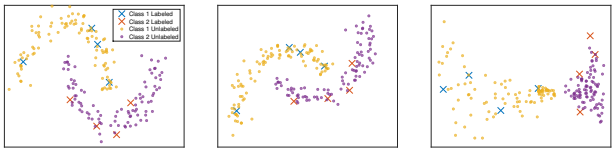
For the test data, we should first compute the kernel similarity between test and training samples. Suppose the entries of the $n \times n_{ts}$ matrix K_{ts} stores the similarities between each pair of training and test data points. Then the p- dimensional test data is: $Z_{ts} = \beta^\top K_{ts}$.

V. Experiment Results

In this section, the evaluation of applying the above algorithm on different synthetic and real datasets is presented. The parameter of the algorithm for each experiment is obtained by leave-one-out cross-validation. We also use RBF kernel similarity in (11).

A. Toy Example

First, to demonstrate the capabilities of the SSRL-PL algorithm, we apply it on a toy dataset. The two-moon dataset is a well-known for illustrating the effectiveness of an algorithm on a small set of points. The dataset has 200 samples in two almost balanced classes. Here in Fig. 1, the results of applying the SSRL-PL algorithm on the dataset is demonstrated, for both kernelized and non-kernelized versions. The number of labeled points in each class is 4, i.e. 0.04 of all points. As it can be easily seen, the algorithm is able to identify the correct labels based the label probability assignments. In the kernelized version, the new representation also provides the ability to classify the points using a linear discriminant.



(a) (b) (c)

Fig. 1: (a) Original dataset with 4 labeled data points in each class, (b) SSLR-PL without using kernel $k = 1$, (c) SSLR-PL with RBF kernel $\sigma = 0.15$, $k = 3$.

B. Demonstration and Benchmarks

Here, we present the results of applying the algorithm on more challenging datasets. The USPS dataset is used to show the generalizabilty of the algorithm and some other datasets from UCI repository are used to show the effectiveness of the algorithm in finding a good representation of data that is suitable for classification, despite the fact the dimensionality of the projected space is much lower than the dimensionality of the original space.

1) USPS: USPS hand-written digit dataset consists of 11000 data points in 10 classes. The classes are balanced and each of them has 1100 images of size 16×16 from hand-written digits 0 to 9. Therefore, the dimensionality of samples is 256. In this experiments, we randomly chose 2000 samples from them for training and the rest is only used for the testing. The training set is divided into labeled and unlabeled sets. In fact, 10% of the data is labeled. The models is trained by the training set and the obtained transformation matrix, V, is applied on both training and test sets. Figure 2 shows the result of applying kernelized SSRL-PL, with RBF kernel, on the dataset. The data is mapped into a three-dimensional space. The left-hand side plot shows the result for only labeled samples of the training set and the right-hand side plot shows the result for both the unlabeled samples of the training set and the test set. We can easily see from this plot that the algorithm is generalizable as its performance on the training set and the large unseen test set is the same.

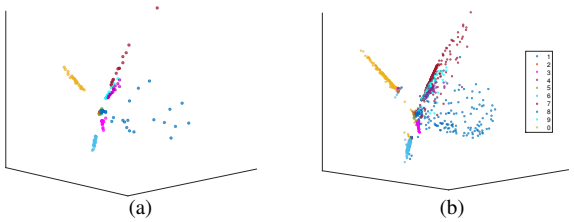


Fig. 2: (a) Labeled dataset in a 3-dimensional space, (b) Unlabeled and test dataset in a 3-dimensional space, with $k = 1$, and $\sigma = 2$.

