IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 21, 2020, accepted January 31, 2020, date of publication February 4, 2020, date of current version February 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2971562

Semi-Supervised Dimensionality Reduction by Linear Compression and Stretching

ZHIGUO LONG[1,2], HUA MENG[3,4], AND MICHAEL SIOUTIS[5]
[1]School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China
[2]Institute of Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China
[3]School of Civil Engineering, Southwest Jiaotong University, Chengdu 611756, China
[4]School of Mathematics, Southwest Jiaotong University, Chengdu 611756, China
[5]Faculty of Information Systems and Applied Computer Sciences, University of Bamberg, 96047 Bamberg, Germany

Corresponding author: Hua Meng (menghua@swjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61806170, in part by the Humanities and Social Sciences Fund of Ministry of Education under Grant 18XJC72040001, and in part by the Fundamental Research Funds for the Central Universities under Grant 2682018CX25.

ABSTRACT Dimensionality reduction is a fundamental and important research topic in the field of machine learning. This paper focuses on a dimensionality reduction technique that exploits semi-supervising information in the form of pairwise constraints; specifically, these constraints specify whether two instances belong to the same class or not. We propose two dual linear methods to accomplish dimensionality reduction under that setting. These two methods overcome the difficulty of maximizing between-class difference and minimizing within-class difference at the same time, by transforming the original data into a new space in such a way that the bi-objective problem is (almost) equivalently reduced to a single objective problem. Empirical evaluations on a broad range of public datasets show that the two proposed methods are superior to several existing methods for semi-supervised dimensionality reduction.

INDEX TERMS Dimensionality reduction, pairwise constraints, PCA, FLD.

I. INTRODUCTION

High-dimensional data are common in various machine learning applications, from text document and image processing [1]–[3] to biological data analysis [4], [5]. Because of the curse of dimensionality [6], dimensionality reduction methods are fundamental to the success of many machine learning algorithms. Classical and simple dimensionality reduction techniques, such as Principal Component Analysis (PCA) [7] and Fisher Linear Discriminant (FLD) [8], are widely used in the preprocessing of data. PCA is an unsupervised technique that does not make use of any label information for dimensionality reduction, whereas FLD is a supervised technique that requires full information of labels. However, labelled data are rare, and are also expensive to acquire in practical applications. Semi-supervised techniques [9] were thus invented to overcome this difficulty for machine learning tasks.

Dimensionality reduction with semi-supervising information is an active research branch. There are various types of semi-supervising information, including pairwise constraints and incomplete class labels. Based on the type of semi-supervising information, methods for dimensionality reduction can be categorised as label-based, pairwise constraint-based, and other types. For each type, there are linear and non-linear methods. Representative label-based linear methods include: Semi-Supervised Probabilistic Principal Component Analysis (S2PPCA) [10], which exploits probabilistic PCA [11] to model the difference between feature space and label space; Semi-supervised Discriminant Analysis (SDA) [12], which is like FLD, but uses labelled data to maximize between-class difference and unlabelled data to infer intrinsic geometric structure of all the data; and many more [13]–[15]. Representative pairwise constraint-based linear methods include: constraints based Fisher Linear Discriminant (cFLD) [16], which is similar to FLD or SDA, but where the within-class difference is obtained by checking must-link constraints (i.e., pairs of instances that are known to belong to the same class or have the same label); Semi-Supervised Dimensionality Reduction (SSDR) [17], which considers differences of both must-link and cannot-link constraints (i.e., pairs of instances that are known to belong to

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

different classes or have different labels); and many more [18], [19]. Non-linear methods are usually either kernel based methods [12], [18]–[21], or embedding methods [22]–[25], which assume data to have a manifold structure and, consequently, try to embed the data into a lower dimensional space that maintains that structure. Compared to non-linear ones, linear methods are usually easier to compute and produce more understandable results. This paper focuses on linear methods.

There are many different objectives for dimensionality reduction. For example, PCA tries to maximize the separability of data projected into some subspace. For supervised and semi-supervised methods, naturally the objective would be to either maximize the difference between data points with different labels or minimize the difference between data points with the same label. However, the *bi-objective problem*, i.e. maximizing difference between classes (J_B) and minimizing difference within classes (J_W), usually does not have a feasible solution [26]. Therefore, methods have been proposed to balance the two different objectives. For example, FLD seeks to maximize $\frac{J_B}{J_W}$, because when $\frac{J_B}{J_W}$ is maximized, J_B would be relatively large and J_W would be relatively small. This is also the case for cFLD, where J_B and J_W are defined over pairwise constraints instead of data points with classes. Similarly, SSDR considers $\alpha J_B - \beta J_W$ as the objective to maximize, which can also be seen as obtaining a balance between the two objectives of maximizing J_B and minimizing J_W.

A. CONTRIBUTION

In this paper, we consider another idea to bypass the dilemma of the bi-objective problem for semi-supervised dimensionality reduction. Basically, the idea is to first transform the data in such a way that one of the objectives is achieved trivially after the transformation, and then naturally treat the resulting reduced problem as a single objective problem. For example, Fig. 1a shows two sets of data points. Based on our idea, we can first embed the original data into another space, where the within-class difference on any direction is a constant, as shown on Fig. 1b. After this transformation, it can be seen in the figure that the objective of minimizing within-class difference is trivially satisfied, because the resulting within-class difference on any direction is the same. Then one only needs to find directions maximizing between-class difference, reducing the bi-objective problem into a single-objective problem.

Based on this idea, two dual algorithms are proposed to accomplish semi-supervised dimensionality reduction. Theoretically, we show that after the proposed transformation, either the between-class difference on any direction will be a constant or the within-class difference on any direction will be relatively small (up to a constant). Based on these results, we also show that any solution of a specific single objective problem is (almost) a solution of the bi-objective problem corresponding to the transformed data. As illustrated in our experiments, these two algorithms can make classification and clustering tasks work well on various datasets, which means that important structural information in the data is well preserved by the proposed algorithms.

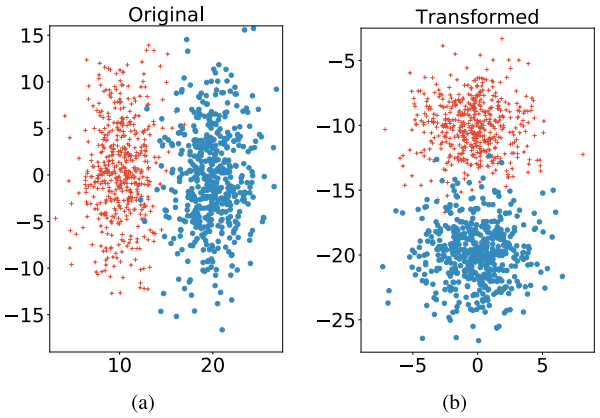


FIGURE 1. Illustration of transforming data to bypass the dilemma of the bi-objective problem. (a) Original data. (b) Transformed data.

In the sequel, we first introduce some necessary preliminaries, then describe and analyse the two new algorithms, and then evaluate them in several experiments.

II. PRELIMINARIES

In this paper, a vector is always a column vector. An *instance* $\mathbf{x}_i$ is a p-dimensional vector, i.e. $\mathbf{x}_i \in \mathbb{R}^p$. A *label* y_i is a scalar in $\mathbb{R}$. *Pairwise constraints* are more general than labels and provide *semi-supervising information* for machine learning tasks through a set $\mathcal{C}$ and a set $\mathcal{M}$ of *cannot-link* and *must-link* constraints respectively. In particular, given a set of instances $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the set $\mathcal{C}$ of cannot-link constraints consists of pairs of instances, each of which indicates that the two instances have different labels (or, equivalently, belong to different classes); and the set $\mathcal{M}$ of must-link constraints consists of pairs of instances, each of which indicates that the two instances have the same label (or, equivalently, belong to the same class).

