Semi-Supervised Dimensionality Reduction*

Daoqiang Zhang[1,2] Zhi-Hua Zhou[1] Songcan Chen[2]

[1]National Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
[2]Department of Computer Science and Engineering
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
dqzhang@nuaa.edu.cn zhouzh@nju.edu.cn s.chen@nuaa.edu.cn

Abstract

Dimensionality reduction is among the keys in mining high-dimensional data. This paper studies semi-supervised dimensionality reduction. In this setting, besides abundant unlabeled examples, domain knowledge in the form of pairwise constraints are available, which specifies whether a pair of instances belong to the same class (*must-link* constraints) or different classes (*cannot-link* constraints). We propose the SSDR algorithm, which can preserve the intrinsic structure of the unlabeled data as well as both the must-link and cannot-link constraints defined on the labeled examples in the projected low-dimensional space. The SSDR algorithm is efficient and has a closed form solution. Experiments on a broad range of data sets show that SSDR is superior to many established dimensionality reduction methods.

1 Introduction

With the rapid accumulation of high-dimensional data such as digital images, financial time series and gene expression microarrays, dimensionality reduction has been a fundamental tool for many data mining tasks. According to whether supervised information is available or not, existing dimensionality reduction methods can be roughly categorized into supervised ones and unsupervised ones. Fisher Linear Discriminant (FLD) [7] is an example of supervised dimensionality reduction methods, which can extract the optimal discriminant vectors when class labels are available; while Principal Component Analysis (PCA) [11] is an example of unsupervised dimensionality reduction methods, which works through trying to preserve the global covariance structure of data when class labels are not available.

Semi-supervised dimensionality reduction can be seen as a new issue in semi-supervised learning, which learns from a combination of both labeled and unlabeled data. In many practical data mining applications, unlabeled training examples are readily available but labeled ones are fairly expensive to obtain, therefore semi-supervised learning has attracted much attention. Current research on semi-supervised learning could be roughly categorized into three classes, i.e. semi-supervised classification [5, 19], semi-supervised regression [6, 20], and semi-supervised clustering [2, 16]. Research advances of semi-supervised learning can be found in an excellent recent survey [21].

Utilizing domain knowledge has been an important issue in many data mining tasks [1, 16, 17]. In general, domain knowledge can be expressed in diverse forms, such as class labels, pairwise constraints or other prior information. We focus on domain knowledge in the form of pairwise constraints, i.e. pairs of instances known as belonging to the same class (*must-link* constraints) or different classes (*cannot-link* constraints). Pairwise constraints arise naturally in many tasks such as image retrieval. In those applications, considering the pairwise constraints is more practical than trying to obtain class labels, because the true labels may not be known *a priori*, while it could be easier for a user to specify whether some pairs of instances belong to the same class or not. Moreover, the pairwise constraints can be derived from labeled data but not vice versa. Furthermore, unlike class labels, the pairwise constraints can sometimes be automatically obtained without human intervention [1].

Several recent works have attempted to exploit pairwise constraints or other prior information in dimensionality reduction. Bar-Hillel et al. [1] proposed the constrained FLD (cFLD) for dimensionality reduction from equivalence constraints, as a interim-step for Relevant Component Analysis (RCA). However, cFLD can only deal with the must-link constraints. Also, as in FLD, cFLD has the singular problem when constraints are limited. Tang and Zhong [15] used pairwise constraints to guide dimensionality reduction, which can exploit both must-link constraints and cannot-link constraints but does not consider the usefulness of abundant unlabeled data. Yang et al. [18] exploited prior information in the form of on-manifold coordinates of certain data sam-

*Supported by NSFC (60473035, 60505004, 60635030), JiangsuSF (BK2004001, BK2006521) and FANEDD (200343).

ples for dimensionality reduction. It is evident that usually obtaining the pairwise constraints is much easier than obtaining the on-manifold coordinates of data samples.

In this paper, we study the dimensionality reduction problem where both unlabeled data and pairwise constraints are available. We propose a simple but efficient algorithm called SSDR (Semi-Supervised Dimensionality Reduction), which can simultaneously preserve the structure of original high-dimensional data and the pairwise constraints specified by users. Moreover, SSDR has a closed solution of an eigen-problem of some specific Laplacian matrix [10, 19] and therefore it is quite efficient. In the following we start by presenting SSDR and then reporting on the experiments.