2) Benchmark datasets: In [6], multiple benchmarks for the task of semi-supervised learning have been introduced for a fair comparison between algorithms. Datasets can be accessed publicly at **`http://olivier.chapelle.cc/ssl-book/benchmarks.html`**. The sets we have used among them are g241c, g241d, and BCI. g241c and g241d both have 1500 data points and 241 dimensions, while BCI has 400 points and 117 dimensions. For each dataset, 12 different splits exist, which divide the data into labeled and unlabeled sets. The number of labeled points based on these splits can be either 10 or 100. Therefore, the average error rate can be easily reported on these benchmarks. The table below shows the results of applying SSRL-PL on these datasets, according to the provided splits. For comparison, the results of some other algorithms are also reported in the table. These algorithms are LapSVM, LapSVMp[13], and Semi-KSC[2]. The first column of the table, which is titled by l, indicates the number of labeled points in the set.

TABLE II: Comparison of the classification error rate of the proposed algorithm with other methods for different datasets. The bold numbers show the best results

l	Algorithm	g241c	g241d	BCI
10	LapSVM	0.48 ± 0.02	0.42 ± 0.03	0.48 ± 0.03
	LapSVMp	0.49 ± 0.01	0.43 ± 0.03	0.48 ± 0.02
	Semi-KSC	**0.42 ± 0.03**	0.43 ± 0.04	0.46 ± 0.03
	SSRL-PL	0.43 ± 0.02	**0.38 ± 0.03**	**0.42 ± 0.03**
100	LapSVM	0.40 ± 0.06	0.31 ± 0.03	0.37 ± 0.04
	LapSVMp	0.36 ± 0.07	0.31 ± 0.02	0.32 ± 0.02
	Semi-KSC	0.29 ± 0.05	0.28 ± 0.05	0.22 ± 0.02
	SSRL-PL	**0.27 ± 0.05**	**0.25 ± 0.03**	**0.19 ± 0.02**

C. Real-world datasets

Now we examine the performance of the algorithm on six real-world datasets. MNIST-10K is a set of 10000 images of hand-written digits, which are randomly selected from the MNIST dataset. USPS is also set of images of hand-written digits. UMIST a face recognition dataset. The COIL-20 and SBdata are sets of images of different objects. Reuters dataset [10], contains 810000 English news stories in different categories. We followed the same procedure in [19] to obtain 10000 samples from this set in 4 categories. Other statistics of the datasets are mentioned in table III.

TABLE III: Datasets Statistics

Datasets Name	# of points	Dimensionality	# of classes
MNIST-10K	10000	784	10
USPS	11000	256	10
COIL-20	1440	1024	20
Reuters-10K	10000	2000	4
UMIST	564	750	20
SBData	3192	638	40

We compare the performance of the algorithm by multiple dictionary learning algorithms. Discriminative K-SVD (DKSVD)[23], Fisher Discrimination Dictionary Learning (FDDL)[20], and Label Consistent K-SVD (LCKSVD)[12] are three supervised dictionary learning algorithms. Also two important semi-supervised dictionary learning algorithm, i.e. OSSDL [24] and S2D2 [18].

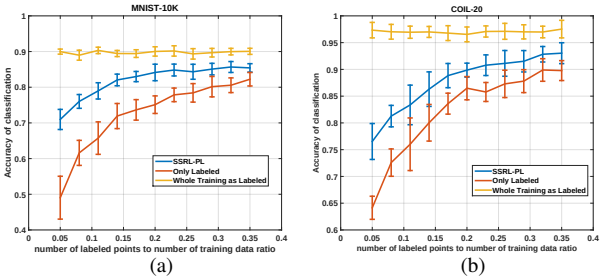
We first divide the datasets in two parts, 50% for training and 50% for test. Among the training points we choose l points as labeled and the rest unlabeled such that there is at least one labeled point in each class. We repeat this process 10 times. Results in table I show the mean and standard deviation of the classification error on the test set. As we can see, the proposed method in this work outperform the other methods . The two other semi-supervised learning algorithms also perform very well. We also include the dimensionality of the target space in the table, which shows that the reduction in dimensionality is significant.