Common objectives of dimensionality reduction include maximizing between-class difference (J_B) and minimizing within-class difference (J_W). J_B and J_W can be measured in several ways. With fully supervising information, methods like FLD consider the sum of distances between instances within a class and the centroid of this class, as the measure of J_W, and consider the sum of distances between centroids of classes as the measure of J_B. With semi-supervising information, methods like cFLD consider using must-link constraints to form connected components, and use the components as classes; J_B and J_W can then be measured similarly as in the fully supervising case. However, this is not the only way for semi-supervising information. Methods like SSDR choose to directly add up the distances between pairs in must-link constraints as J_W, and add up the distances between pairs in cannot-link constraints as J_B. Compared to the one used by cFLD, this measure does not need to construct

connected components and exploits both must-link and cannot-link information. We follow this convention in this article. Formally, between-class difference J_B is measured by

$$\sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{C}} \|\mathbf{x}_j - \mathbf{x}_k\|^2, \tag{1}$$

where $\mathcal{C}$ is the set of cannot-link constraints. Within-class difference J_W is measured by

$$\sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{M}} \|\mathbf{x}_j - \mathbf{x}_k\|^2, \tag{2}$$

where $\mathcal{M}$ is the set of cannot-link constraints.

Let $\mathbf{S}_B = \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{C}}(\mathbf{x}_j - \mathbf{x}_k)(\mathbf{x}_j - \mathbf{x}_k)^T$, and $\mathbf{S}_W = \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{M}}(\mathbf{x}_j - \mathbf{x}_k)(\mathbf{x}_j - \mathbf{x}_k)^T$. It is clear that between-class difference and within-class difference can be rewritten as $\mathrm{tr}(\mathbf{S}_B)$ and $\mathrm{tr}(\mathbf{S}_W)$, where $\mathrm{tr}(A)$ (the *trace* of A) for a square matrix A calculates the sum of main diagonal elements of A, i.e., $\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$ for A being an $n \times n$ matrix.

Linear transformation is a common way to reduce dimensionality of a dataset. Denote by $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ the original data, and by $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_K)$ a transformation matrix. Each $\mathbf{x}_i$ is transformed into $\mathbf{U}^T\mathbf{x}_i$ when $\mathbf{X}$ is transformed by $\mathbf{U}$ to a subspace. For the transformed data by $\mathbf{U}$, the between-class difference can be calculated with

$$J_B(\mathbf{U}) = \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{C}} \|\mathbf{U}^T\mathbf{x}_j - \mathbf{U}^T\mathbf{x}_k\|^2 = \mathrm{tr}(\mathbf{U}^T\mathbf{S}_B\mathbf{U}), \tag{3}$$

and the within-class difference can be calculated with

$$J_W(\mathbf{U}) = \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{M}} \|\mathbf{U}^T\mathbf{x}_j - \mathbf{U}^T\mathbf{x}_k\|^2 = \mathrm{tr}(\mathbf{U}^T\mathbf{S}_W\mathbf{U}). \tag{4}$$

Dimensionality reduction with linear transformation can be formally characterized by the following optimization problem:

$$\begin{aligned} &\max_{\mathbf{U}} J_B(\mathbf{U}) \\ &\min_{\mathbf{U}} J_W(\mathbf{U}) \\ &\text{subject to} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}_K \end{aligned} \tag{5}$$

However, the above bi-objective problem usually does not have a solution, because a direction that results in large between-class difference is unlikely to have a small within-class difference, or vice versa. Therefore, over the years researchers proposed approximations to such optimization problems, where the goal is to find a sub-optimal $\mathbf{U}$ such that the total between-class difference is relatively large, while the total within-class difference remains relatively small. One should note that the measurements of differences in these models are not precisely the ones defined here, but the essential idea is the same, as mentioned above. For example, cFLD considers the ratio of between-class difference to within-class difference as the objective function that approximates the above bi-objective functions, i.e. $\max_{\mathbf{U}} J_B(\mathbf{U})/J_W(\mathbf{U})$, while SSDR considers the weighted subtraction of within-class difference from between-class difference as the approximation, i.e. $\max_{\mathbf{U}} \alpha J_B(\mathbf{U}) - \beta J_W(\mathbf{U})$.

Unfortunately, the objective functions of cFLD and SSDR can only find directions that result in a relatively small total within-class difference and a relatively large between-class difference, i.e. a compromise between both objectives. The result of dimensionality reduction by them is thus sometimes unsatisfactory [17] (we also demonstrate this in our experimentation).

In this paper, we consider a different idea to approximate the bi-objective problem. The idea consists of two steps. The first step is to *compress* or *stretch* the data with respect to some directions. In this way, we show that either the between-class difference or the within-class difference of the transformed data becomes (almost) the same for any direction. Since after the first step one of the objective functions will have obtained an (almost) constant value, the bi-objective problem is reduced to a single-objective problem that is much easier to solve. Then, the second step is to find directions that result in either smallest within-class difference or largest between-class difference (depending on which of the objective functions was replaced by a constant value) for the transformed data. Dimensionality reduction is achieved by adjusting the number of selected directions in the second step. In what follows, we introduce two dual methods based on this idea.

III. APPROACH

A. INCREASING BETWEEN-CLASS DIFFERENCES

This method first projects the data by some directions, so that the between-class difference becomes the same for every direction after projection.

Suppose the eigenvalues of $\mathbf{S}_B$ are $\lambda_1 \geq \ldots \geq \lambda_p$ and the corresponding unit eigenvectors are $\mathbf{e}_1, \ldots, \mathbf{e}_p$. To make between-class difference constant for every direction, we construct projection directions by adjusting the eigenvectors. First, we choose the eigenvectors corresponding to the first i largest eigenvalues as projecting directions, and discard the others. This is because large eigenvalues indicate large differences in the data, and are thus more likely to correspond to directions that can distinguish between-class instances well; on the other hand, directions with too small eigenvalues are more likely to be directions with noise, and we would not want data to be projected into those directions. Secondly, we construct projection directions as

$$\mathbf{V}_S = (\sqrt{\frac{\lambda_1}{\lambda_1}}\mathbf{e}_1, \ldots, \sqrt{\frac{\lambda_1}{\lambda_i}}\mathbf{e}_i). \tag{6}$$

Here, $\mathbf{V}_S$ is a matrix where columns are *stretched* eigenvectors. In fact, $\mathbf{V}_S$ gives a linear transformation from R^n to R^i. The coordinates of the transformed data $\mathbf{X}\mathbf{V}_S$ on each direction are magnified because of the coefficient $\sqrt{\frac{\lambda_1}{\lambda_j}}\mathbf{e}_j$ ($\lambda_j \leq \lambda_1$). Therefore, the value of between-class difference on each direction $\mathbf{e}_j$ also becomes larger by $\frac{\lambda_1}{\lambda_j}$ times.

Let us denote by $J_B(\mathbf{w}, \mathbf{V}_S) = \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{C}} \|\mathbf{w}^T\mathbf{V}_S^T\mathbf{x}_j - \mathbf{w}^T\mathbf{V}_S^T\mathbf{x}_k\|^2$ the *between-class difference on the direction* of $\mathbf{w}$ for $\mathbf{X}\mathbf{V}_S$. The following proposition shows that, after

projection by $\mathbf{V}_S$, the between-class difference becomes the same for *every direction* in space R^i.

Proposition 1: For any unit vector $\mathbf{w} \in R^i$, $J_B(\mathbf{w}, \mathbf{V}_S) = \lambda_1$.

