Pattern Recognition 43 (2010) 720–730

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Multiple view semi-supervised dimensionality reduction

Chenping Hou[a,b,*], Changshui Zhang[b], Yi Wu[a], Feiping Nie[b]

[a]Department of Mathematics and Systems Science, National University of Defense Technology, Changsha 410073, China
[b]State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 21 September 2008
Received in revised form 14 July 2009
Accepted 24 July 2009

Keywords:
Dimensionality reduction
Semi-supervised
Multiple view
Domain knowledge

ABSTRACT

Multiple view data, together with some domain knowledge in the form of pairwise constraints, arise in various data mining applications. How to learn a hidden consensus pattern in the low dimensional space is a challenging problem. In this paper, we propose a new method for multiple view semi-supervised dimensionality reduction. The pairwise constraints are used to derive embedding in each view and simultaneously, the linear transformation is introduced to make different embeddings from different pattern spaces comparable. Hence, the consensus pattern can be learned from multiple embeddings of multiple representations. We derive an iterating algorithm to solve the above problem. Some theoretical analyses and out-of-sample extensions are also provided. Promising experiments on various data sets, together with some important discussions, are also presented to demonstrate the effectiveness of the proposed algorithm.

1. Introduction

With the rapid increase of high dimensional data, such as digital images, financial time series and web texts, dimensionality reduction has been widely used as a fundamental tool for many data mining tasks. Different from traditional applications, some data may have special characters in some real cases. How to reduce their dimensionality is a big challenge. For example, in some applications, an instance may have multiple views, i.e., there are multiple representations from different feature spaces or graph spaces for an instance [1]. Simultaneously, we may also be given some domain knowledge in the form of pairwise constraints. Nevertheless, the number of constraints is limited since examples are readily available but constructing links between two points is fairly expensive [2]. One typical application is to predict a person's political view (left, right) from his/her online blogs. The dimensionality of blog page is often very high and each page has disparate descriptions: textual content, inbound and out-bound links, etc. Moreover, the fact that person *B* quotes person *A* and uses expletives near the quotation is a strong indication that *B* disagrees with *A* [3]. We may create a dissimilarity pair to reflect our knowledge that *A* and *B* probably have different political views. On the contrary, if *A* uses praises, we probably know that they have the same political view.

Together with some domain knowledge, multiple view high dimensional data raise a natural, yet non-standard new problem: how to use the multiple representations and the given domain knowledge to learn a consensus pattern, which is more accurate than the pattern based on a single view. This is the main task of multiple view semi-supervised dimensionality reduction and the consensus pattern should be shared by multiple representations as much as possible. Commonly, since the structure of each view is disparate, it is unwise to regard multiple representations as one view by simply connecting all features. This would treat the representation of each view in the same way and ignore their diversities. Previous approaches have also shown the advantages of multiple view learning [1].

Traditional well-known dimensionality reduction methods, such as principle component analysis (PCA) [4] and locally linear embedding (LLE) [5], are not suitable to reduce the dimensionality of this kind of data, since (1) they are unsupervised and they cannot incorporate the domain knowledge. If they were employed, these algorithms will suffer from a low discriminant power due to their unsupervised nature. (2) It is unsuitable to apply these approaches to directly reduce dimensionality of the data that are formed by connecting representations of each view, since each view may have very different formulations or statistical properties. For example, genes can be represented in the expression vector space (corresponding to the genetic activity) and also in the term vector space (corresponding to the text information) [6].

* Corresponding author at: Department of Mathematics and Systems Science, National University of Defense Technology, Changsha 410073, China.
Tel.: +86 731 84573260; fax: +86 731 84573265.
E-mail address: hcpnudt@gmail.com (C. Hou).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.07.015

As far as we know, there is little work dedicating to this topic in the literature. Foster et al. have employed canonical correlation analysis (CCA) approach [7] to derive the low dimensional embeddings of two-view data and compute the regression function based on these embeddings [8]. They focus on different domain knowledge (the exact label) and different problem (regression). There are also some researches for multiple view unsupervised learning and semi-supervised dimensionality reduction (SSDR). We will review them since they have close relationships with this topic.

There are two directions for seeking solutions to multiple view unsupervised learning. One is to design the centralized algorithms that use multiple solutions simultaneously to mine hidden pattern from the data. The top challenge for these approaches is the diversity of multiple representations, since it is difficult to design a single algorithm to accomplish several different tasks which are handled by separate algorithms. The existing efforts usually restrict themselves to special cases. For example, Bickel and Scheffer assumed that the predefined features are independent and can be handled by the same algorithms [9]. Zhou et al. focused on the data that can be well modeled by a mixture of Markov chains [10]. The other direction is distributed, or namely they learn hidden patterns individually from each representation and then compute the consensus hidden pattern from those multiple patterns. Under this framework, the problem of choosing the most appropriate method is left to domain experts and the main challenge is to compare different patterns from different representations, since these patterns exist in different spaces. For multiple view dimensionality reduction, Long et al. have addressed this problem by proposing a unified framework [11]. It is assumed that there exists a linear mapping between the consensus embedding and the individual pattern of each single view. They first learn the individual pattern and then construct the linear mapping. Nevertheless, it does not take domain knowledge into consideration. In summary, we will face the first problem, traditional multiple view methods cannot be directly used to solve multiple view semi-supervised dimensionality reduction problem.

For semi-supervised dimensionality reduction, considering the types of prior knowledge, we can classify SSDR approaches into three categories. (1) The first kind of approaches adopts the pre-defined low dimensional representations of several points. Typical method is proposed by Yang et al. [12]. They first reformulated several typical nonlinear dimensionality reduction techniques into a common problem and then constructed a linear relationship between the predefined embeddings and the unknown embeddings by minimizing above common problem. (2) Methods of the second type employ domain knowledge in the form of pairwise constraints. Zhang et al. [13] first specified whether a pair of instances belongs to the same class (must-link constraint) or different classes (cannot-link constraint) and then computed linear transformations by maximizing distances between the cannot-link pairs and minimizing distances between the must-link pairs. (3) The third type of SSDR methods uses label information directly. They employ labeled and unlabeled data points to construct a weight graph. Typical method may include semi-supervised discriminant analysis (SDA) [14], which is an extension of linear discriminant analysis (LDA) [4]. It used the labeled data points to maximize the separability between different classes and the unlabeled data points to estimate the intrinsic geometric structure of the data. However, these approaches do not concern about the disparate structures of multiple view high dimensional data, they treat all views of a point in the same way. Traditional semi-supervised multiple learning approaches have also shown that treating multiple representations instead of connecting all features into one view can improve classification performance [1,15]. In summary, we will face the second problem, it seems that all of these semi-supervised dimensionality reduction methods are not suitable for multiple view data, since they all focus on data points with a single view.

In this paper, we will propose a new approach named multiple view semi-supervised dimensionality reduction (MVSSDR) to solve the above-mentioned problems. We also focus on the domain knowledge in the form of pairwise constraints since it arises naturally in some applications. The embedding for each view is computed and simultaneously the transformation from consensus pattern to representations of each view is constructed. The consensus pattern can be effectively computed by alternative optimization [16]. We prove theoretically that the objective function is non-increasing and SSDR approach in [13] can be regarded as a special case of our method. We also provide an easy method for extending MVSSDR to out-of-sample data points. Experiments on different real data sets are presented to show the effectiveness of our method. Finally, we provide some basic discussions about multiple view dimensionality reduction.