2 SSDR

Here we formulate semi-supervised dimensionality reduction as follows: Given a set of data samples $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ together with some pairwise must-link constraints (M) and cannot-link constraints (C), find a set of projective vectors $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_d]$, such that the transformed low-dimensional representations $\boldsymbol{y}_i = \boldsymbol{W}^T\boldsymbol{x}_i$ can preserve the structure of the original data set as well as the pairwise constraints M and C, i.e. instances involved by M should be close while instances involved by C should be far in the low-dimensional space.

Define the objective function as maximizing $J(\boldsymbol{w})$

$$\begin{aligned} J(\boldsymbol{w}) &= \frac{1}{2n_C}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in C}(y_i - y_j)^2 \\ &\quad -\frac{\beta}{2n_M}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in M}(y_i - y_j)^2 \\ &= \frac{1}{2n_C}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in C}(\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{x}_j)^2 \\ &\quad -\frac{\beta}{2n_M}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in M}(\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{x}_j)^2 \end{aligned} \tag{2.1}$$

w.r.t. $\boldsymbol{w}^T\boldsymbol{w} = 1$. Here $\boldsymbol{y}_i = \boldsymbol{w}^T\boldsymbol{x}_i$ is the transformed low-dimensional representation of $\boldsymbol{x}_i$. For the convenience of discussion, one-dimensional case is considered here but it is not difficult to extend to high-dimensions. n_C and n_M are the number of cannot-link and must-link constraints, respectively.

The intuition behind Eq.2.1 is to let the average distance in the transformed low-dimensional space between instances involved by the cannot-link set C as large as possible, while distances between instances involved by the must-link set M as small as possible. Since the distance between instances in the same class is typically smaller than that in different classes, we add a scaling parameter β to balance the contributions of the two terms in Eq.2.1.

Eq.2.1 considers only the constraints. When there are abundant unlabeled examples, Eq.2.1 should be extended such that both the constraints and the unlabeled data are considered. Here 'unlabeled' means the data has neither class labels nor pairwise constraints involvements. The extended objective function is defined as maximizing $J(\boldsymbol{w})$ w.r.t. $\boldsymbol{w}^T\boldsymbol{w} = 1$, where

$$\begin{aligned} J(\boldsymbol{w}) &= \frac{1}{2n^2}\sum_{i,j}(\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{x}_j)^2 \\ &\quad +\frac{\alpha}{2n_C}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in C}(\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{x}_j)^2 \\ &\quad -\frac{\beta}{2n_M}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in M}(\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{x}_j)^2 \end{aligned} \tag{2.2}$$

The first term of Eq.2.2 expresses the average squared distance between all data samples in the transformed space, which is equivalent to the PCA criterion. The motivation for exploiting unlabeled data is to use them to enhance performance when constraints are few. Since the contribution of abundant unlabeled data is included, it is expected to be more stable than using only the constraints. As in Eq.2.1, we add another scaling parameter α to balance the contribution of the cannot-linked constraints. Intuitively, distance of samples involved in the cannot-link set C should typically be near to the expected distance, so we empirically set $\alpha = 1$ and $\beta > 1$.

Eq.2.2 is evidently more general than Eq.2.1, and when both α and β take big values, the constraints will dominate the equation, then Eq.2.2 will degrade to Eq.2.1.

There exists a concise form for Eq.2.2:

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{i,j}(\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{x}_j)^2\boldsymbol{S}_{ij} \tag{2.3}$$

where

$$\boldsymbol{S}_{ij} = \begin{cases} \frac{1}{n^2} + \frac{\alpha}{n_C} & \text{if } (x_i, x_j) \in C \\ \frac{1}{n^2} - \frac{\beta}{n_M} & \text{if } (x_i, x_j) \in M \\ \frac{1}{n^2} & \text{otherwise} \end{cases} \tag{2.4}$$