Proof:

$$
\begin{aligned}
&J_B(\mathbf{w}, \mathbf{V}_S) \\
&= \sum_{(\mathbf{x}_j, \mathbf{x}_k) \in \mathcal{C}} \|\mathbf{w}^T \mathbf{V}_S^T \mathbf{x}_j - \mathbf{w}^T \mathbf{V}_S^T \mathbf{x}_k\|^2 \\
&= \sum_{(\mathbf{x}_j, \mathbf{x}_k) \in \mathcal{C}} \mathbf{w}^T \mathbf{V}_S^T (\mathbf{x}_j - \mathbf{x}_k)(\mathbf{x}_j - \mathbf{x}_k)^T (\mathbf{w}^T \mathbf{V}_S^T)^T \\
&= \mathbf{w}^T \mathbf{V}_S^T \mathbf{S}_B \mathbf{V}_S \mathbf{w} \\
&= \mathbf{w}^T (\sqrt{\frac{\lambda_1}{\lambda_1}}\mathbf{e}_1, \ldots, \sqrt{\frac{\lambda_1}{\lambda_i}}\mathbf{e}_i)^T \mathbf{S}_B (\sqrt{\frac{\lambda_1}{\lambda_1}}\mathbf{e}_1, \ldots, \sqrt{\frac{\lambda_1}{\lambda_i}}\mathbf{e}_i)\mathbf{w} \\
&= \mathbf{w}^T (\sqrt{\frac{\lambda_1}{\lambda_1}}\mathbf{e}_1, \ldots, \sqrt{\frac{\lambda_1}{\lambda_i}}\mathbf{e}_i)^T (\lambda_1\sqrt{\frac{\lambda_1}{\lambda_1}}\mathbf{e}_1, \ldots, \lambda_i\sqrt{\frac{\lambda_1}{\lambda_i}}\mathbf{e}_i)\mathbf{w} \\
&= \mathbf{w}^T \operatorname{diag}(\lambda_1, \lambda_1, \ldots, \lambda_1)\mathbf{w} \\
&= \lambda_1 \mathbf{w}^T \mathbf{w} \\
&= \lambda_1.
\end{aligned}
$$

□

In order to find directions with largest between-class difference and smallest within-class difference in the new space R^i, we rewrite the bi-objective problem (5) as the following *projected bi-objective problem*.

$$
\begin{aligned}
&\max_{\mathbf{U}} J_B(\mathbf{U}, \mathbf{V}_S) = \sum_{(\mathbf{x}_j, \mathbf{x}_k) \in \mathcal{C}} \|\mathbf{U}^T \mathbf{V}_S^T \mathbf{x}_j - \mathbf{U}^T \mathbf{V}_S^T \mathbf{x}_k\|^2 \\
&\min_{\mathbf{U}} J_W(\mathbf{U}, \mathbf{V}_S) = \sum_{(\mathbf{x}_j, \mathbf{x}_k) \in \mathcal{M}} \|\mathbf{U}^T \mathbf{V}_S^T \mathbf{x}_j - \mathbf{U}^T \mathbf{V}_S^T \mathbf{x}_k\|^2 \\
&\text{subject to} \quad \mathbf{U}^T \mathbf{U} = \mathbf{I}_K
\end{aligned} \tag{7}
$$

Here, $J_B(\mathbf{U}, \mathbf{V}_S)$ and $J_W(\mathbf{U}, \mathbf{V}_S)$ are the total between-class difference and the total within-class difference, respectively, on directions $\mathbf{U}$ for the projected data $\mathbf{X}\mathbf{V}_S$. Note that by Proposition 1, for each direction $\mathbf{w}$ of R^i, the difference $J_B(\mathbf{w}, \mathbf{V}_S)$ is the same. Then, the value of the first objective in (7) is always a constant, as shown in the following proposition.

Proposition 2: Let $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_K)$ *and* $\mathbf{U}' = (\mathbf{u}'_1, \ldots, \mathbf{u}'_K)$, *where* $\mathbf{u}_j, \mathbf{u}'_j \in R^i$ *and* $\mathbf{U}^T\mathbf{U} = \mathbf{U}'^T\mathbf{U}' = \mathbf{I}_K$. *Then* $J_B(\mathbf{U}, \mathbf{V}_S) = J_B(\mathbf{U}', \mathbf{V}_S)$.

Proof:

$$
\begin{aligned}
J_B(\mathbf{U}, \mathbf{V}_S) &= \sum_{(\mathbf{x}_j, \mathbf{x}_k) \in \mathcal{C}} \|\mathbf{U}^T \mathbf{V}_S^T \mathbf{x}_j - \mathbf{U}^T \mathbf{V}_S^T \mathbf{x}_k\|^2 \\
&= \operatorname{tr}(\mathbf{U}^T \mathbf{V}_S^T \mathbf{S}_B \mathbf{V}_S \mathbf{U}) \\
&= \sum_{j=1}^{K} J_B(\mathbf{u}_j, \mathbf{V}_S) \\
&= \sum_{j=1}^{K} J_B(\mathbf{u}'_j, \mathbf{V}_S) \text{ (By Proposition 1)} \\
&= \sum_{(\mathbf{x}_j, \mathbf{x}_k) \in \mathcal{C}} \|\mathbf{U}'^T \mathbf{V}_S^T \mathbf{x}_j - \mathbf{U}'^T \mathbf{V}_S^T \mathbf{x}_k\|^2 = J_B(\mathbf{U}', \mathbf{V}_S).
\end{aligned}
$$

□

Therefore, the bi-objective optimization problem (7) is equivalent to the following single-objective problem for projected within-class difference.

$$
\begin{aligned}
&\min_{\mathbf{U}} \sum_{(\mathbf{x}_j, \mathbf{x}_k) \in \mathcal{M}} \|\mathbf{U}^T \mathbf{V}_S^T \mathbf{x}_j - \mathbf{U}^T \mathbf{V}_S^T \mathbf{x}_k\|^2 \\
&\text{subject to} \quad \mathbf{U}^T \mathbf{U} = \mathbf{I}_K.
\end{aligned} \tag{8}
$$

The theorem below formalizes the observation.

Theorem 3: Any solution $\mathbf{U}$ *of the optimization problem* (8) *is a solution of the optimization problem* (7).

Note that $J_W(\mathbf{U}, \mathbf{V}_S) = \sum_{(\mathbf{x}_j, \mathbf{x}_k) \in \mathcal{M}} \|\mathbf{U}^T \mathbf{V}_S^T \mathbf{x}_j - \mathbf{U}^T \mathbf{V}_S^T \mathbf{x}_k\|^2 = \operatorname{tr}(\mathbf{U}^T \mathbf{V}_S^T \mathbf{S}_W \mathbf{V}_S \mathbf{U})$. Therefore, the optimization problem (8) is a typical *eigenvalue problem* [27], and it can be easily and efficiently solved by computing the unit eigenvectors of $\mathbf{V}_S^T \mathbf{S}_W \mathbf{V}_S$ corresponding to the largest eigenvalues.

The algorithm BWDR shown in Algorithm 1 shows the detailed steps for dimensionality reduction based on the above idea. Note that in the first step, the first i largest eigenvalues are selected by computing the *aggregated contribution ratio* α_i of $\mathbf{e}_i$ as follows

$$
\alpha_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{j=1}^{p} \lambda_j}. \tag{9}
$$

With this notion, for a given threshold t_0, we consider the first i directions with $\alpha_i \leq t_0$ and $\alpha_{i+1} > t_0$ to have relatively larger differences and thus contain useful between-class information.

Algorithm 1 BWDR: Dimensionality Reduction by Increasing Between-Class Differences

Input: dataset $\mathbf{X}$; must-link $\mathcal{M}$; cannot-link $\mathcal{C}$; $t_0 \in (0, 1)$; required lower dimensionality K.
Output: lower-dimensional representation $\mathbf{X}'$.