It is worthwhile to highlight several aspects of the proposed approach here:

- To the best of our knowledge, our approach is the first one to address this kind of multiple view semi-supervised dimensionality reduction problem.
- Comparing with traditional multiple view learning approaches, MVSSDR performs better since it could effectively use the domain knowledge. Comparing with traditional dimensionality reduction methods (e.g. PCA, LDA and SSDR), which can only be used by connecting representations of all views, MVSSDR also performs better since it concerns about the disparate structures and different statistical properties of different views.
- The proposed method can be effectively solved by alternative optimization. We prove that the objective function is non-increasing under the updating rules and hence the convergence of our algorithm is guaranteed. We also provide some discussions about how to use MVSSDR effectively.
- MVSSDR contains other methods, such as SSDR, as special cases. It is also easy to extend MVSSDR to out-of-sample data set.

The remainder of this paper is organized as follows. In Section 2, we will briefly review the SSDR approach in [13]. Section 3 will show MVSSDR algorithm in detail. The analysis and extension of MVSSDR will be proposed in Section 4. Section 5 presents experimental results on real-world data sets and some important discussions. The conclusions and future works are in Section 6.

2. Notations and related works

In this section, we will briefly review SSDR method. Let us introduce some notations first. A set of n data points in R^D is represented by $\mathcal{X}=\{x_1,x_2,\ldots,x_n\}$. For each column vector $x_k \in \mathcal{X}$, there are l views, i.e., $x_k=[x_{k(1)}^T,x_{k(2)}^T,\ldots,x_{k(l)}^T]^T$ with column vector $x_{k(i)} \in R^{D_i}$, $\sum_{i=1}^{l} D_i=D$. For each view v, there are some pairwise must-link constraints M_v and cannot-link constraints C_v, i.e., instances involved by M_v should be close while instances involved by C_v should be far away in the view v. Here M_v is a set, which contains all must-link point pairs and C_v consists of all cannot-link pairs. The goal of MVSSDR is to find a consensus pattern based low dimensional representations $\mathcal{Y}=\{y_1,y_2,\ldots,y_n\}$. Here $y_i \in R^d$ and $d \ll D$. For convenience, let us denote $X=[x_1,x_2,\ldots,x_n]$, $X^{(i)}=[x_{1(i)},x_{2(i)},\ldots,x_{n(i)}]$ and $Y=[y_1,y_2,\ldots,y_n]$. Thus, $X \in R^{D\times n}$, $X^{(i)} \in R^{D_i\times n}$, $Y \in R^{d\times n}$. We summarize these notations in Table 1.

Due to its simplicity and efficiency in some cases, SSDR is one of the most widely used semi-supervised dimensionality reduction methods. However, it mainly focuses on data of only one view. Without confusion, we also assume that $X=[x_1,x_2,\ldots,x_n]$ is the high dimensional data of only one view. M contains point pairs that have must-link constraints and C consists of the cannot-link point pairs. The goal of SSDR is to find a set of projective vectors

Table 1
Some frequently used notations.

n	The number of data points
D	The dimensionality of high dimensional data points of all views
D_v	The dimensionality of high dimensional data points of the v th view
d	The dimensionality of consensus pattern
d_v	The dimensionality of v th view's pattern
X	The data matrix of the size $D \times n$
$X^{(v)}$	The data matrix of the v th view
M_v	The set of point pairs who have must-link constraints in the v th view
C_v	The set of point pairs who have cannot-link constraints in the v th view
Y	The consensus pattern of the size $d \times n$

$W=[w_1, w_2, \ldots, w_d]$, such that the transformed low dimensional representations, $y_i = W^T x_i$ for $i = 1, 2, \ldots, n$, can preserve the structures of original data as well as the pairwise constraints. For convenience, we consider one dimensional case here, i.e., $d = 1$. It is not difficult to extend it to high dimensions. The objective function of SSDR is defined as maximizing $J(W)$, w.r.t. $W^T W = I$, where

$$J(W) = \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(W^T x_i - W^T x_j)^2 + \frac{\alpha}{2n_C}\sum_{(x_i,x_j)\in C}(W^T x_i - W^T x_j)^2 - \frac{\beta}{2n_M}\sum_{(x_i,x_j)\in M}(W^T x_i - W^T x_j)^2. \quad (1)$$

Here, n_C and n_M are the numbers of cannot-link and must-link constraints. α and β are scaling parameters to balance the contributions of different kinds of constraints. The first item in Eq. (1) is the average squared distance between all data points. Through this way, the contribution of unlabeled data (the data without prior link knowledge) is included.

As shown in [13], there exists a concise form for $J(W)$ as follows:

$$J(W) = \frac{1}{2}\sum_{i,j}(W^T x_i - W^T x_j)^2 S_{ij}, \quad (2)$$

where

$$S_{ij} = \begin{cases} \frac{1}{n^2} + \frac{\alpha}{n_C} & \text{if } (x_i, x_j) \in C, \\ \frac{1}{n^2} - \frac{\beta}{n_M} & \text{if } (x_i, x_j) \in M, \\ \frac{1}{n^2} & \text{otherwise.} \end{cases} \quad (3)$$

If we denote D as a diagonal matrix whose non-zero elements are column sums of S, i.e., $D_{ii} = \sum_j S_{ij}$ and $L = D - S$, Eq. (2) can be simplified as maximizing $J(W)$ w.r.t. $W^T W = I$, where

$$J(W) = W^T X L X^T W. \quad (4)$$

It is clear that the solution to Eq. (4) can be easily and effectively derived by eigen-decomposition of XLX^T [5]. As pointed in [13], SSDR can preserve both the structures of original high dimensional data and the pairwise constraints specified by users. However, as shown in the following experiments, SSDR is not suitable for multiple view data since it mainly focuses on the single view data.

3. The algorithm

In this section, we will formally present our MVSSDR algorithm, which aims to discover the consensus low dimensional representations of multiple view points.

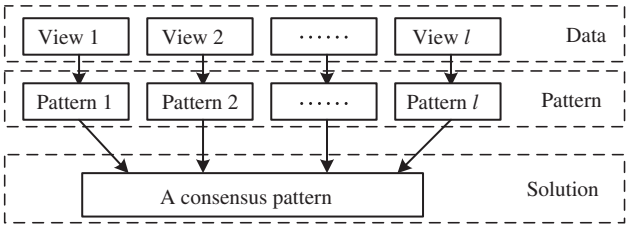

Fig. 1. The basic procedure of MVSSDR.

3.1. Data structure representation

Assume that $X^{(v)} \in R^{D_v \times n}$ is the data matrix of the v th view together with some pairwise must-link constraints M_v and cannot-link constraints C_v. We apply the same strategy as SSDR to represent the data structure and pairwise constraints for each view. More concretely, we compute the Laplacian matrix $L^{(v)}$ for the v th view, $v = 1, 2, \ldots, l$.

$$L^{(v)} = D^{(v)} - S^{(v)}, \quad (5)$$

where

$$S_{ij}^{(v)} = \begin{cases} \frac{1}{n^2} + \frac{\alpha_v}{n_C^{(v)}} & \text{if } (x_{i(v)}, x_{j(v)}) \in C_v, \\ \frac{1}{n^2} - \frac{\beta_v}{n_M^{(v)}} & \text{if } (x_{i(v)}, x_{j(v)}) \in M_v, \\ \frac{1}{n^2} & \text{otherwise,} \end{cases} \quad (6)$$

and $D^{(v)}$ is a diagonal matrix whose elements are the column sums of $S^{(v)}$, i.e., $D_{ii}^{(v)} = \sum_j S_{ij}^{(v)}$. $n_C^{(v)}$ is the cardinality of C_v and $n_M^{(v)}$ is the cardinality of M_v. α_v and β_v are two balance parameters. Intuitively, distances between samples involved in the cannot-link set C_v should typically be close to the expected distance, so we empirically set $\alpha_v = 1$ and $\beta_v > 1$.