For MNIST-10K and COIL-20 we performed another experiment. Again we first divide the datasets in half. Then for different number of labeled points we apply the SSRL-PL algorithm to the resulting training data, for 10 random

TABLE I: Comparison of classification accuracy (%). ℓ/tr = portion of the training set that is labeled. The bold numbers show the best results. p = dimensionality of the projected space, k = number of labeled neighbors for each unlabeled data point.

Dataset	l/tr	DKSVD	FDDL	LCKSVD2	OSSDL	S2D2	SSRL-PL	p	k
MNIST-10K	0.1	67.18 ± 1.4	74.32±2.8	69.91±1.2	75.15±1.7	76.18±1.5	**77.18 ±1.6**	10	5
	0.2	70.32±1.8	79.41±1.4	72.56±2.2	78.52±1.5	83.61±0.9	**85.41±2.3**	10	5
USPS	0.1	60.12±4.5	75.63±3.6	75.91±2.6	79.13±1.3	79.61±2.4	**80.15±1.9**	12	5
	0.2	66.61±4.1	80.12±1.6	78.64±1.6	81.35±1.7	**85.45±2.1**	85.31±2.3	12	5
COIL-20	0.05	52.26±3.1	68.31±3.8	70.23±3.1	81.06±3.4	80.25±3.8	**82.34±1.2**	10	5
	0.1	56.31±6.1	73.56±4.1	76.63±3.7	86.91±1.5	88.88±1.0	**89.71±0.8**	10	5
Reuters-10K	0.1	44.91±3.6	49.81±3.7	55.18±3.1	60.21±1.9	59.31±1.8	**61.12±3.1**	24	9
	0.2	49.32±1.6	57.18±1.2	59.31±1.7	65.12±2.3	65.18±3.1	**66.91±1.2**	24	9
UMIST	0.1	75.6±1.3	80.36±2.2	77.33±2.1	79.18±2.5	79.65±1.9	**81.21±2.3**	20	5
	0.2	79.2±1.6	83.78±1.2	81.18±1.3	83.41±2.1	82.11±2.3	**84.31±2.1**	20	5
SBData	0.1	40.31±3.9	52.34±1.2	51.23±2.2	49.36±2.2	50.87±2.1	**56.12±2.6**	10	5
	0.2	43.69±3.4	57.36±2.8	55.37±1.6	52.34±2.1	55.62±1.2	**61.74±1.4**	10	5

splits. We compare the performance of the algorithm with two other scenarios. 1) When only use labeled points to find the mapping V, using kernelized version of (5). 2) When we use all the labels of the training data and find the mapping V, using kernelized version of (6). Figure 3 shows the results of these experiments. As we can see the SSRL-PL performs close to the case when we know all the labels, which shows that the algorithm could convey the label information very well. The fluctuation in the whole labeled results is due to the first random split of dataset to test and train sets.



(a) (b)

Fig. 3: Classification accuracy under three different scenarios (a) MNIST-10K (b) COIL-20

VI. Bound on the Performance of the SSRL-PL Algorithm

In this section, we derive a bound on the performance of the algorithm. The bound is dependent on the way we assign the probabilities to the unlabeled data points. Of course, there are situations in which the distribution of the observed unlabeled data points in the space can make them useless, and sometimes destructive, in deriving the final mapping. However, this can happen for any semi-supervised algorithm.

Let us assume a special case of the SSRL-PL which we call winner take all, or WTA for short. In fact, for any label vector we set the element with the highest probability to one and rest of the elements to zero. Therefore, the u bottom rows of the label matrix Y will also have only 0 and 1. Also suppose X is the data matrix which contains n data points.