1 Compute the difference matrices $\mathbf{S}_B$ and $\mathbf{S}_W$;
2 Compute all the eigenvalues of $\mathbf{S}_B$: $\lambda_1 \geq \ldots \geq \lambda_p$, and the corresponding unit eigenvectors: $\mathbf{e}_1, \ldots, \mathbf{e}_p$;
3 Find i s.t. $\alpha_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{j=1}^{p} \lambda_j} \leq t_0$ and $\alpha_{i+1} > t_0$;
4 **if** $i < K$ **then** $i \leftarrow K$;
5 $\mathbf{V}_S \leftarrow (\sqrt{\frac{\lambda_1}{\lambda_1}}\mathbf{e}_1, \ldots, \sqrt{\frac{\lambda_1}{\lambda_i}}\mathbf{e}_i)$;
6 $\mathbf{X} \leftarrow \mathbf{X}\mathbf{V}_S$;
7 $\mathbf{S}'_W \leftarrow \mathbf{V}_S^T \mathbf{S}_W \mathbf{V}_S$;
8 Compute the eigenvalues of $\mathbf{S}'_W$: $\mu_1 \leq \ldots \leq \mu_i$, and the corresponding unit eigenvectors: $\mathbf{u}_1, \ldots, \mathbf{u}_i$;
9 $\mathbf{X}' \leftarrow (\mathbf{u}_1, \ldots, \mathbf{u}_K)^T \mathbf{X}$;
10 **return** $\mathbf{X}'$.

B. DECREASING WITHIN-CLASS DIFFERENCES

The dual of the previous method is to decrease the within-class differences first. Suppose all the eigenvalues of $\mathbf{S}_W$ are $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$, and the corresponding unit eigenvectors are $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p$. This time we decrease the within-class differences on some directions, so that after projection the within-class difference becomes almost constant for every direction. To this end, we construct the projection matrix as a collection of compressed eigenvectors:

$$\mathbf{V}_C = (\sqrt{\frac{\lambda_i}{\lambda_1}}\mathbf{e}_1, \ldots, \sqrt{\frac{\lambda_i}{\lambda_i}}\mathbf{e}_i, \mathbf{e}_{i+1}, \ldots, \mathbf{e}_p). \tag{10}$$

When $\mathbf{X}$ is transformed to $\mathbf{X}\mathbf{V}_C$, the difference on the direction $\mathbf{e}_j$ with $1 \leq j \leq i$ becomes smaller, as the coordinate of the projection on $\mathbf{e}_j$ is decreased by $\sqrt{\frac{\lambda_i}{\lambda_j}}$ times.

Let $J_W(\mathbf{w}, \mathbf{V}_C) = \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{M}} \|\mathbf{w}^T\mathbf{V}_C^T\mathbf{x}_j - \mathbf{w}^T\mathbf{V}_C^T\mathbf{x}_k\|^2$ be the *within-class difference on the direction* of $\mathbf{w}$ for the transformed data $\mathbf{X}\mathbf{V}_C$. Then the within-class difference of $\mathbf{X}\mathbf{V}_C$ on any direction will be no larger than λ_i, as ensured by the following proposition.

Proposition 4: For any unit vector $\mathbf{w} \in R^p$, $J_W(\mathbf{w}, \mathbf{V}_C) \leq \lambda_i$.

Proof:

$$\begin{aligned}
J_W(\mathbf{w}, \mathbf{V}_C) &= \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{M}} \|\mathbf{w}^T\mathbf{V}_C^T\mathbf{x}_j - \mathbf{w}^T\mathbf{V}_C^T\mathbf{x}_k\|^2 \\
&= \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{M}} \mathbf{w}^T\mathbf{V}_C^T(\mathbf{x}_j - \mathbf{x}_k)(\mathbf{x}_j - \mathbf{x}_k)^T(\mathbf{w}^T\mathbf{V}_C^T)^T \\
&= \mathbf{w}^T\mathbf{V}_C^T\mathbf{S}_W\mathbf{V}_C\mathbf{w} \\
&= \mathbf{w}^T(\sqrt{\frac{\lambda_i}{\lambda_1}}\mathbf{e}_1, \ldots, \sqrt{\frac{\lambda_i}{\lambda_i}}\mathbf{e}_i, \mathbf{e}_{i+1}, \ldots, \mathbf{e}_p)^T\mathbf{S}_W \\
&\quad (\sqrt{\frac{\lambda_i}{\lambda_1}}\mathbf{e}_1, \ldots, \sqrt{\frac{\lambda_i}{\lambda_i}}\mathbf{e}_i, \mathbf{e}_{i+1}, \ldots, \mathbf{e}_p)\mathbf{w} \\
&= \mathbf{w}^T \operatorname{diag}(\lambda_i, \ldots, \lambda_i, \lambda_{i+1}, \ldots, \lambda_p)\mathbf{w} \\
&\leq \lambda_i\mathbf{w}^T\mathbf{w} \\
&= \lambda_i.
\end{aligned}$$

□

From the above proof, we can easily observe the following conclusion.

Corollary 5: For $\mathbf{V}_C = (\sqrt{\frac{\lambda_i}{\lambda_1}}\mathbf{e}_1, \ldots, \sqrt{\frac{\lambda_i}{\lambda_i}}\mathbf{e}_i, \mathbf{e}_{i+1}, \ldots, \mathbf{e}_p)$, *if* $i = p$, *then for any unit vector* $\mathbf{w} \in R^p$, $J_W(\mathbf{w}, \mathbf{V}_C) = \lambda_i$.

For the transformed data $\mathbf{X}\mathbf{V}_C$, the optimization problem of finding directions with largest between-class differences and smallest within-class differences can be written as a projected bi-objective problem:

$$\begin{aligned}
&\max_{\mathbf{U}} J_B(\mathbf{U}, \mathbf{V}_C) = \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{C}} \|\mathbf{U}^T\mathbf{V}_C^T\mathbf{x}_j - \mathbf{U}^T\mathbf{V}_C^T\mathbf{x}_k\|^2 \\
&\min_{\mathbf{U}} J_W(\mathbf{U}, \mathbf{V}_C) = \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{M}} \|\mathbf{U}^T\mathbf{V}_C^T\mathbf{x}_j - \mathbf{U}^T\mathbf{V}_C^T\mathbf{x}_k\|^2 \\
&\text{subject to} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}_K
\end{aligned} \tag{11}$$

Note that by Proposition 4, the within-class difference on each direction is no larger than λ_i, and these differences are not all the same. This case is different from the former between-class one. However, the total within-class difference on any K collection of orthonormal directions is bounded by a (small) constant, as asserted by the following proposition.

Proposition 6: Suppose $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_K)$ *with* $\mathbf{U}^T\mathbf{U} = \mathbf{I}_K$, *where* $\mathbf{u}_j \in R^p$ $(j = 1, \ldots, K)$. *Then* $J_W(\mathbf{U}, \mathbf{V}_C) = \sum_{j=1}^K J_W(\mathbf{u}_j) \leq K\lambda_i$.