3.2. Problem formulation

We now show how to derive the consensus low dimensional representations in this subsection. Assume that our MVSSDR algorithm is to learn a hidden pattern individually from each representation of multiple view data and simultaneously to learn the consensus hidden pattern from those multiple patterns. The basic procedure of MVSSDR is shown in Fig. 1.

Assume that the projection matrix is denoted by $W_v = [w_1^{(v)}, w_2^{(v)}, \ldots, w_{d_v}^{(v)}]$ for the v th view, the low dimensional embedding of the v th view is $W_v^T X^{(v)}$. As pointed in [11], we expect that the consensus pattern is shared by multiple patterns as much as possible. Since the multiple patterns are learned in a separate way, they are not directly comparable. To solve this problem, we assume that there exists a matrix P_v, which transforms the consensus hidden pattern Y to the v th view pattern $W_v^T X^{(v)}$ linearly. In other words, each view pattern can be approximated by the consensus pattern through linear transformation. Thus, the objective function of MVSSDR is

$$J(W_1, \ldots, W_l, P_1, \ldots, P_l, Y) = \sum_{v=1}^{l}\|W_v^T X^{(v)} - P_v Y\|^2 - \lambda\sum_{v=1}^{l}\mathrm{tr}(W_v^T X^{(v)} L^{(v)} (X^{(v)})^T W_v). \quad (7)$$

Here $X^{(v)} \in R^{D_v \times n}$, $W_v \in R^{D_v \times d_v}$, $L^{(v)} \in R^{n \times n}$, $Y \in R^{d \times n}$ and $P_v \in R^{d_v \times d}$. d_v is the dimensionality of the embedding in the v th view. The first part measures the errors of approximating each view pattern by the consensus pattern and the second part is the objective function in computing the embedding of each view. λ is the parameter which can balance the weights of two items. It should not be too large or too small. In the following experiments, it is determined by five fold cross validation.

We now explain how to construct the objective function in Eq. (7). There are mainly two optimization problems, i.e., to compute the transformation matrix W_v for the v th view and to compute the transformation matrix P_v. We can employ two different strategies. One way is to solve these two problems orderly and the other is to optimize them simultaneously by adding their objective functions. We employ the second strategy. The reasons are: (1) Since the consensus pattern is not only related to transformation matrix P_v, but also related to each view pattern. It should be determined by these two factors simultaneously, not orderly. (2) As shown in the next subsection, we initialize Eq. (7) by employing the first strategy in our procedure. After several times iteration, the objective function in Eq. (7) achieves smaller optimal value and our solution is much closer to the optimal solution to Eq. (7).

Similarly, we add some constraints about these unknown variables as in SSDR. Commonly, the transformation matrix W_v is orthogonal, i.e., $W_v^T W_v = I_{d_v \times d_v}$. In order to remove the rotational degree of freedom and to fix the scale, we constrain that $YY^T = I_{d \times d}$ as in most dimensionality reduction methods [5].

Generally, MVSSDR can be regarded as solution to the following problem:

$$\begin{aligned} \min \quad & J(W_1, \ldots, W_l, P_1, \ldots, P_l, Y) \\ \text{s.t.} \quad & W_v^T W_v = I_{d_v \times d_v} \quad \text{for} \quad v = 1, 2, \ldots, l \\ & YY^T = I_{d \times d}. \end{aligned} \tag{8}$$

This problem cannot be directly solved by simple methods, such as spectral decomposition. In the next subsection, we will apply the alternative optimization approach to find the approximated solution.

A natural question about the model in Eq. (7) is why we do not map $W_v^T X^{(v)}$ into Y. Or equivalently, we employ the following objective function:

$$\begin{aligned} J(W_1, \ldots, W_l, P_1, \ldots, P_l, Y) = & \sum_{v=1}^{l} \|(W_v^T X^{(v)}) P_v - Y\|^2 \\ & - \lambda \sum_{v=1}^{l} \operatorname{tr}(W_v^T X^{(v)} L^{(v)} (X^{(v)})^T W_v). \end{aligned} \tag{9}$$

The above function looks similar to the model in Eq. (7). However, it has a serious problem. Since Y integrates the embeddings of each view, it is common that Y contains more information than $W_v^T X_{(v)}$ for $v=1, 2, \ldots, l$. From information theorem, the transformation P_v cannot add information. Moreover, the problem in Eq. (9) is hard to solve because W_v and P_v occur in the same item and it is impossible to distinguish them without adding extra constraints. On the contrary, we do not have this problem in Eq. (7).

In summary, we have integrated two distinct optimization problems into a united one, in which we can consider the interaction of two parts. This interaction is very important for the problem of multiple view semi-supervised dimensionality reduction (see the results in experiments).

3.3. Solution

The optimization problem in Eq. (8) is a multi-variable function, there are mainly three kinds of unknown parameters, i.e., W_v, P_v and Y. Since this problem is non-convex, it is difficult to optimize them simultaneously. Hence, we apply the alternative optimization strategy, which fixes some parameters first and optimizes the other parameters. Since only the first part involves P_v, we will first determine this parameter when W_v and Y are fixed.

For any given W_v and Y, the optimal solution to Eq. (8) is

$$P_v = W_v^T X^{(v)} Y^T. \tag{10}$$

We will prove this result in Section 4.1. Meanwhile, since $\|W_v^T X^{(v)} - P_v Y\|^2 = \operatorname{tr}((W_v^T X^{(v)} - P_v Y)^T (W_v^T X^{(v)} - P_v Y))$, the objective function in Eq. (7) becomes

$$\begin{aligned} J(W_1, \ldots, W_l, P_1, \ldots, P_l, Y) = & \sum_{v=1}^{l} \{\operatorname{tr}((X^{(v)})^T W_v W_v^T X^{(v)}) \\ & - 2 \operatorname{tr}(Y^T P_v^T W_v^T X^{(v)}) + \operatorname{tr}(Y^T P_v^T P_v Y)\} \\ & - \lambda \sum_{v=1}^{l} \operatorname{tr}(W_v^T X^{(v)} L^{(v)} (X^{(v)})^T W_v). \end{aligned} \tag{11}$$

Since $P_v = W_v^T X^{(v)} Y^T$, $YY^T = I_{d \times d}$ and $\operatorname{tr}(A^T A) = \operatorname{tr}(AA^T)$, Eq. (11) becomes

$$\begin{aligned} J(W_1, \ldots, W_l, P_1, \ldots, P_l, Y) = & \sum_{v=1}^{l} \operatorname{tr}(W_v^T X^{(v)} (X^{(v)})^T W_v) \\ & - \sum_{v=1}^{l} \operatorname{tr}(W_v^T X^{(v)} Y^T Y (X^{(v)})^T W_v) \\ & - \lambda \sum_{v=1}^{l} \operatorname{tr}(W_v^T X^{(v)} L^{(v)} (X^{(v)})^T W_v), \end{aligned} \tag{12}$$

or more concretely,

$$\begin{aligned} J(W_1, \ldots, W_l, P_1, \ldots, P_l, Y) = & \sum_{v=1}^{l} \operatorname{tr}\{W_v^T [X^{(v)} (X^{(v)})^T - \lambda X^{(v)} L^{(v)} (X^{(v)})^T \\ & - X^{(v)} Y^T Y (X^{(v)})^T] W_v\}. \end{aligned} \tag{13}$$