Consider the objective in (12). We define the following function:

$$f_X(V,Y) \triangleq \frac{1}{(n-1)^2}\,\mathbf{tr}(V^\top X H_n Y Y^\top H_n X^\top V) \qquad (14)$$

Let $V^\dagger$ be the solution to (12) when there is l labeled points and $u = n - l$ unlabeled data point in the dataset. Assume Y_p denotes the label matrix in this situation. In addition, consider another situation in which labels of all data points in X are known. In fact, a completely supervised problem. Let us denote by Y_n the label matrix in this scenario. Suppose V^* is the optimal mapping for the supervised problem. So:

$$V^\dagger = \arg\max_V f_X(V,Y_p) \qquad \text{where} \qquad {V^\dagger}^\top V^\dagger = I$$
$$V^* = \arg\max_V f_X(V,Y_n) \qquad \text{where} \qquad {V^*}^\top V^* = I$$

Our goal is to bound the following expression:

$$f_X(V^*,Y_n) - f_X(V^\dagger,Y_n) \qquad (15)$$

In fact, we want to see how much deviation exists between the transformation by V^* and the transformation by $V^\dagger$. As $f_X(V,Y)$ is a measure of similarity between the labels and the low-dimensional data points, the bound in (15) shows the extent to which the low-dimensional representation of the data by $V^\dagger$ is similar to the real labels of the data points. Note that since V^* is optimal solution for $f_X(V,Y_n)$, the quantity in (15) is always non-negative.

Lemma 2. *Suppose X is a $d \times n$ matrix of data points and Y is a $n \times C$ matrix of labels. Based on the definition in (14)*

$$f_X(V,Y) = \frac{n^2}{(n-1)^2} \parallel V^\top(\overline{XY} - \bar{X}\bar{Y}) \parallel_F^2 \qquad (16)$$

where $\parallel . \parallel_F$ is the Frobenius norm of matrix. $\bar{X}_{d\times 1}$ and $\bar{Y}_{1\times C}$ are average of data points and label vectors, respectively, and columns of $\overline{XY}_{d\times C}$ are the weighted average of data points, where weights are columns of Y.

Proof. In Appendix. □

Based on the above lemma, we can conclude that:

$$\arg\max_V f_X(V,Y) = \arg\max_V \sqrt{f_X(V,Y)} \qquad (17)$$

$V^\dagger$ and V^* are still the maximizers of $\sqrt{f_X(V,Y_p)}$ and $\sqrt{f_X(V,Y_n)}$, respectively. As we have bounded V by the constraint $V^\top V = I$, the values of $f_X(V,Y)$, and subsequently $\sqrt{f_X(V,Y)}$, are also bounded. Therefore, instead of bounding (15), we can bound the square root of the functions, i.e.:

$$\sqrt{f_X(V^*,Y_n)} - \sqrt{f_X(V^\dagger,Y_n)} \qquad (18)$$

We do this to be able to use the properties of the Frobenius norm ($\parallel . \parallel_F$ is a norm, $\parallel . \parallel_F^2$ is not).

The following theorem states the bound on (18). Some intermediate steps go to the appendix.

Theorem 3. *Suppose $\mathcal{X}$ is a unit ball in $\mathbb{R}^d$. For n samples drawn iid, according to some probability measure, from $\mathcal{X}$, where the label of only l of them is known and the rest u points are unlabeled, the mapping learned by SSRL-PL algorithm causes at most the following deviation from the mapping that maximizes the HSIC similarity measure between data points and all their revealed real labels.*

$$\sqrt{f_X(V^*,Y_n)} - \sqrt{f_X(V^\dagger,Y_n)} \;\le \frac{2(2+\sqrt{2})u}{n-1} P_e^{WTA}$$

where P_e^{WTA} is the error of WTA classifier.

Proof. In Appendix. □

As we can see from this theorem, the gap between the two functions vanishes when u is reduced, which shows the

consistency of the derived bound. Another important observation about this bound is its independence to dimensionality of original and target space. Therefore, it can be extended to the kernel version as well. Furthermore, suppose that $\hat{V} = \arg\max_{V} \lim_{n \to \infty} f_X(V, Y_n)$. In [3], it has been shown that the deviation of the $f_X(V, Y_n)$ under $\hat{V}$ and V^* is of order $O(1/\sqrt{n})$. This, together with the results of Theorem 3 can yield a generalization bound on SSRL-PL.