Proof: By Proposition 4, we know for any unit vector $\mathbf{u} \in R^p$, $J_W(\mathbf{u}, \mathbf{V}_C) \leq \lambda_i$. Then

$$\begin{aligned}
J_W(\mathbf{U}, \mathbf{V}_C) &= \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{M}} \|\mathbf{U}^T\mathbf{V}_C^T\mathbf{x}_j - \mathbf{U}^T\mathbf{V}_C^T\mathbf{x}_k\|^2 \\
&= \operatorname{tr}(\mathbf{U}^T\mathbf{V}_C^T\mathbf{S}_W\mathbf{V}_C\mathbf{U}) = \sum_{j=1}^K J_W(\mathbf{u}_j, \mathbf{V}_S) \leq K\lambda_i.
\end{aligned}$$

□

With that property, when K and λ_i are small, the within-class difference given any $\mathbf{U}$ will also be small, and different $\mathbf{U}$ will result in similar within-class difference. Therefore, ignoring the objective $\max_{\mathbf{U}} J_B(\mathbf{U}, \mathbf{V}_C)$ will not change the quality of a solution significantly. In other words, the bi-objective problem (11) can be *approximated* by the following projected single-objective problem:

$$\begin{aligned}
&\max_{\mathbf{U}} J_B(\mathbf{U}, \mathbf{V}_C) = \sum_{(\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{C}} \|\mathbf{U}^T\mathbf{V}_C^T\mathbf{x}_j - \mathbf{U}^T\mathbf{V}_C^T\mathbf{x}_k\|^2 \\
&\text{subject to} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}_K.
\end{aligned} \tag{12}$$

The theorem below formalizes the aforementioned observation.

Theorem 7: Any solution $\mathbf{U}$ *of the optimization problem* (12) *is an approximation of a solution* $\mathbf{U}'$ *of the optimization problem* (11), *in the sense that* $J_B(\mathbf{U}, \mathbf{V}_C) \geq J_B(\mathbf{U}', \mathbf{V}_C)$ *and* $|J_W(\mathbf{U}, \mathbf{V}_C) - J_W(\mathbf{U}', \mathbf{V}_C)| \leq K\lambda_i$. *In addition, when* $i = p$, $J_B(\mathbf{U}, \mathbf{V}_C) = J_B(\mathbf{U}', \mathbf{V}_C)$ *and* $J_W(\mathbf{U}, \mathbf{V}_C) = J_W(\mathbf{U}', \mathbf{V}_C)$, *i.e.* $\mathbf{U}$ *is a solution of* (11).

Therefore, in order to find better directions for dimensionality reduction, we solve (12) instead of (11). By Theorem 7, the effect of the solution to (12) will be similar to that of (11) for producing large between-class difference and small within-class difference. Note that (12) is again an eigenvalue problem, and can be solved by finding K unit eigenvectors of $\mathbf{V}_C S_B \mathbf{V}_C^T$ corresponding to the largest K eigenvalues. Algorithm 2 shows the detailed steps for dimensionality reduction based on the above idea.

Remark 1: For both BWDR and WBDR, here we choose to stretch or to compress eigenvectors, so that the eigenvalues are increased or decreased respectively to some value λ_1 or λ_i. In practice, one can choose any positive value that is appropriate instead of λ_1 or λ_i. For example, considering computation error, we might prefer a larger value like 1.0 over a small value like 10^{-6} as the target value.

In the algorithm, to determine the first i directions whose within-class difference is to be decreased, we can again use

Algorithm 2 WBDR: Dimensionality Reduction by Compression of Within-Class Differences

Input: Dataset $\mathbf{X}$, Must-link $\mathcal{M}$; Cannot-link $\mathcal{C}$; $t_0 \in (0, 1)$, Required Dimensionality K.
Output: Lower-dimensional representation $\mathbf{X}'$.

Compute the difference matrices $\mathbf{S}_W$ and $\mathbf{S}_B$;
Compute all the eigenvalues of $\mathbf{S}_W$: $\lambda_1 \geq \ldots \geq \lambda_p$, and the corresponding unit eigenvectors: $\mathbf{e}_1, \ldots, \mathbf{e}_p$;
Find i s.t. $\alpha_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{j=1}^{p} \lambda_j} \leq t_0$ and $\alpha_{i+1} > t_0$;
if $i < K$ **then** $i \leftarrow K$;
$\mathbf{V}_C \leftarrow (\sqrt{\frac{\lambda_i}{\lambda_1}}\mathbf{e}_1, \ldots, \sqrt{\frac{\lambda_i}{\lambda_i}}\mathbf{e}_i, \mathbf{e}_{i+1}, \ldots, \mathbf{e}_p)$;
$\mathbf{X} \leftarrow \mathbf{X}\mathbf{V}_C$;
$\mathbf{S}'_B \leftarrow \mathbf{V}_C^T \mathbf{S}_B \mathbf{V}_C$;
Compute the eigenvalues of $\mathbf{S}'_B$: $\mu_1 \geq \ldots \geq \mu_i$, and the corresponding unit eigenvectors: $\mathbf{u}_1, \ldots, \mathbf{u}_i$;
$\mathbf{X}' \leftarrow (\mathbf{u}_1, \ldots, \mathbf{u}_K)^T \mathbf{X}$;
return $\mathbf{X}'$.

the aggregated contribution ratio α_j: given a threshold t_0, if $\alpha_i \leq t_0$ and $\alpha_{i+1} > t_0$, then the first i directions will be selected.

IV. EXPERIMENTS

A. DATASETS

In the experiments, we used 16 UCI datasets [28], and two image datasets, viz., the Extended Yale Face Database B (YaleB) [29] and a subset of MNIST [30]. The information of these datasets are listed in Table 1. In all of the figures, C represents the percentage of constraints that are used for semi-supervised dimensionality reduction. The constraints are created by randomly sampling pairs of instances and checking their labels (if they have the same label then must-link constraints are created, and cannot-link constraints otherwise).

B. EFFECTS OF THE THRESHOLD

In the two proposed algorithms, there is a parameter t_0. Here we briefly discuss the choice of the value for t_0. A representative result on one UCI dataset with different values of the threshold t_0 is shown in Fig. 2. The results on the other chosen datasets are similar. From this result, we can see how the accuracy of both methods is affected by the value of the threshold. When t_0 is small, the accuracy is relatively low. With t_0 increasing, the accuracy first becomes higher but then drops before it rises again. This is expected, as the threshold might exclude some projection directions that are useful or include directions that are harmful. On the other hand, when BWDR takes $t_0 = 0.95$ or WBDR takes $t_0 = 1$, the performance is relatively good for this dataset. This is also usually the case for the other datasets used in this article. Therefore, in all of the following experiments, we fix the value of t_0 for BWDR at 0.95 and for WBDR at 1.0.

TABLE 1. Datasets information. The first 16 ones are UCI datasets. Here "(N, D, L)" represents the number of instances, the number of features, and the number of classes (labels), respectively.

No.	(N, D, L)	Description
1	(569, 30, 2)	breast cancer Wisconsin (diagnostic)
2	(1080, 856, 9)	free text business descriptions
3	(540, 18, 2)	climate model input values
4	(270, 13, 2)	heart disease data
5	(155, 19, 2)	hepatitis data
6	(606, 100, 2)	hill/valley of graph with noise data
7	(1560, 617, 2)	sound features of speaking letters (1st set)
8	(1559, 617, 26)	sound features of speaking letters (5th set)
9	(3196, 36, 2)	chess (King Rook vs. King Pawn)
10	(57, 16, 2)	labour relations
11	(2000, 216, 10)	features of handwritten numerals
12	(2000, 64, 10)	features of handwritten numerals
13	(2000, 47, 10)	features of handwritten numerals
14	(210, 7, 3)	measurements of geometrical properties of kernels of wheat
15	(2100, 19, 7)	image segmentation data
16	(1593, 256, 10)	handwritten digits
YaleB	(2414, 1024,38)	near frontal images under different illuminations, around 64 images for each of 38 individuals
MNIST	(6000,784,10)	a smaller dataset of handwritten digits constructed from the original MNIST training dataset, by randomly selecting 600 instances for each label.