If we have fixed the matrix Y, the problem in Eq. (8) will become

$$\begin{aligned} \min \quad & \sum_{v=1}^{l} \operatorname{tr}\{W_v^T [X^{(v)} (X^{(v)})^T - \lambda X^{(v)} L^{(v)} (X^{(v)})^T \\ & - X^{(v)} Y^T Y (X^{(v)})^T] W_v\} \\ \text{s.t.} \quad & W_v^T W_v = I_{d_v \times d_v} \quad \text{for} \quad v = 1, 2 \ldots, l. \end{aligned} \tag{14}$$

Since W_v is independent with each other, we can compute the optimal W_v, which minimizes the objective function in Eq. (14), by spectral decomposition of $[X^{(v)} (X^{(v)})^T - \lambda X^{(v)} L^{(v)} (X^{(v)})^T - X^{(v)} Y^T Y (X^{(v)})^T]$. W_v is formed by the eigenvectors corresponding to the d_v smallest eigenvalues for $v = 1, 2, \ldots, l$.

Correspondingly, if we fix all W_vs, the problem in Eq. (8) is equivalent to

$$\begin{aligned} \max \quad & \operatorname{tr}\left\{ Y \left[\sum_{v=1}^{l} (X^{(v)})^T W_v W_v^T X^{(v)} \right] Y^T \right\} \\ \text{s.t.} \quad & YY^T = I_{d \times d}. \end{aligned} \tag{15}$$

This problem can also be solved by spectral decomposition of the matrix $[\sum_{v=1}^{l} (X^{(v)})^T W_v W_v^T X^{(v)}]$. Different from previous problem in Eq. (14), Y should be assigned by the eigenvectors corresponding to the d largest eigenvalues.

We solve the above two optimization problems alternatively. This is an iterating algorithm and we must determine its initialization.

Table 2
Multiple view semi-supervised dimensionality reduction.

Input
The whole data matrix X
Must-link constraints M_v, cannot-link constraints C_v, for $v=1,2,\ldots,l$

Output
The consensus pattern Y, the transformation matrices P_v and W_v

1. Compute $L^{(v)}$ by Eq. (5) for $v=1,2,\ldots,l$
2. Initialize W_v by the eigenvectors corresponding to the d_i largest eigenvalues of $X^{(v)}L^{(v)}(X^{(v)})^T$
3. Repeat the following steps until convergence:
   (a) Fixing W_v, updating Y by eigenvectors corresponding to the d largest eigenvalues of $[\sum_{v=1}^{l}(X^{(v)})^T W_v W_v^T X^{(v)}]$
   (b) Fixing Y, updating W_v by eigenvectors corresponding to the d_v smallest eigenvalues of $[X^{(v)}(X^{(v)})^T - \lambda X^{(v)}L^{(v)}(X^{(v)})^T - X^{(v)}Y^T Y(X^{(v)})^T]$
4. Compute $P_v = W_v^T X^{(v)} Y^T$

Fortunately, W_v has a natural initialization. We can apply the SSDR solution of each view to determine W_v. More concretely, we initialize W_v by the eigenvectors corresponding to the d_i largest eigenvalues of $X^{(v)}L^{(v)}(X^{(v)})^T$. Here, W_v is determined without considerations about its influence on the transformations from consensus pattern to each view pattern. Once W_vs are determined, Y can be derived by solving problem in Eq. (15), then the new Y is substituted in Eq. (14) and the new W_vs are computed. Through this iteration, we can find the approximate solutions of Y and W_vs. Finally, P_v is computed by Eq. (10).

Another concern is the terminal conditions and convergence behavior. We will prove that this kind of iteration converges in Section 4.1. The terminal conditions are determined by measuring the Frobenius norm between the computed Y and the previous Y. Simultaneously, we also compute the distance between the updated W_v and the previous W_v. If they are all small enough ($1e-5$ in the following experiments), we will stop the iteration. Since W_v is well initialized, there are only a few times of iterations. The number of iterations is between 5 and 15 in the following experiments.

Finally, we would like to analyze the computational complexity of MVSSDR. In each iteration, the complexities for two spectral decompositions are $\sum_{v=1}^{l} O(D_v^3)$ and $O(N^3)$. If these matrices are sparse, the complexity could also be reduced. Thus, after t times iteration, the total computational complexity is $t(\sum_{v=1}^{l} O(D_v^3) + O(N^3))$. Since we have initialized this problem effectively, t can be neglected. Therefore, the adding computational requirement is also negligible.

The main procedure of MVSSDR is summarized in Table 2.

4. Analysis and extensions

First, we will prove that the optimal P_v, which minimizes the objective function in Eq. (8) can be determined by Eq. (10). Second, the convergence behavior of our algorithm is shown. Then, the relationship between our method and SSDR, the relationship between MVSSDR and the method proposed in [11] are also analyzed. Finally, we give a simple way to extend MVSSDR to the out-of-sample data.

4.1. Performance analysis

Theorem 1. *For any fixed W_v and Y, the optimal P_v, which minimizes the objective function in Eq.* (8)*, is $P_v = W_v^T X^{(v)} Y^T$.*

Proof. Since P_v only occurs in the first part of $J(W_1,\ldots,W_l,P_1,\ldots,P_l,Y)$, let $f(W_1,\ldots,W_l,P_1,\ldots,P_l,Y)$ denote the first part of Eq. (7) and the minimization of $J(W_1,\ldots,W_l,P_1,\ldots,P_l,Y)$ is equivalent to the minimization of $f(W_1,\ldots,W_l,P_1,\ldots,P_l,Y)$, which has the following form:

$$\begin{aligned} f(W_1,\ldots,W_l,P_1,\ldots,P_l,Y) &= \sum_{v=1}^{l} \mathrm{tr}((W_v^T X^{(v)} - P_v Y)^T (W_v^T X^{(v)} - P_v Y)) \\ &= \sum_{v=1}^{l} \mathrm{tr}((X^{(v)})^T W_v W_v^T X^{(v)} - 2Y^T P_v^T W_v^T X^{(v)} + Y^T P_v^T P_v Y) \\ &= \sum_{v=1}^{l} \mathrm{tr}((X^{(v)})^T W_v W_v^T X^{(v)} - 2W_v^T X^{(v)} Y^T P_v^T + P_v P_v^T). \end{aligned}$$

In the last step of the above deduction, we use the conclusion that $\mathrm{tr}(AB)=\mathrm{tr}(BA)$ if AB and BA are all multiplicative, and $YY^T = I_{d\times d}$. Take the derivation of $f(W_1,\ldots,W_l,P_1,\ldots,P_l,Y)$ w.r.t. P_v, we obtain

$$\frac{\partial f(W_1,\ldots,W_l,P_1,\ldots,P_l,Y)}{\partial P_v} = -2W_v^T X^{(v)} Y^T + 2P_v \quad \text{for} \quad v=1,2,\ldots,l.$$

According to the KKT condition, we solve $\partial f(W_1,\ldots,W_l,P_1,\ldots,P_l,Y)/\partial P_v = 0$ to obtain Eq. (10). The proof completes. □

We will show that the objective function $J(W_1,\ldots,W_l,P_1,\ldots,P_l,Y)$ in Eq. (8) is non-increasing under the pre-defined updating rules of Y and W_v.