From Eq.2.3, we have

$$\begin{aligned} & \frac{1}{2}\sum_{i,j}(\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{x}_j)^2\boldsymbol{S}_{ij} \\ = & \frac{1}{2}\sum_{i,j}(\boldsymbol{w}^T\boldsymbol{x}_i\boldsymbol{x}_i^T\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{x}_j\boldsymbol{x}_j^T\boldsymbol{w} - 2\boldsymbol{w}^T\boldsymbol{x}_i\boldsymbol{x}_j^T\boldsymbol{w})\boldsymbol{S}_{ij} \\ = & \sum_{i,j}\boldsymbol{w}^T\boldsymbol{x}_i\boldsymbol{S}_{ij}\boldsymbol{x}_i^T\boldsymbol{w} - \sum_{i,j}\boldsymbol{w}^T\boldsymbol{x}_i\boldsymbol{S}_{ij}\boldsymbol{x}_j^T\boldsymbol{w} \\ = & \sum_{i}\boldsymbol{w}^T\boldsymbol{x}_i\boldsymbol{D}_{ii}\boldsymbol{x}_i^T\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{X}\boldsymbol{S}\boldsymbol{X}^T\boldsymbol{w} \\ = & \boldsymbol{w}^T\boldsymbol{X}(\boldsymbol{D} - \boldsymbol{S})\boldsymbol{X}^T\boldsymbol{w} \\ = & \boldsymbol{w}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{w} \end{aligned}$$

Here $\boldsymbol{D}$ is a diagonal matrix whose entries are column (or row) sums of $\boldsymbol{S}$, i.e. $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{S}_{ij}$. $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$ is called the Laplacian matrix in spectral graph theory. Thus, Eq.2.2 or Eq.2.3 can be simplified as maximizing $J(\boldsymbol{w})$ w.r.t. $\boldsymbol{w}^T\boldsymbol{w} = 1$, where

$$J(\boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T \boldsymbol{w} \tag{2.5}$$

Clearly, the problem expressed by Eq.2.5 is a typical eigen-problem, which can be easily and efficiently solved by computing the eigenvectors of $\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T$ corresponding to the largest eigenvalues.

We have some remarks on the SSDR algorithm:

- SSDR can have different implementations which are determined by the weights $\boldsymbol{S}$ it used. In this paper, we focus on three ways to construct the weights and denote the corresponding algorithms as:

  (1) SSDR-M: Using only the must-link constraints, with

$$\boldsymbol{S}_{ij} = \begin{cases} -\frac{\beta}{n_M} & \text{if } (x_i, x_j) \in M \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

  (2) SSDR-CM: Using both the cannot-link and must-link constraints, with

$$\boldsymbol{S}_{ij} = \begin{cases} \frac{\alpha}{n_C} & \text{if } (x_i, x_j) \in C \\ -\frac{\beta}{n_M} & \text{if } (x_i, x_j) \in M \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

  (3) SSDR-CMU: Using both the cannot-link and must-link constraints together with unlabeled data, with the weights $\boldsymbol{S}$ defined in Eq.2.4.

- Although the form of Eq.2.5 is similar as that of the Laplacian spectral embedding methods [3, 10, 13], actually they are completely different. First, their purposes are different. Laplacian spectral embedding methods originate from unsupervised dimensionality reduction with locality preserving, while SSDR is proposed for semi-supervised dimensionality reduction with constraints preserving. Moreover, their adjacent graph and weights are constructed in different ways. Laplacian spectral embedding methods usually employ the k-nearest neighborhood or ε-neighborhood and it is hard to select the appropriate value for k or ε, while SSDR is based on the constraints and its weights are directly determined by Eq.2.4 (or Eq.2.6 and Eq.2.7).

3 Experiments

In this section, we evaluate the performance of the SSDR algorithms on a broad range of data sets, including six UCI data sets [4], i.e. *balance*, *ionosphere*, *iris*, *sonar*, *soybean* and *wine*, YaleB facial image data set [8], and three text data sets derived from 20-Newsgroup [2], i.e. News-Different-300, News-Similar-300 and News-Same-300. In our experiments, the pairwise constraints are obtained by randomly selecting pairs of instances from the training set (for UCI data) or the whole data set (for YaleB and 20-Newsgroups), and creating must-link or cannot-link constraints depending on whether the underlying classes of the two instances are the same or different. After obtaining the constraints, data without constraints in the training set (for UCI data) or the whole data set (for YaleB and 20-Newsgroup) are used as unlabeled data. Different levels of constraints are generated relative to the number of total data samples. In all cases, results are averaged over 100 runs with different generation of constraints. The parameters in SSDR are always set to $\alpha = 1$ and $\beta = 20$ if without extra explanations.

3.1 Results on UCI Data Sets In this section we assess the relative performance of SSDR over other dimensionality reduction methods for classification. We choose the fully unsupervised PCA as the baseline. We also test the performance of supervised FLD which uses the ground-truth class labels of all the training data. We compare SSDR (including SSDR-M, SSDR-CM and SSDR-CMU) with cFLD under different level of constraints. After dimensionality reduction, nearest neighborhood (1-NN) classifier is employed for classification. For each data set, we use the first half of the data for training (learning the projections) and the remaining data for testing.