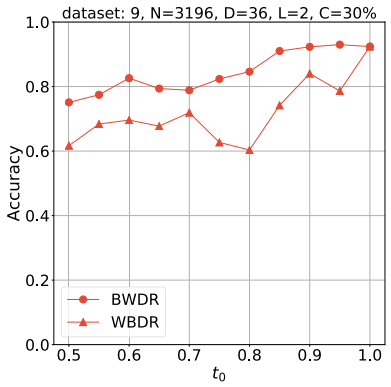


FIGURE 2. Result on one UCI dataset with different values of the threshold t_0.

C. EFFECTS OF THE NUMBER OF CONSTRAINTS

As we are considering semi-supervised dimensionality reduction, the number of constraints used for finding proper projection directions affects the performance of the methods. To illustrate this, we show the results of varying the number of constraints on two UCI datasets in Fig. 3. As we can see from Fig. 3, the overall trend is that the more semi-supervising information is available, the better the performance is. Moreover, relatively good performance is achieved with only a small portion of semi-supervising information, e.g., the performance becomes stable when 0.002% must-link and cannot-link constraints are available for dataset 12, and for dataset 2 that portion is about 0.04%. These results indicate that the performance of the two proposed methods would be positively affected by just a small number of

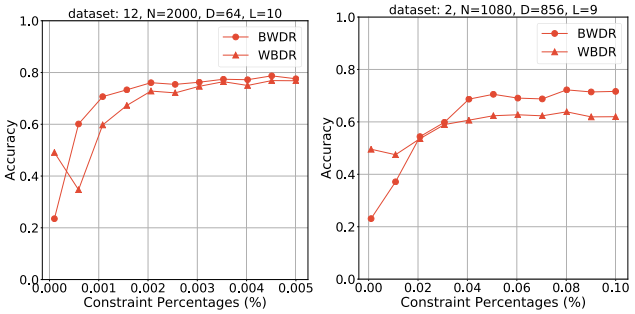


FIGURE 3. Results on three UCI datasets with different number of constraints.

semi-supervising constraints, or, in other words, that the methods can achieve relatively good results with only limited supervising information. In all the following evaluations, if not specified otherwise, we will set $\mathsf{C} = 30\%$.

D. RESULTS ON UCI DATASETS

1) 1-NN CLASSIFICATION

A usual measurement for the performance of dimensionality reduction methods is to perform 1-NN classification with the data after dimensionality reduction. In all experiments of 1-NN classification, results are averaged over 3 runs of 5-fold cross validation. Specifically, for each run, 4 folds of data are used to learn the projections, and the other fold is used to test the 1-NN classifier trained with the 4 folds of projected data. The target dimensionalities of data are 1 to 9, that is, we reduce data to one dimension, two dimensions, ..., and nine dimensions, respectively.

Table 2 shows the accuracy results of 1-NN classification on UCI datasets after dimensionality reduction. For conciseness, for each dataset and each method, we only report the maximal accuracy for the resulting dimensionalities of 1 to 9. For almost all of the tested datasets, either BWDR or WBDR achieves the best result. For the one dataset on which neither of the two methods is the best, viz., dataset 10, we can see that the difference with the best result is actually small. Particularly, the best accuracy is 0.90 while BWDR has 0.88. It is also worth noting that for dataset 1 and dataset 7, the difference between the performance of BWDR and WBDR is quite large. This indicates that the selected projecting directions of BWDR and WBDR can be very different.

Fig. 4 shows the more detailed results for three of these datasets, viz., dataset 7, dataset 9, and dataset 11. Interestingly, the two proposed methods can sometimes have a much higher accuracy in low dimensionality (e.g., 1 or 2), compared to the other methods. In most cases, as the resulting dimensionality increases, the accuracy of each of the methods also increases and quickly becomes stable. The two proposed methods have dominating accuracy for most of the resulting dimensionalities. As opposed to the two proposed methods, cFLD is not good for all cases and it can even fail in some cases (e.g., the third dataset in the figure, and SSDR and PCA usually have low accuracy in low dimensionality.

2) K-MEANS CLUSTERING RESULTS

Clustering is another measurement for the performance of dimensionality reduction methods. To measure the accuracy of clustering, following [31], we use the normalized "rand index":

$$\sum_{i>j} = \frac{1\{1\{c_i = c_j\} = 1\{\hat{c}_i = \hat{c}_j\}\}}{0.5m(m-1)}, \tag{13}$$

where m is the number of clusters, $1\{\text{True}\} = 1$ and $1\{\text{False}\} = 0$, and c_i and $\hat{c}_i$ are the real class and the predicted class for the i-th instance, respectively. Intuitively, it measures how many pairs of instances are clustered correctly (same class instances are clustered as same class and different class instances are clustered as different class). It is normalized so that the number of different-class pairs is equal to the number of same-class pairs.

The results of clustering on UCI datasets after dimensionality reduction are shown shown in Table 3. The number of clusters was chosen to be the same as the number of classes in the dataset. The numbers denote maximal clustering accuracy for the resulting dimensionalities. The other settings are the same as in 1-NN classification.

Again, BWDR and WBDR are dominant for most of the datasets and there is sometimes large improvement over the other methods, e.g., for dataset 12 WBDR has a measure of 0.95 while the highest measure of any of the other methods except BWDR is only 0.79.

E. RESULTS ON YALEB AND MNIST DATASETS

We also conducted 1-NN classification tests on the two popular image datasets YaleB and MNIST; the settings are the same as in 1-NN classification for UCI datasets. The results are shown in Fig. 5. For these datasets, it is interesting to see that the proposed methods have the best performance comparatively to the rest of the methods when the features are reduced to low dimensionalities. Comparing these two methods, for YaleB, WBDR is better than BWDR, while for MNIST BWDR becomes better than WBDR. This again implies that the performance of these two methods on different types of data can vary significantly. Also, BWDR and WBDR are relatively more stable compared to SSDR across datasets. In fact, SSDR performs well on YaleB but has the worst performance on MNIST, while the two proposed methods achieve accuracy of more than 0.8 on both datasets. From the results on MNIST, we can see that as the dimensionality becomes larger, PCA can capture most of the important information in the original features and consequently it has the best performance; the two proposed methods have performance that is close to PCA, whereas the other two methods, especially SSDR, are worse. The aforementioned observations could be explained by the fact that PCA uses the total difference of data, while the other methods consider the between-class difference and the within-class difference separately, and that the two proposed methods can balance the effects of between-class difference and within-class difference better.

TABLE 2. Accuracy results of 1-NN classification on UCI datasets after dimensionality reduction.

Dataset No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
PCA	0.93	0.80	0.88	0.75	0.60	**0.97**	0.81	0.70	0.87	**0.90**	0.90	**0.95**	0.75	0.94	**0.97**	0.86
cFLD	0.40	–	0.87	0.52	0.53	0.58	0.71	0.27	0.57	0.72	0.23	0.92	0.42	0.38	0.54	0.59
SSDR	0.88	0.93	0.92	0.78	**0.61**	0.54	0.67	0.78	0.91	0.84	0.90	0.52	0.43	0.94	0.82	0.25
BWDR	**0.94**	**0.94**	**0.93**	0.76	**0.61**	**0.97**	**0.85**	**0.81**	0.96	0.88	0.91	**0.95**	0.77	**0.97**	**0.97**	**0.89**
WBDR	**0.94**	0.87	**0.93**	**0.79**	0.60	0.65	0.82	0.75	**0.97**	0.84	**0.98**	**0.95**	**0.79**	**0.97**	0.96	0.87