Theorem 2. *The objective function $J(W_1,\ldots,W_l,P_1,\ldots,P_l,Y)$ of MVSSDR shown in Eq.* (8) *is non-increasing under the updating rules of Y and W_v, which are shown in Table* 1.

Proof. From Theorem 1, we know that P_v can be expressed by W_v and Y. Thus, we only consider the updating rule of W_v and Y. For brief, denote the objective function of MVSSDR in Eq. (7) as $J(W_1,\ldots,W_l,Y)$. Assume that, after i times iterations, we now achieve $W_v^{(i)}$ and $Y^{(i)}$. When we fix $Y^{(i)}$ and update $W_v^{(i)}$ by solving the optimization problem in Eq. (14), $W_v^{(i+1)}$ can be derived by eigen-decomposition. Thus,

$$J(W_1^{(i+1)},\ldots,W_l^{(i+1)},Y^{(i)}) \le J(W_1^{(i)},\ldots,W_l^{(i)},Y^{(i)}). \tag{16}$$

Similarly, when we fix $W_v^{(i+1)}$ for $v=1,\ldots,l$ and update $Y^{(i)}$ by solving the optimization problem in Eq. (15), we have

$$J(W_1^{(i+1)},\ldots,W_l^{(i+1)},Y^{(i+1)}) \le J(W_1^{(i+1)},\ldots,W_l^{(i+1)},Y^{(i)}). \tag{17}$$

By Eqs. (16) and (17), we can conclude that

$$J(W_1^{(i+1)},\ldots,W_l^{(i+1)},Y^{(i+1)}) \le J(W_1^{(i)},\ldots,W_l^{(i)},Y^{(i)}). \tag{18}$$

The above equation indicates that MVSSDR achieves smaller objective values after $i+1$ times iterations. This implies that the objective function is non-increasing and the result follows. □

Theorem 2 shows that the objective function of MVSSDR is non-increasing under our updating rule. Moreover, $J(W_1,\ldots,W_l,Y)$ has lower boundaries. Thus, our algorithm is convergent when $i\to\infty$.

4.2. Relations to other approaches

In this section, we will first show the relationship between our method and SSDR [13]. As what we have mentioned earlier, in the iteration of MVSSDR, SSDR is employed to derive the low dimensional embeddings for each view. If we connect representations in all views to form the data matrix X, SSDR can be directly employed on X. Now, $l=1$ and the objective function $J(W_1,\ldots,W_l,P_1,\ldots,P_l,Y)$ shown in Eq. (7) becomes

$$J(W_1,P_1,Y)=\|(W_1^TX^{(1)})-P_1Y\|^2-\lambda\,\mathrm{tr}(W_1^TX^{(1)}L^{(1)}(X^{(1)})^TW_1). \tag{19}$$

Here $X^{(1)}=X$ since $l=1$. Obviously, SSDR can be regarded as the degradation of MVSSDR when l=1. In other words, we have extended SSDR to deal with data with multiple views.

Moreover, our method also has close relationship with the method proposed in [11]. In essential, the problem of multiple view dimensionality reduction is firstly addressed in [11]. It assumes that the dimensionality reduction results for each view are known. Assume that A_i are these representations, the method in [11] aims to find solution to the following problem:

$$\begin{aligned}\min\quad & \sum_{i=1}^{l}\|A_i-P_iY\|^2\\ \text{s.t.}\quad & P_i^TP_i=I.\end{aligned} \tag{20}$$

Comparing with the formulation of MVSSDR shown in Eq. (8), the problem in Eq. (20) takes no consideration about the pairwise constraints. It assumes that the low dimensional embedding for each view has already known. Moreover, the derivations of each view pattern have no relationships with the computing of consensus pattern. Our method, however, combines these two procedures together and derives more realistic embedding when the data have multiple views.

4.3. Induction for out-of-sample data

Until now, we have introduced the main procedure. In this section, we will extend MVSSDR to the out-of-sample data.

Since the transformation matrices W_v and P_v have been derived based on the ever-known data. They can be directly used for the out-of-sample data. Take one out-of-sample data $\xi=[\xi_1^T,\xi_2^T,\ldots,\xi_l^T]^T$ as an example. Here $\xi_v\in R^{D_v}$, for $v=1,2,\ldots,l$. We expect to find the consensus pattern $\eta\in R^d$. Similarly, the low dimensional embedding of ξ in the v th view is $W_v^T\xi_v$, for $v=1,2,\ldots,l$. Based on the same strategy, which we have used in MVSSDR to approximate each view pattern by the consensus pattern η, we expect η to be the solution to the following problem:

$$\min\sum_{v=1}^{l}\|W_v^T\xi_v-P_v\eta\|^2. \tag{21}$$

In order to solve this problem in a closed form, we assume

$$P=\begin{bmatrix}P_1\\P_2\\\vdots\\P_l\end{bmatrix},\quad B=\begin{bmatrix}W_1^T\xi_1\\W_2^T\xi_2\\\vdots\\W_l^T\xi_l\end{bmatrix}.$$

Thus, Eq. (21) becomes to

$$\min(P\eta-B)^T(P\eta-B). \tag{22}$$

The following theorem shows that the consensus pattern for ξ is $\eta=(P^TP)^\dagger P^TB$.

Theorem 3. *When P and B are known, the optimal solution to the problem in Eq. (22) is $\eta=(P^TP)^\dagger P^TB$.*

The proof is simple and we would like to omit it.

It is obvious that the consensus pattern for an out-of-sample data can be easily obtained. Consequently, our algorithm is inductive. Different from the previous method, the consensus pattern η cannot be directly derived only through ever-known transformations P, since we need to employ the consensus pattern to approximate the single view pattern B.

5. Experiments and discussions

In this section, several experiments are performed for illustration. There are mainly four commonly used data, the WebKB data set[1] [17], the Internet advertisements data set,[2] the 20-Newsgroups data set[3] and the Sonar data set.[4] After reducing the dimensionality, we employ K-means for clustering, except for the specified comparisons. Moreover, in the following experiments, the parameters are determined by five fold cross validation and all numerical results are averaged by 50 independent trials. We also provide some important discussions about multiple view dimensionality reduction.

For illustration, we compare our method with unsupervised methods: PCA and CCA, supervised method: LDA and semi-supervised method SSDR. CCA is first used to derive separate embeddings of two views as in [8]. We then connect them as the embedding of original multiple view data. The accuracy of K-means on the original data is regarded as the baseline. We employ SSDR on each separate view to derive the low dimensional embeddings. After that, K-means is applied 10 times with different starting points and the best result in terms of the objective function of K-means is recorded. Since the performance of method in [11] has close relationship with the embedding of each view, it is not convenient to compare it with our method.

5.1. Data set description and evaluation metrics

The above data sets are widely used in multiple view learning [11,17]. Since they are too large to process, we apply some preprocessing techniques.

The 20-Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) across 20 different newsgroups. Each newsgroup corresponds to a topic. Some of the newsgroups are closely related with each other (e.g., comp.sys.ibm.pc.hardware/comp.sys.mac.hardware), while others are highly unrelated (e.g., misc.forsale/soc.religion.christian). We apply the same preprocessing technique as in [13] to derive three feature sets, including News-Different-300, News-Similar-300 and News-Same-300. The representation of a point is divided into three parts. As denoted by their names, these features are intentionally selected to satisfy the independent assumption. We pre-reduce the dimensionality of a point in each feature set to 100. As the same way in [11], since the three features are descriptions from independent

[1] http://www.cs.cmu.edu/~webkb/

[2] http://archive.ics.uci.edu/ml/datasets/Internet+Advertisements

[3] http://people.csail.mit.edu/jrennie/20Newsgroups/

[4] http://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)

Table 3
Descriptions of WebKB data set.