Figures 1 shows that SSDR-CMU nearly always achieves the highest accuracy on all data sets. In particular, when the number of constraints are limited, SSDR-CMU outperforms other algorithms significantly. It can also be shown from Figure 1 that in most cases the performance of PCA is the worst. We believe that this is because PCA does not use the constraints. On the other hand, the poor performance of SSDR-M implies that only using the must-link constraints is not sufficient. Actually, by exploiting the cannot-constraints, SSDR-CM significantly improves the performance of SSDR-M. Both SSDR-CM and SSDR-CMU are superior to cFLD. It is amazing that the performance of SSDR-CMU is even better than that of the supervised FLD.

To see how the dimensionality of the projected space affects the accuracy, we compare the classification accuracies on 3 UCI data sets with different number of dimensions, as shown in Figure 2. It is impressive that SSDR-CMU almost always achieves the highest accuracy no matter under which dimensionality. Moreover, it seems that SSDR-CMU is little affected by the dimensionality and is stable for a wide range of dimensions, while other methods typically require relative more dimensions to obtain a good accuracy. This is an advantageous of SSDR-CMU since in most cases, working in lower-dimensional space is much easier than in a higher-dimensional space.

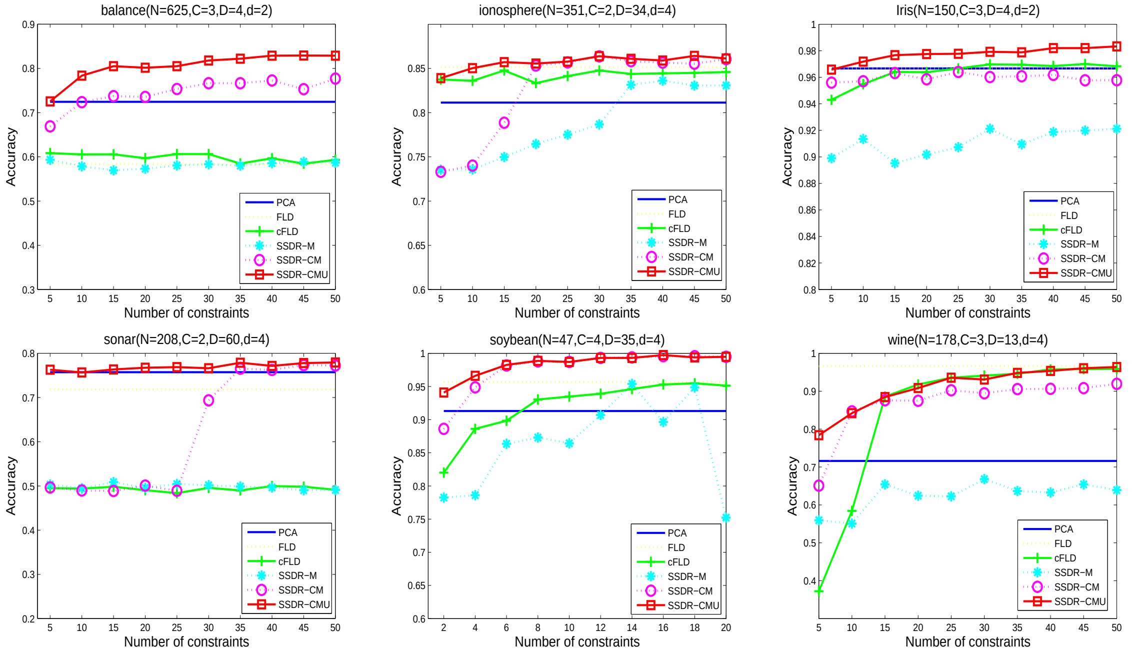

Figure 1: Classification accuracy on 6 UCI data sets with different number of constraints. Also shown in each panel are N-the number of data examples, C-the number of classes, D-the dimension of original data and d-the reduced dimension in projected space. Here, FLD use C-1 discriminant vectors, while other algorithms use d projective vectors.