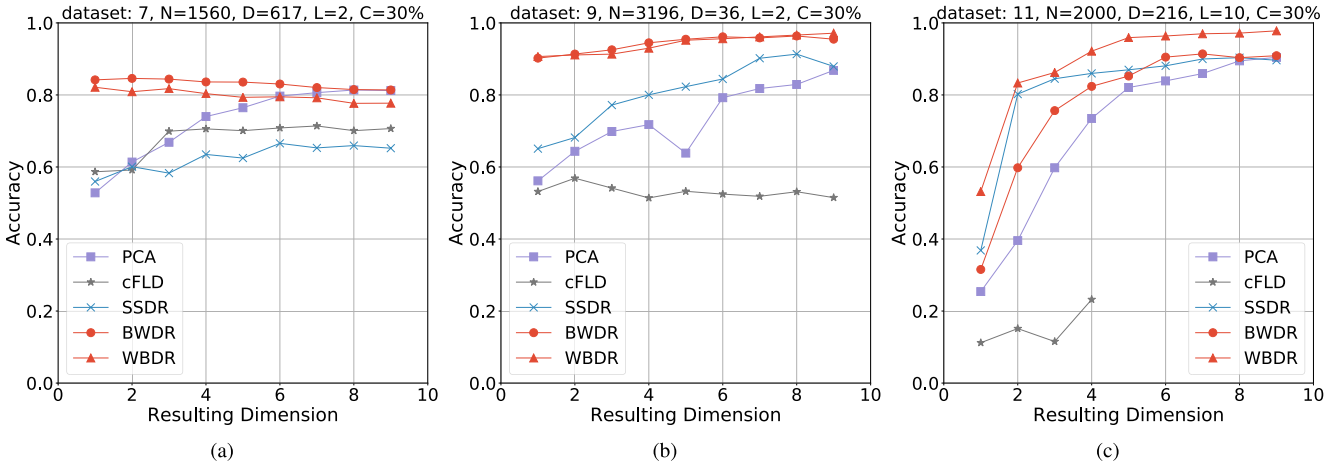


FIGURE 4. Accuracy results of 1-NN classification on three of the tested UCI datasets. (a) Dataset 7. (b) Dataset 9. (c) Dataset 11.

TABLE 3. Accuracy results of clustering on UCI datasets after dimensionality reduction.

Dataset No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
PCA	0.86	0.65	0.51	0.53	0.52	**0.50**	0.51	0.69	0.50	0.49	0.74	0.78	0.67	0.85	0.70	0.70
cFLD	**0.90**	–	0.52	0.71	0.52	**0.50**	0.70	0.80	**0.89**	**0.61**	0.68	0.91	0.81	0.86	0.70	**0.85**
SSDR	0.63	**0.87**	**0.53**	0.63	0.52	**0.50**	0.53	0.69	0.54	0.58	0.79	0.66	0.57	0.81	0.69	0.53
BWDR	**0.90**	**0.87**	**0.53**	0.58	0.52	**0.50**	**0.78**	**0.85**	0.88	0.55	0.86	0.90	0.75	0.82	0.75	**0.85**
WBDR	0.88	0.79	0.52	**0.73**	**0.53**	**0.50**	0.72	0.83	**0.89**	0.60	**0.95**	**0.93**	**0.86**	**0.95**	**0.86**	0.84

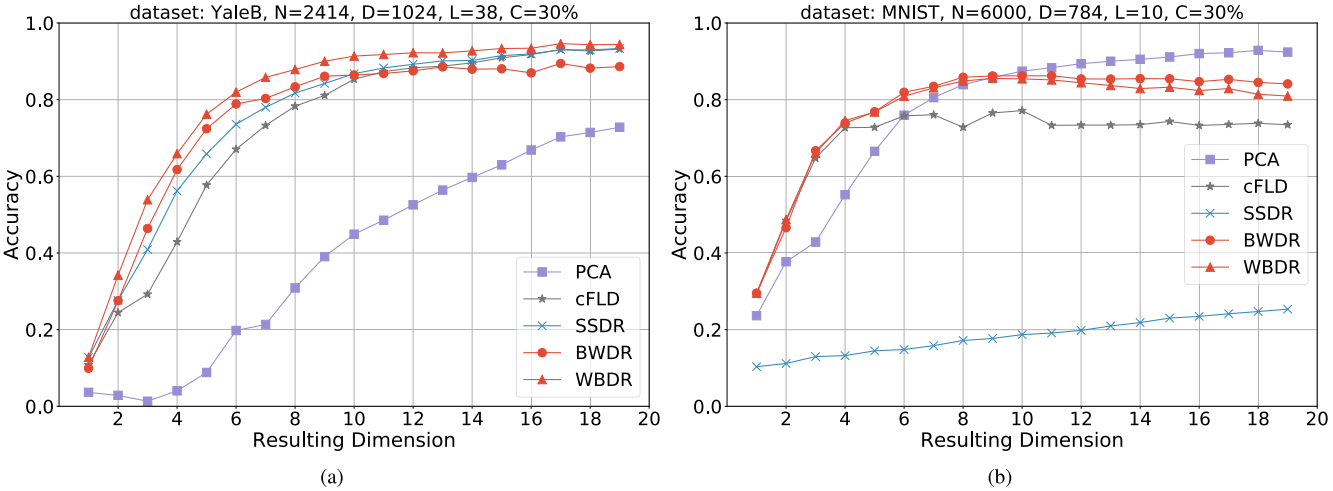


FIGURE 5. Accuracy results of 1-NN classification on YaleB and MNIST. (a) Dataset YaleB. (b) Dataset MNIST.

We also measured cumulative purity [16] for dimensionality reduction on these two datasets. Cumulative purity measures the correctness of neighbours after projecting data into lower dimensional space. The purity here is defined as the percentage of *correct neighbours* (neighbours with the same label as the current data point) for each data point averaged

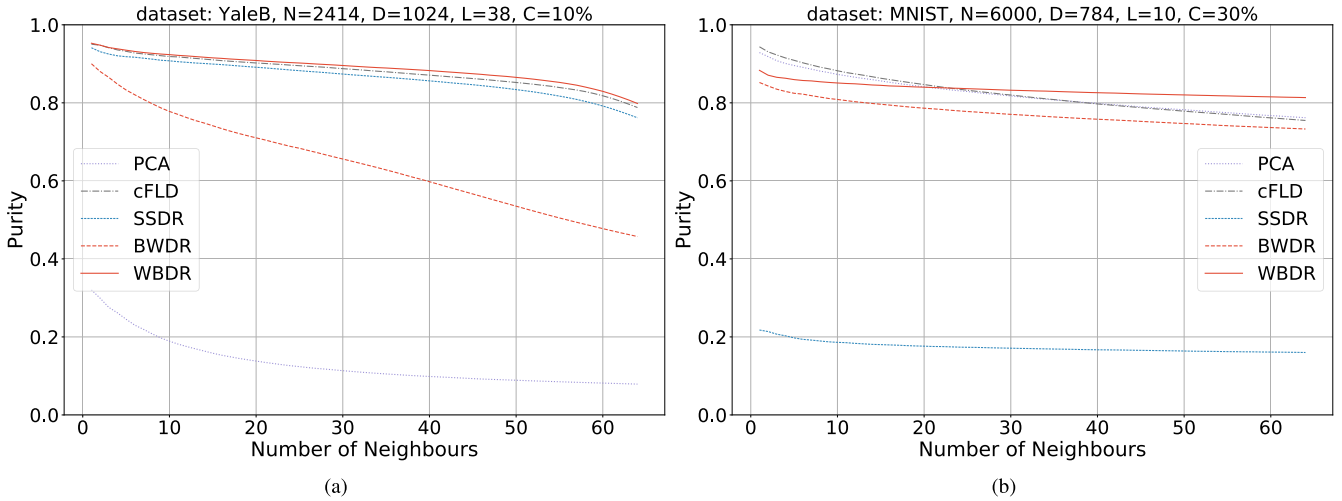


FIGURE 6. Purity graph on YaleB and MNIST. (a) Dataset YaleB. (b) Dataset MNIST.

over the whole data. The target dimensionality is set to 15, and all of the data are used to obtain a projection matrix. The results are shown in Fig. 6. For YaleB, WBDR has always higher purity than the others. For MNIST, at first WBDR has lower purity but, as the number of neighbours increases, WBDR becomes better than all the other methods. BWDR has low purity for both datasets, but it is far from being the worst one. These results indicate that WBDR and BWDR can maintain correct neighbours for each data points, which can therefore improve the results of both classification and clustering.