Data sets	Sizes	Classes	View1	View2
WebKB-I				
Course and non	1051	2	2949	334
WebKB-II				
Cornell	827	7	4134	827
Washington	1166	7	4165	1166
Wisconsin	1210	6	4189	1210
Texas	814	7	4029	814

feature spaces, we construct the multiple view data by selecting features from any two feature subsets.

The Internet advertisements data are a set of possible advertisements on web pages. The task is to predict whether a web is an advertisement ("ad") or not ("non-ad"). This data set contains 3264 examples, among which 458 examples are advertisements. We select five feature sets.

url: information of phrases occurring in the URL, dimensionality: 457;
origurl: information of the image's URL, dimensionality: 495;
alt: information of the alt terms, dimensionality: 111;
ancurl: information of the anchor text, dimensionality: 472;
caption: information of the words occurring near the anchor text, dimensionality: 19.

Since these features are disparate descriptions, we can consider them as five different views of a web and combine every two views to form a multiple view data.

The third data set is the WebKB. It consists of web pages that are collected from computer science departments of various universities. There are two independent descriptions: fulltext—the text on the web pages, and inlinks—the anchor text on the hyperlinks pointing to the page. It is natural to take these two descriptions as two views since they are independent descriptions from different feature spaces. In this way, the features of each sample can be split into two views: the content features and link features.

We classify the WebKB data set from two different aspects and formulate two different subsets. From one aspect, it can be classified into two categories: course and non-course. We formulate a subset called WebKB-I. The 1051 pages are manually classified into two categories: course (230) and non-course (821). From the other aspect, we construct the WebKB-II by collecting about 6000 web pages from computer science department in four Universities, i.e., Cornell, Texas, Washington, and Wisconsin. Table 3 summarizes the characteristics of these two subsets of WebKB.

Finally, we also employ the Sonar data set from the UCI. The feature of this data is the energy within a particular frequency band, integrated over a certain period of time. Since these features have relaxed relationship with each other, we can formulate the multiple view data by splitting the feature equally and orderly. For example, if the original data has D dimensions and the multiple view data has two views, the first $D/2$ features are regarded as the first view and the rest are the second view. The same strategy is applied for constructing data with more than two views.

To evaluate the performances of different methods quantitatively, we employ two evaluation metrics: clustering accuracy (Acc) [18] and normalized mutual information (NMI) [19].

Acc discovers one-to-one relationship between clusters and the true classes. It measures the extent to which cluster contains examples from the corresponding category. We compute it by summing up the whole matching degree between all pair clusters. The higher clustering accuracy means the better clustering performance. It can be computed by

$$\mathrm{Acc} = \frac{1}{n}\sum_{i=1}^{n}\delta(y_i, \mathrm{map}(c_i)), \tag{23}$$

where map$(\cdot)$ is a function that maps each cluster index to a class label. It can be found by the Hungarian algorithm [18]. c_i and y_i are the cluster index and true class label of x_i. $\delta(a,b)$ is the function that equals 1 when a equals b and 0 otherwise.

NMI could measure the similarity between the clustering results and the true classes. Assume that A is the clustering result and B is the true class. The normalized mutual information NMI(A,B) is

$$\mathrm{NMI}(A,B) = \frac{I(A,B)}{\sqrt{H(A)H(B)}}. \tag{24}$$

Here, $I(A,B)$ denotes the mutual information between A and B. $H(A)$ denotes the entropy of A. Thus, the larger NMI(A,B) is, the better this method performs. Please see [19] for more discussions.

5.2. Experiments on the 20-Newsgroups data set

In order to illustrate that the consensus pattern could represent the original data more realistic than each view's pattern, we have done an experiment for visualization on the 20-Newsgroups data set. As mentioned above, we select the data, whose features are in News-Different-300 and News-Same-300, from the first and the second categories. The features in News-Different-300 are regarded as the first view. Thus, $l=2$, $c=2$ and $D=200$, here c is the number of classes. We randomly choose 60 pairs of points for each view. If two points belong to the same class, we connect them by a must-link, otherwise, a dissimilarity pair is created.

We apply SSDR in each view of the data and reduce the dimensionality to two. These embeddings are shown in Fig. 2(a) and (b) respectively. Different types of representations belong to different classes. We also apply MVSSDR to achieve the consensus pattern, which is shown in Fig. 2(c).

Intuitively, comparing with the results shown in Fig. 2(a) and (b), the results in Fig. 2(c) can represent original data in a more reasonable way. The original multiple view data are projected in a separable way. However, if we consider the embedding of each view, it is difficult to distinguish points in class one from that in class two.

Similarly, we have also compared the performances of SSDR, method in [11] with our method. Representations from the News-Different-300 and News-Same-300 are chosen. Different from the previous data, these points belong to the first and the third categories. Other settings are the same as previous. We show their results in Fig. 3. It is obvious that MVSSDR projects the original data from different categories in a separable way. The embeddings of SSDR and method in [11], however, are overlapped, they cannot represent the structure of original data suitably in this situation.

We have also performed some experiments to compare our method with other methods quantitatively. Every two views of the samples from every two classes are combined to form the multiple view data. We compare our method with K-means on each separate view and all views, SSDR on each view and all views. The unsupervised PCA, CCA approaches and supervised LDA approach are also employed directly on the connected vectors of all views. We compare the clusters generated by these algorithms with the true classes by computing the evaluation metrics: clustering accuracy (Acc). Denote the News-Different-300 as view1 (V_1), the News-Same-300 as view2 (V_2), the News-Similar-300 as view3 (V_3) and the three classes as C_1, C_2 and C_3, all the Acc results, which are averaged by 50 independent trials, are listed in Table 4.

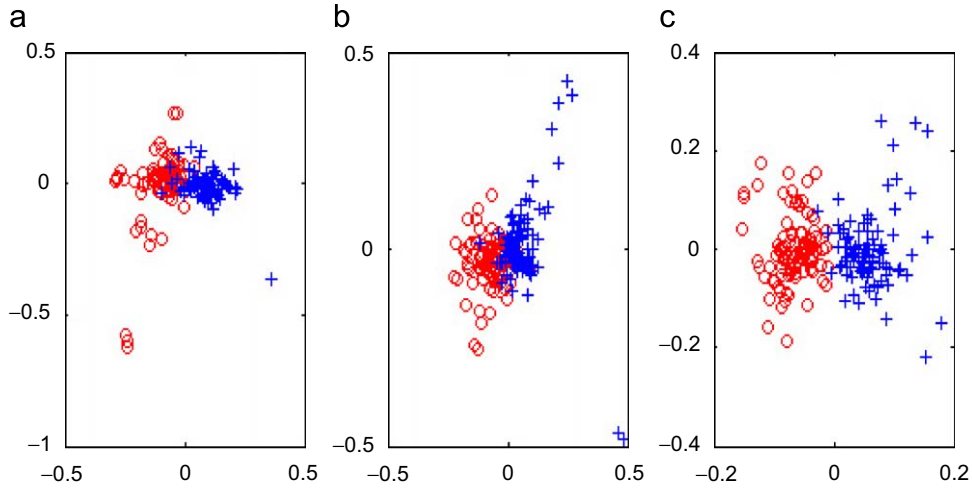

Fig. 2. Semi-supervised dimensionality reduction results on the multiple view 20-Newsgroups data: (a) SSDR embedding of the first view, (b) SSDR embedding of the second view and (c) MVSSDR embedding for multiple view data.