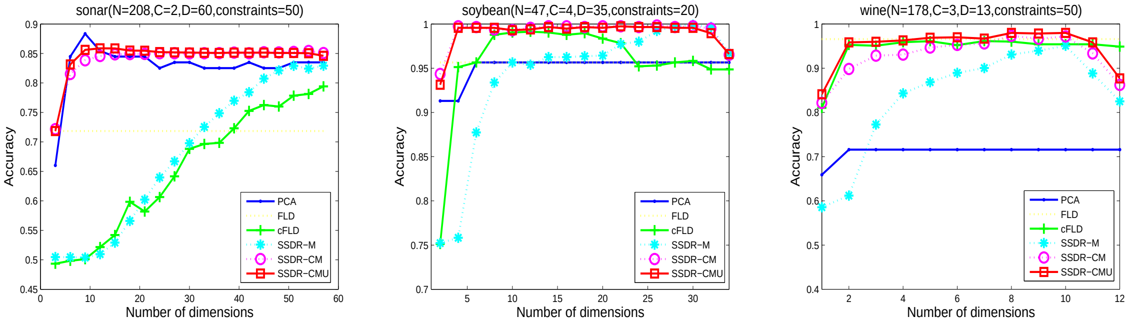

Figure 2: Classification accuracy on 3 UCI data sets with different number of dimensions.

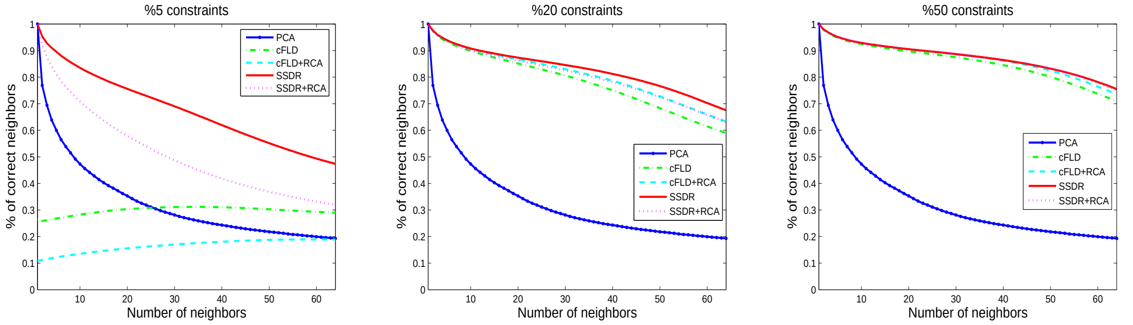

Figure 3: Cumulative purity graph on YaleB data set with different percentiles of constraints.

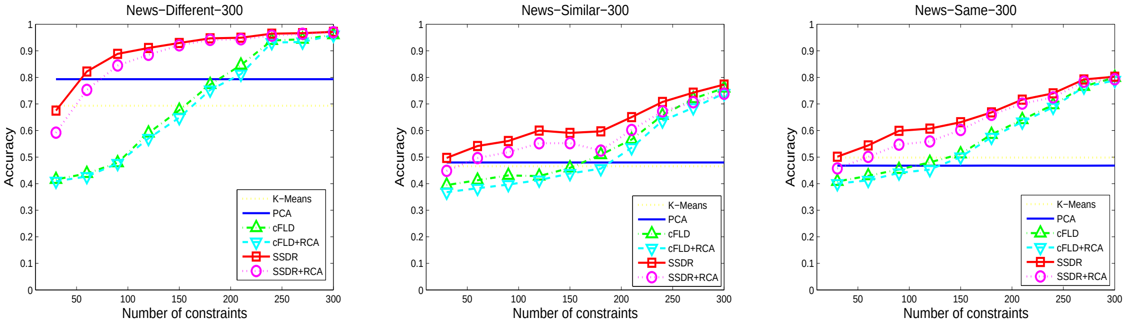

Figure 4: Clustering accuracy on 3 newsgroups data sets with different number of constraints.

As shown in the experiments on UCI data sets, SSDR-CMU nearly always outperforms SSDR-M and SSDR-CM. Thus in the rest of this paper, we only consider SSDR-CMU, denoted as SSDR for short.

3.2 Results on YaleB Face Image Data Set In this section, we investigate the performance of SSDR for face recognition. Figure 3 displays the cumulative purity [1] graph on YaleB data set with different levels of constraints. Here the cumulative purity graph denotes the average (over all data points) percentage of correct neighbors among the first k neighbors, as a function of k, and the level of constraints is relative to the total number of face images. It can be seen that SSDR always outperforms other methods no matter how many constraints are used. Figure 3 also reveals that as the number of constraints increases, the performance of cFLD becomes close to that of SSDR, and both significantly better than PCA. However, when there are only a few constraints, the performance of cFLD degrades severely. We also compare the performances of SSDR and cFLD when combining with RCA. Although RCA can help improve the performance of cFLD, it brings little improvement on SSDR.