VII. Conclusion

We proposed a new algorithm for learning a representation of data when the label information is available for a small portion of the dataset. The algorithm tries to maximize the similarity between the new representation of data and label set, where the label set for unlabeled data is assigned probabilistically and the similarity measure is HSIC. The effectiveness of the proposed algorithm was evaluated on different datasets. We also derived a bound for the proposed algorithm which can be helpful for seeing if the presence of unlabeled data is constructive or destructive.

In terms of time complexity, the proposed algorithm is equivalent to a standard eigenvalue decomposition problem for symmetric matrices. This problem can be solved efficiently, for example, by singular value decomposition (SVD) methods. However, for faster implementation, using deep autoencoders that are able to estimate eigenvector of their input would be interesting in the future. Alternatively, one can train a network that maximizes the dependency between data points and label vector by optimizing HSIC as its objective function and stochastic gradient descent algorithm.

References

[1] M. S. Ahmed and L. Khan. Sisc: A text classification approach using semi supervised subspace clustering. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[2] C. Alzate and J. A. Suykens. A semi-supervised formulation to binary kernel spectral clustering. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012.

[3] H. Ashtiani and A. Ghodsi. A dimension-independent generalization bound for kernel supervised principal component analysis. In *Proceedings of The 1st International Workshop on Feature Extraction: Modern Questions and Challenges, NIPS*, pages 19–29, 2015.

[4] E. Bair and R. Tibshirani. Semi-supervised methods to predict patient survival from gene expression data. *PLoS Biol*, 2(4):e108, 2004.

[5] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceeding of ICML*, 2001.

[6] O. Chapelle, B. Schölkopf, A. Zien, et al. Semi-supervised learning. 2006.

[7] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *AISTATS*, pages 57–64, 2005.

[8] F. G. Cozman, I. Cohen, M. C. Cirelo, et al. Semi-supervised learning of mixture models. In *Proceeding of ICML*, pages 99–106, 2013.

[9] J. A. Drakopoulos. Bounds on the classification error of the nearest neighbor rule. In *Proceedings of ICML*, pages 203–208, 1995.

[10] Y. Y. R. T. G. GLewis, David D and F. Li. A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 2004.

[11] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic learning theory*, pages 63–77. Springer, 2005.

[12] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2651–2664, 2013.

[13] S. Melacci and M. Belkin. Laplacian support vector machines trained in the primal. *The Journal of Machine Learning Research*, 12:1149–1184, 2011.

[14] K. Nigam, A. McCallum, and T. Mitchell. Semi-supervised text classification using em. *Machine Learning*, 39:103–134, 2000.

[15] R. Nock and M. Sebban. An improved bound on the finite-sample risk of the nearest neighbor rule. *Pattern Recognition Letters*, 22(3):407–412, 2001.

[16] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *Seventh IEEE Workshop Application of Computer Vision*, 2005.

[17] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[18] A. Shrivastava, J. K. Pillai, V. M. Patel, and R. Chellappa. Learning discriminative dictionaries with partially labeled data. In *2012 19th IEEE International Conference on Image Processing*, pages 3113–3116. IEEE, 2012.

[19] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[20] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *2011 International Conference on Computer Vision*, pages 543–550. IEEE, 2011.

[21] Z.-H. You, Z. Yin, K. Han, D.-S. Huang, and X. Zhou. A semi-supervised learning approach to predict synthetic genetic interactions by combining functional and topological properties of functional gene network. *Bmc Bioinformatics*, 11(1):1, 2010.

[22] F. Zang and J.-S. Zhang. Label propagation through sparse neighborhood and its applications. *Neurocomputing*, 97:267–277, 2012.

[23] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2691–2698, 2010.