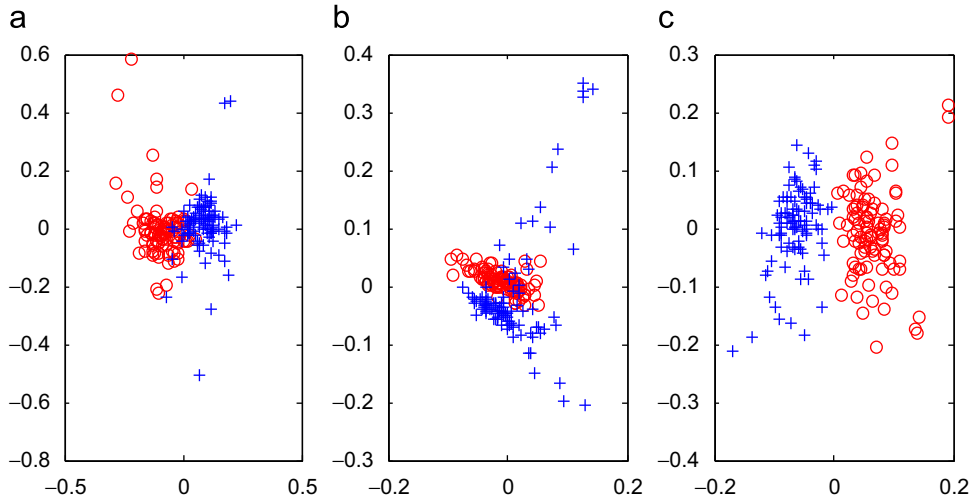

Fig. 3. Dimensionality reduction results on the multiple view 20-Newsgroups data: (a) SSDR embedding of all views, (b) embedding derived by the method in [11] and (c) MVSSDR embedding.

As seen from Table 4, it is obvious that our method outperforms the other approaches, including the unsupervised PCA, CCA approaches and supervised LDA approach, since they treat the multiple view data by simply connecting each representation and take no considerations about its departure properties. Moreover, it achieves higher accuracy than performing SSDR on each view, although we use the same method to derive embeddings on each view. In other words, the interaction of two items in Eq. (7) can help us to achieve a more realistic consensus pattern. More interestingly, the performance of SSDR on data with all views is worse than that of SSDR on each single view. This can partially explain why we cannot treat multiple view data by simply connecting features in each view.

From the results in Table 4, we can also conclude that to reduce the dimensionality is not always beneficial to the following process. Only when the dimensionality reduction approach is suitably selected, it can improve the performance of the further processing.

5.3. Experiments on the Internet advertisements data set

In this section, we address the task of text clustering using the Internet advertisements data set. As mentioned in Section 5.1, there are five views for each point. We construct the multiple data by combining representations of “caption” and “alt” with other views. Since these data points belong to two different classes, “ad” or “non-ad”, K-means are employed for clustering. We also randomly construct 100 links, including must-links and cannot-links.

Similarly, we have employed the same methods as in previous experiments for comparison and set $d_1 = 40$, $d_2 = 20$, $d = 10$ manually. All the other parameters are determined by five fold cross validation. For simplicity, we denote “url” as V_1, “origurl” as V_2, “alt” as V_3, “ancurl” as V_4 and “caption” as V_5. The Acc results, which are averaged over 50 independent trials, are shown in Table 5.

From Table 5, it is obvious that our method performs best. In other words, the computed consensus pattern could reveal the intrinsic structure of multiple view data. Moreover, it achieves the highest accuracy when the two views are “alt” and “caption”. This means that these two features seem to be compatible, i.e., the combination of these two features could represent the original data best. Similarly, we can also conclude that the combination of representations of different views would degrade the performance of SSDR.

Table 4
Clustering accuracies (Acc) results on different subsets of the 20-Newsgroups data set.

Methods	C_1&C_2 V_1&V_2	C_1&C_3 V_1&V_2	C_2&C_3 V_1&V_2	C_1&C_2 V_1&V_3	C_1&C_3 V_1&V_3
Original (all)	0.8146	0.9056	0.8180	0.8210	0.8770
Original (View1)	0.7891	0.8546	0.8044	0.7776	0.8390
Original (View2)	0.6293	0.7360	0.6395	0.7066	0.6365
PCA (all)	0.9184	0.9181	0.9177	0.8976	0.8901
LDA (all)	0.9205	0.9281	0.9221	0.9292	0.9181
CCA (all)	0.8932	0.9097	0.8900	0.9020	0.9026
SSDR (all)	0.5306	0.5293	0.5663	0.5184	0.5301
SSDR (View1)	0.9309	0.9368	0.9458	0.9279	0.9551
SSDR (View2)	0.8367	0.8457	0.7908	0.9026	0.7487
MVSSDR	**0.9611**	**0.9658**	**0.9643**	**0.9589**	**0.9743**

Methods	C_2&C_3 V_1&V_3	C_1&C_2 V_2&V_3	C_1&C_3 V_2&V_3	C_2&C_3 V_2&V_3
Original (all)	0.8846	0.7789	0.8168	0.7145
Original (View1)	0.8166	0.7013	0.6296	0.6564
Original (View2)	0.6567	0.6474	0.7120	0.6393
PCA (all)	0.8800	0.8526	0.8639	0.8272
LDA (all)	0.9077	0.8632	0.8667	0.8318
CCA (all)	0.8910	0.8426	0.8564	0.8044
SSDR (all)	0.5129	0.5184	0.5262	0.5146
SSDR (View1)	0.9541	0.8905	0.7408	0.7504
SSDR (View2)	0.6218	0.8618	0.8635	0.8000
MVSSDR	**0.9733**	**0.9263**	**0.8966**	**0.8684**

Table 5
Clustering accuracies (Acc) results on different subsets of the Internet advertisements data set.

Methods	V_1&V_5	V_2&V_5	V_3&V_5	V_4&V_5	V_1&V_3	V_2&V_3	V_4&V_3
Original (all)	0.8772	0.8620	0.8919	0.8828	0.8613	0.8921	0.8828
Original (View1)	0.8611	0.8602	0.8917	0.8828	0.8688	0.8599	0.8789
Original (View2)	0.8689	0.8689	0.8689	0.8689	0.8917	0.8917	0.8917
PCA (all)	0.8500	0.8583	0.8917	0.8828	0.8922	0.8922	0.8828
LDA (all)	0.8722	0.8672	0.8867	0.8917	0.8961	0.8811	0.8950
CCA (all)	0.8728	0.8694	0.8633	0.8717	0.8978	0.8883	0.8967
SSDR (all)	0.8585	0.8580	0.8622	0.8667	0.8472	0.8521	0.8767
SSDR (View1)	0.8746	0.8624	0.8678	0.8939	0.8686	0.8650	0.8943
SSDR (View2)	0.8593	0.8593	0.8593	0.8593	0.8757	0.8757	0.8757
MVSSDR	**0.9121**	**0.9017**	**0.9250**	**0.9209**	**0.9134**	**0.9214**	**0.9218**

5.4. Experiments on the WebKB data set

The WebKB data set is of multiple view and we have constructed two subsets of WebKB, i.e., WebKB-I and WebKB-II, in our experiments.

The first kind of experiments performs on WebKB-I that has two categories. With different number of links, we have employed the above methods on the data formed by connecting representations of all views, except for MVSSDR. Please note that the results of *K*-means, PCA, LDA and CCA have no relationship with the number of links. We compute their accuracies as the baselines. The Acc results averaged over 50 independent trials are summarized in Fig. 4.