3.3 Results on 20-Newsgroups Data Set This section reports on the experiments on three text data sets including News-Different-300, News-Similar-300 and News-Same-300, derived from 20-Newsgroup.

Figure 4 shows the clustering accuracies of k-means in the original 100-dimensional space, PCA, cFLD (with and without RCA) and SSDR (with and without RCA) on reduced 3-dimensional space. At each test k-means is applied 10 times with different starting points and the best result in term of the objective function of k-means is recorded. Here *clustering accuracy* is defined as

$$ClusAcc = \frac{1}{n}\sum_{i=1}^{n}\delta(s_i, map(r_i))$$

where r_i and s_i are respectively the obtained cluster label and the ground-truth label of instance $\boldsymbol{x}_i$, n is the number of instances, $\delta(x, y)$ equals one if $x = y$ and zero otherwise, and $map(r_i)$ is a permutation mapping function that maps each cluster label r_i to the equivalent label from the data set. As can be seen from Figure 4, for both cFLD and SSDR, increasing the number of constraints leads to the improvement on the performance. When there are enough number of constraints, the accuracy of cFLD is near to that of SSDR and both considerably outperform k-means and PCA. However, when there are only limited number of constraints, SSDR is significantly superior to cFLD. It can also be seen that on these data sets, RCA seems help little on improving the accuracies of cFLD as well as SSDR.

Table 1: F-scores on Newsgroups data with 10% constraints

Methods	Different-300	Similar-300	same-300
Shental[14]	0.554	0.553	0.429
Basu[2]	0.582	0.492	0.459
Lange[12]	0.658	0.540	0.588
SSDR	0.692	0.504	0.492
SSDR(best)	0.878	0.557	0.546

Table 2: F-scores on Newsgroups data with 30% constraints

Methods	Different-300	Similar-300	same-300
Shental[14]	0.871	0.532	0.487
Basu[2]	0.608	0.530	0.552
Lange[12]	0.594	0.514	0.507
SSDR	0.836	0.542	0.532
SSDR(best)	0.923	0.606	0.615

We also compare SSDR with some recent algorithms which also use pairwise constraints on the same data sets. Following [12], we first use SSDR to project the original data to a 20-dimensional space and then perform k-means on the reduced space. Tables 1 and 2 give the F-scores (i.e. the harmonic mean of precision and recall) [12] with 10% and 30% constraints, respectively. Here for SSDR, we perform 100 constraints realizations for each case, and both the average (denoted as SSDR in tables) and the best results are recorded. As can be seen, SSDR is very competitive with other algorithms. To make it more clearly, we compute the total rank of the five algorithms on all six cases. For example, on News-Different-300 with 10% constraints, the rank of the five algorithms is: SSDR (best) (1) $>$ SSDR (2) $>$ Lange et al. (3) $>$ Basu et al. (4) $>$ Shental et al. (5). The total rank on these three data sets is: SSDR (best) (7) $>$ SSDR (17) $>$ Lange et al. (21) $>$ Shental et al. (22) $>$ Basu et al. (23). Here the values in the brackets show the ranks.

3.4 Runtime Performance Computationally, SSDR is a standard eigen-problem on a symmetric matrix, which can be efficiently computed, e.g. by the singular value decomposition (SVD). For large sparse high-dimensional data such as text data, there exists efficient sparse SVD algorithms [9] which can also be used by SSDR.

For dimensionality reduction, we empirically find that the efficiency of SSDR is nearly the same as that of PCA, both being a bit superior to that of cFLD. For clustering, since SSDR considers the constraints only once for dimensionality reduction and then performs k-means, it is much efficient than algorithms which use constraints in the process of k-means clustering, e.g. Basu et al's algorithm where the constraints are considered in each iterative step of k-means. This is a clear advantage of SSDR over its competitors.

4 Conclusion

In this paper, we propose a simple but efficient semi-supervised dimensionality reduction algorithm called SSDR, which exploits both cannot-link and must-link constraints together with unlabeled data. SSDR can preserve the intrinsic structure of the data set as well as the pairwise constraints specified by users in the projected low-dimensional space. Experiments show that SSDR leads to considerable improvements in embedding, classification and clustering over conventional dimensionality reduction methods.