[24] X. Zhang, D. Wang, Z. Zhou, and Y. Ma. Simultaneous rectification and alignment via robust recovery of low-rank tensors. In *Advances in Neural Information Processing Systems*, pages 1637–1645, 2013.

[25] Z. Zhang, M. Zhao, and T. W. Chow. Graph based constrained semi-supervised learning framework via label propagation over adaptive neighborhood. *Knowledge and Data Engineering, IEEE Transactions on*, 27(9):2362–2376, 2015.

[26] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceeding of ICML*, 2001.

VIII. Appendix

Proof of Lemma 2. It is known that: $\sqrt{\mathbf{tr}(AA^\top)} = \| A \|_F$. We denote the i^{th} column of Y by $\mathbf{y}^i$.

$$f_X(V,Y) = \frac{1}{(n-1)^2}\,\mathbf{tr}(\overbrace{V^\top X H_n Y}^{A}\overbrace{Y^\top H_n X^\top V}^{A^\top})$$

$$= \frac{1}{(n-1)^2}\| \overbrace{V^\top X H_n Y}^{A} \|_F^2$$

$$= \frac{1}{(n-1)^2}\| V^\top \Big(\big[\sum_{i=1}^n (\mathbf{x}_i - \bar{X})\mathbf{y}_i^1 \quad \sum_{i=1}^n (\mathbf{x}_i - \bar{X})\mathbf{y}_i^2 \quad \dots \quad \sum_{i=1}^n (\mathbf{x}_i - \bar{X})\mathbf{y}_i^C \big] \Big) \|_F^2$$

$$= \frac{1}{(n-1)^2}\| nV^\top \Big(\frac{1}{n}\big[\sum_{i=1}^n \mathbf{x}_i\mathbf{y}_i^1 \quad \sum_{i=1}^n \mathbf{x}_i\mathbf{y}_i^2 \quad \dots \quad \sum_{i=1}^n \mathbf{x}_i\mathbf{y}_i^C \big] - \frac{1}{n}\big[\sum_{i=1}^n \bar{X}\mathbf{y}_i^1 \quad \sum_{i=1}^n \bar{X}\mathbf{y}_i^2 \quad \dots \quad \sum_{i=1}^n \bar{X}\mathbf{y}_i^C \big] \Big) \|_F^2$$

$$= \frac{n^2}{(n-1)^2}\| V^\top \Big(\big[\overline{X\mathbf{y}^1} \quad \overline{X\mathbf{y}^2} \quad \dots \quad \overline{X\mathbf{y}^C} \big] - \bar{X}\bar{Y} \big] \Big) \|_F^2$$

$$= \frac{n^2}{(n-1)^2}\| V^\top (\overline{XY} - \bar{X}\bar{Y}) \|_F^2$$

□

Obtaining the final result for the theorem, needs bounding both $\| \bar{Y}_n - \bar{Y}_p \|_2$ and $\| \overline{XY_n} - \overline{XY_p} \|_F$. Lets denote by $\mathbf{e}_i$ the difference between the real label vector and the assigned label vector of point $\mathbf{x}_i$, $\mathbf{e}_i = \mathbf{y}_{n_i} - \mathbf{y}_{p_i}$. For the labeled points $\mathbf{e}_i$ is an all zero vector, for the unlabeled points, if an error happens, the length of $\mathbf{e}_i$ is $\sqrt{2}$. So:

$$\| \bar{Y}_n - \bar{Y}_p \|_2 = \| \bar{\mathbf{e}} \|_2 \leq \frac{\sqrt{2}u}{n} P_e^{\text{WTA}} \tag{19}$$