V. CONCLUSION

In this paper, we proposed two dual methods, called BWDR and WBDR, for semi-supervised dimensionality reduction. These two methods make use of the idea of first transforming the data to make the between-class or the within-class difference (almost) constant on any directions. In that way, the bi-objective optimization problem of finding directions with largest between-class differences and smallest within-class differences is reduced to a single-objective optimization problem, and good projection directions for dimensionality can be more easily obtained as a consequence. Experiments show that these two methods work very well on several standard datasets, improving classification and clustering performance over several existing methods. In the future, we would like to explore non-linear versions of the proposed methods, e.g., based on kernel trick [32]. Also, investigating how to overcome inconsistent pairwise constraints or to deal with noisy pairwise information [33] under the current framework is another interesting direction for future work.

REFERENCES

[1] L. M. Abualigah, A. T. Khader, M. A. Al-Betar, and O. A. Alomari, "Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering," *Expert Syst. Appl.*, vol. 84, pp. 24–36, Oct. 2017.

[2] Y. Jia, Y. Zheng, L. Gu, A. Subpa-Asa, A. Lam, Y. Sato, and I. Sato, "From RGB to spectrum for natural scenes via manifold-based mapping," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4715–4723.

[3] S. Gong, V. N. Boddeti, and A. K. Jain, "On the intrinsic dimensionality of image representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3987–3996.

[4] M. He and S. Petoukhov, *Mathematics of Bioinformatics: Theory, Methods and Applications*. New York, NY, USA: Wiley, 2011.

[5] J. Lee, S. Ciccarello, M. Acharjee, and K. Das, "Dimension reduction of gene expression data," *J. Stat. Theory Pract.*, vol. 12, no. 2, pp. 450–461, Apr. 2018.

[6] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *Proc. Int. Conf. Database Theory (ICDT)*, 2001, pp. 420–434.

[7] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 1986.

[8] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[9] X. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, Wisconsin, Tech. Rep. TR1530, 2005. [Online]. Available: http://digital.library.wisc.edu/1793/60444

[10] S. Yu, K. Yu, V. Tresp, H.-P. Kriegel, and M. Wu, "Supervised probabilistic principal component analysis," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2006, pp. 464–473.

[11] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Roy. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 61, no. 3, pp. 611–622, 1999.

[12] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–7.

[13] M. Sugiyama, T. Idé, S. Nakajima, and J. Sese, "Semi-supervised local Fisher discriminant analysis for dimensionality reduction," *Mach. Learn.*, vol. 78, nos. 1–2, pp. 35–61, Jan. 2010.

[14] W. Liao, A. Pizurica, P. Scheunders, W. Philips, and Y. Pi, "Semisupervised local discriminant analysis for feature extraction in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 184–198, Jan. 2013.

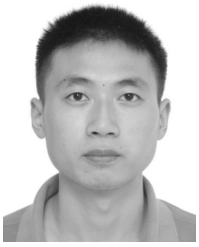
[15] S. Wang, J. Lu, X. Gu, H. Du, and J. Yang, "Semi-supervised linear discriminant analysis for dimension reduction and classification," *Pattern Recognit.*, vol. 57, pp. 179–189, Sep. 2016.

[16] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a Mahalanobis metric from equivalence constraints," *J. Mach. Learn. Res.*, vol. 6, no. 1, pp. 937–965, 2006.

[17] D. Zhang, Z. Zhou, and S. Chen, "Semi-supervised dimensionality reduction," in *Proc. SIAM Int. Conf. Data Mining*, 2007, pp. 629–634.

[18] H. Cevikalp, J. Verbeek, F. Jurie, and A. Klaser, "Semi-supervised dimensionality reduction using pairwise equivalence constraints," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2008, pp. 489–496.

[19] J. Wei and H. Peng, "Neighbourhood preserving based semi-supervised dimensionality reduction," *Electron. Lett.*, vol. 44, no. 20, pp. 1190–1191, 2008.
[20] R. Chatpatanasiri and B. Kijsirikul, "A unified semi-supervised dimensionality reduction framework for manifold learning," *Neurocomputing*, vol. 73, nos. 10–12, pp. 1631–1640, Jun. 2010.
[21] S. Yan, S. Bouaziz, D. Lee, and J. Barlow, "Semi-supervised dimensionality reduction for analyzing high-dimensional data with constraints," *Neurocomputing*, vol. 76, no. 1, pp. 114–124, Jan. 2012.
[22] R. Memisevic and G. E. Hinton, "Multiple relational embedding," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2004, pp. 913–920.
[23] J. Costa and A. Hero, "Classification constrained dimensionality reduction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Oct. 2006, pp. 1077–1080.
[24] X. Yang, H. Fu, H. Zha, and J. Barlow, "Semi-supervised nonlinear dimensionality reduction," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 1065–1072.
[25] H. Wu and S. Prasad, "Semi-supervised dimensionality reduction of hyperspectral imagery using pseudo-labels," *Pattern Recognit.*, vol. 74, pp. 212–224, Feb. 2018.
[26] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Dec. 1994.
[27] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, "Data transformations," in *Data Mining*, 4th ed, I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Eds. San Mateo, CA, USA: Morgan Kaufmann, 2017, pp. 285–334.
[28] D. Dua and C. Graff. *UCI Machine Learning Repository*. Accessed: 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[29] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.
[30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[31] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 521–528.
[32] S. Theodoridis and K. Koutroumbas, Eds., *Pattern Recognition*. New York, NY, USA: Academic, 2009.
[33] Z. Lu and L. Wang, "Noise-robust semi-supervised learning via fast sparse coding," *Pattern Recognit.*, vol. 48, no. 2, pp. 605–612, Feb. 2015.

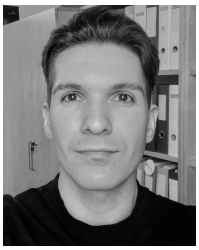


ZHIGUO LONG received the B.S. degree in mathematics from Sichuan University, China, in 2012, and the Ph.D. degree from the University of Technology Sydney, Australia, in 2017.

Since 2017, he has been a Lecture with the School of Information Science and Technology, Southwest Jiaotong University. His research interests include fundamental and practical techniques in knowledge representation and reasoning, especially qualitative spatial and temporal reasoning, and representation problems in machine learning and computer vision.

HUA MENG received the B.S. and Ph.D. degrees in mathematics from Sichuan University, China, in 2005 and 2010, respectively. He was a Visiting Scholar with the University of Technology Sydney, Australia, in 2014.

He currently works with the School of Mathematics, Southwest Jiaotong University. His research interests include knowledge representation and reasoning, machine learning, and general topology.



MICHAEL SIOUTIS received the Ph.D. degree from Artois University, France, in February 2017. He was a Postdoctoral Researcher with Örebro University, Sweden, from May 2017 to December 2018, and a Postdoctoral Researcher with Aalto University, in 2019.

Since January 2020, he has been a Research Fellow with the University of Bamberg, Germany, in the area of computer science. His general interests include artificial intelligence, data mining, Linux system engineering, logic programming, and semantic Web.