In this paper, the intrinsic structure preserved by SSDR is the global covariance structure. Investigating that whether SSDR can preserve local structures together with constraints is an interesting future work. In our experiments the pairwise constraints are randomly generated and have no contradiction, investigating the performance of SSDR with inconsistent constraints is also an interesting future work.

References

[1] A. BAR-HILLEL, T. HERTZ, N. SHENTAL, AND D. WEINSHALL, *Learning a mahalanobis metric from equivalence constraints*, Journal of Machine Learning Research, 6 (2005), pp. 937–965.

[2] S. BASU, M. BILENKO, AND R. MOONEY, *A probabilistic framework for semi-supervised clustering*, in KDD'04, Seattle, WA, 2004, pp. 59–68.

[3] M. BELKIN AND P. NIYOGI, *Laplacian eigenmaps and spectral techniques for embedding and clustering*, in NIPS 14, MIT Press, Cambridge, MA, 2002.

[4] C. BLAKE, E. KEOGH, AND C. J. MERZ, *UCI repository of machine learning databases*. [http://www.ics.uci.edu/∼mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, 1998.

[5] A. BLUM AND T. MITCHELL, *Combining labeled and unlabeled data with co-training*, in COLT'98, Madison, WI, 1998, pp. 92–100.

[6] U. BREFELD, T. GÄRTNER, T. SCHEFFER, AND S. WROBEL, *Efficient co-regularised least squares regression*, in ICML'06, Pittsburgh, PA, 2006, pp. 137–144.

[7] R. A. FISHER, *The use of multiple measurements in taxonomic problems*, Annals of Eugenics, 7 (1936), pp. 179–188.

[8] A. S. GEORGHIADES, P. N. BELHUMEUR, AND D. J. KRIEGMAN, *From few to many: Illumination cone models for face recognition under variable lighting and pose*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23 (2001), pp. 643–660.

[9] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 3rd ed., 1996.

[10] X. HE AND P. NIYOGI, *Locality preserving projections*, in NIPS 16, MIT Press, Cambridge, MA, 2004.

[11] I. JOLIFFE, *Principal Component Analysis*, Springer, New York, NY, 1986.

[12] T. LANGE, M. H. LAW, A. K. JAIN, AND J. BUHMANN, *Learning with constrained and unlabeled data*, in CVPR'05, San Diego, CA, 2005, pp. 731–738.

[13] L. K. SAUL, K. Q. WEINBERGER, J. H. HAM, F. SHA, AND D. D. LEE, *Spectral methods for dimensionality reduction*, MIT Press, Cambridge, MA, 2005.

[14] N. SHENTAL, A. BAR-HILLEL, T. HERTZ, AND D. WEINSHALL, *Computing gaussian mixture models with em using equivalence constraints*, in NIPS 16, MIT Press, Cambridge, MA, 2004.

[15] W. TANG AND S. ZHONG, *Pairwise constraints-guided dimensinality reduction*, in SDM'06 Workshop on Feature Selection for Data Mining, Bethesda, MD, 2006.

[16] K. WAGSTAFF, C. CARDIE, S. ROGERS, AND S. SCHROEDL, *Constrained k-means clustering with background knowledge*, in ICML'01, Williamstown, MA, 2001, pp. 577–584.

[17] E. P. XING, A. Y. NG, M. I. JORDAN, AND S. RUSSELL, *Distance metric learning, with application to clustering with side-information*, in NIPS 15, MIT Press, Cambridge, MA, 2003, pp. 505–512.

[18] X. YANG, H. FU, H. ZHA, AND J. L. BARLOW, *Semi-supervised nonlinear dimensionality reduction*, in ICML'06, Pittsburgh, PA, 2006, pp. 1065–1072.

[19] T. ZHANG AND R. K. ANDO, *Analysis of spectral kernel disign based semi-supervised learning*, in NIPS 18, MIT Press, Cambridge, MA, 2006, pp. 1601–1608.

[20] Z.-H. ZHOU AND M. LI, *Semi-supervised learning with co-training*, in IJCAI'05, Edinburgh, Scotland, 2005, pp. 908–913.

[21] X. ZHU, *Semi-supervised learning literature survey*, Tech. Report 1530, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, 2006. http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.