Let $E = Y_n - Y_p$ and $\mathbf{e}^c$ be its c^{th} column. Let also ne^c be the number of errors for class c. Note that whether a point in class c misclassified as another class or a point in another class misclassfied as c, ne^c increases by one. The bound for $\| \overline{XY_n} - \overline{XY_p} \|_F$ is then the following:

$$\| \overline{XY_n} - \overline{XY_p} \|_F = \| \overline{XE} \|_F \leq \sum_{c=1}^C \| \overline{X\mathbf{e}^c} \|_2$$

$$= \sum_{c=1}^C \| \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i (y_{n_i}^c - y_{p_i}^c) \|_2$$

$$\leq \frac{1}{n}\sum_{c=1}^C ne^c \max_i \| \mathbf{x}_i \|_2 \leq \frac{1}{n}\sum_{c=1}^C ne^c \leq \frac{2u}{n} P_e^{\text{WTA}} \tag{20}$$

Proof of Theorem 3. Suppose $\epsilon_1 = \overline{XY_n} - \overline{XY_p}$ and $\epsilon_2 = \bar{Y}_n - \bar{Y}_p$. According to (18):

$$\sqrt{f_X(V^*, Y_n)} - \sqrt{f_X(V^\dagger, Y_n)} = \frac{n}{n-1} \times$$

$$\Big(\| V^{*\top}(\overline{XY_n} - \bar{X}\bar{Y}_n) \|_F - \| V^{\dagger\top}(\overline{XY_n} - \bar{X}\bar{Y}_n) \|_F \Big)$$

$$= \frac{n}{n-1}\Big(\| V^{*\top}(\overline{XY_p} + \epsilon_1 - \bar{X}(\bar{Y}_p + \epsilon_2)) \|_F - \| V^{\dagger t\top}(\overline{XY_p} + \epsilon_1 - \bar{X}(\bar{Y}_p + \epsilon_2)) \|_F \Big)$$

$$\overset{(a)}{\leq} \frac{n}{n-1}\Big(\| V^{*\top}(\overline{XY_p} - \bar{X}\bar{Y}_p) \|_F - \| V^{\dagger t\top}(\overline{XY_p} - \bar{X}\bar{Y}_p) \|_F + \| V^{*\top}\epsilon_1 \|_F + \| V^{\dagger\top}\epsilon_1 \|_F + \| V^{*\top}\bar{X}\epsilon_2 \|_2 + \| V^{\dagger\top}\bar{X}\epsilon_2 \|_2 \Big)$$

$$\overset{(b)}{\leq} \frac{n}{n-1}\Big(\| V^{*\top}\epsilon_1 \|_F + \| V^{\dagger\top}\epsilon_1 \|_F + \| V^{*\top}\bar{X}\epsilon_2 \|_2 + \| V^{\dagger\top}\bar{X}\epsilon_2 \|_2 \Big)$$

$$\overset{(c)}{\leq} \frac{n}{n-1}\Big(\| \epsilon_1 \|_F + \| \epsilon_1 \|_F + \| \epsilon_2 \|_2 + \| \epsilon_2 \|_2 \Big)$$

$$\overset{(d)}{\leq} \frac{2(2+\sqrt{2})u}{n-1} P_e^{\text{WTA}}$$

where inequality (a) comes from triangle inequality, (b) from the fact that $V^\dagger$ is the maximizer of the $\| V^\top(\overline{XY_p} - \bar{X}\bar{Y}_p) \|_F$, (c) from norm properties, and also the fact that orthonormal transformation does not increase the vector length, and finally (d) from (19) and (20).

□

A special case of WTA algorithms is 1-NN. In [9], [15], a bound on the performance of 1-NN was proposed which can be very helpful for our analysis. Given the underlying class-conditional distribution function is Lipschitz, the probability of error of 1-NN classifier which uses m points as the training is:

$$P_e^{\text{1-NN}} \leq 2P_e^* - \frac{C}{C-1}P_e^{*2} + \delta(m) \tag{21}$$

where P_e^* is the error of Bayesian classifier, C is the number of classes, and δ is a penalty factor as a function of number of training points which vanishes as $m \to \infty$. Using (21) we can further bound the algorithm performance which will be independent of the way we assign label to the unlabled data points.