From Fig. 4, we can see that MVSSDR achieves the highest accuracies in all cases. With the increase of links, most of the semi-supervised methods perform better. More importantly, MVSSDR can achieve satisfied accuracy even with a small number of links, e.g., its accuracy is higher than 90% even with 10 links. Comparing with MVSSDR, other semi-supervised methods, however, obtain lower accuracies.

We have also performed some other experiments on another subset of the WebKB data, i.e., the WebKB-II. After using *K*-means for clustering, we compute Acc and NMI between the computed results and the true labels. All these results, which have been averaged over 50 independent trails, are shown in Table 6.

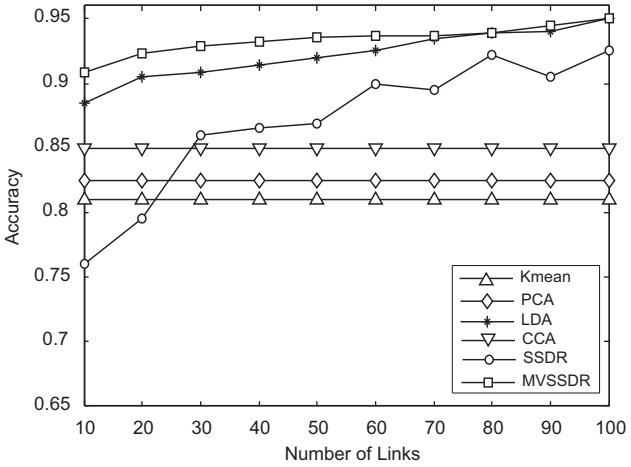

Fig. 4. Acc results on WebKB-I data set.

Table 6
Acc results and NMI on WebKB-II data set.

Methods	*K*-means	PCA	LDA	CCA	SSDR	MVSSDR
Cornell						
Acc	0.5412	0.5937	0.5801	0.6759	0.6439	0.8167
NMI	0.1183	0.1753	0.1686	0.3648	0.3428	0.5230
Wisconsin						
Acc	0.5692	0.5909	0.6084	0.7099	0.7353	0.8033
NMI	0.1571	0.1746	0.1870	0.2811	0.3142	0.4383
Washington						
Acc	0.5661	0.6304	0.6214	0.6639	0.5618	0.7130
NMI	0.2451	0.2829	0.2627	0.3044	0.2413	0.4153
Texas						
Acc	0.5631	0.5835	0.5944	0.6511	0.5958	0.8195
NMI	0.1522	0.1984	0.2050	0.3317	0.2003	0.4370

Obviously, MVSSDR achieves the largest Acc and NMI, it means that the result of MVSSDR is much closer to the truth. More interestingly, although Acc and NMI are two different evaluation metrics, it seems that, in this experiment, the higher Acc is, the larger NMI is and vice versa.

5.5. Discussions

In this section, we will provide some important discussions about the problem of multiple view semi-supervised dimensionality reduction.

The first question is that if the data are only of a single view in essential, should we use MVSSDR by splitting the representations to several views?

Intuitively, the answer is 'not always'. When different dimensions of the original data have close relationship with each other and the original data cannot be well represented without any dimension, the splitting will destroy the intrinsic structure of the original data. Thus, it is difficult for MVSSDR to achieve a satisfied consensus pattern by the inaccurate embedding of each view. Certainly, if some representations are abundant, to employ MVSSDR by splitting the single view data may improve its performance.

We have done some experiments on the 20-Newsgroups data set and the Sonar data set. For each feature set of 20-Newsgroups, we randomly split the 100 dimensional data into two 50 dimensional data and consider them as two views. For Sonar data, the original data points are equally split into two, three and four views in

Table 7
Performance comparison of MVSSDR and SSDR on splitting data.

	C_1&C_2	C_1&C_3	C_2&C_3
News-Different-300			
SSDR	0.9563(0.0175)	0.9635(0.0158)	0.9507(0.0220)
MVSSDR	0.8833(0.0598)	0.8935(0.0786)	0.8342(0.0825)
News-Same-300			
SSDR	0.8099(0.0734)	0.8497(0.1120)	0.7749(0.0946)
MVSSDR	0.6796(0.0713)	0.6835(0.0706)	0.6508(0.0588)
News-Similar-300			
SSDR	0.8292(0.1245)	0.6919(0.1320)	0.7356(0.1019)
MVSSDR	0.6713(0.0579)	0.5861(0.0602)	0.6518(0.0690)
Sonar	Two views	Three views	Four views
SSDR	0.6453(0.0805)	0.6453(0.0736)	0.6453(0.0716)
MVSSDR	0.6776(0.0544)	0.6717(0.0502)	0.6630(0.0532)

Table 8
Acc results of MVSSDR on three-view data sets.

The Internet advertisements data set			
	V_3&V_5&V_1	V_3&V_5&V_2	V_3&V_5&V_4
MVSSDR	0.8755(0.0129)	0.8646(0.0084)	0.9010(0.0086)
The 20-Newsgroups data set			
	C_1&C2	C_1&C_3	C_2&C_3
MVSSDR	0.9276(0.0518)	0.9450(0.0517)	0.9108(0.0657)

sequence (see the data description part for more details). After reducing the dimensionality, we also employ K-means for clustering. SSDR on the original data is also employed for comparison. Table 7 lists the average accuracies (Acc) and the standard derivations (within bracket) of 50 independent runs.

As seen from Table 7, if the data have a single view (the 20-Newsgroups data) in essential, the performance of SSDR is better than that of MVSSDR. Thus, MVSSDR does not always perform well except that the data are known to have multiple views. On the contrary, if different features of the data have relaxed relationships with each other (the Sonar data), MVSSDR performs better than SSDR.

The second question is that whether the adding of views would always improve the performance of MVSSDR. The answer is also 'not definitely'. Intuitively, if the data have more views, it would have more feature information. However, the added information may be redundant. It may be useless for dimensionality reduction, or badly worsen the performance of MVSSDR. For example, in clustering task, if the data are linearly separable in its previous views, the adding of other features, which cannot be separated linearly, would worsen the performance of MVSSDR.

We have done some experiments for illustration. They are performed on the 20-Newsgroups data set and the Internet advertisements data set with three views. We employ MVSSDR for dimensionality reduction and K-means for clustering. Similarly, all these results, i.e., Acc, are averaged over 50 independent trails. Correspondingly, the performances of these methods on the data with one view and two views are also listed in Tables 4 and 5.

Comparing the results shown in Table 8 with that in Tables 4 and 5, we can conclude that the adding of views does not always improve the performance of MVSSDR. For 20-Newsgroups data, the adding of the third view would increase the performance of MVSSDR only when the classes are C_2 and C_3. For the Internet advertisements data, the adding of another view degrades the performance of MVSSDR in all cases. We can find the same phenomena when we compare the results on data of one view with the data of two views. This means that if the adding view is not properly selected, it would not be helpful for our method.

Table 9
Acc results of MVSSDR and SSDR on Sonar data with different views.

	View1	View2	View3
SSDR	0.5707(0.0464)	0.5902(0.0419)	0.6055(0.0492)
	V_1&$V2$	V_1&V_3	V_2&V_3
MVSSDR	0.5798(0.0289)	0.5836(0.0286)	0.6127(0.0336)

Table 10
The normalized cut clustering accuracies (Acc) on several subsets of the 20-Newsgroups data set.

Methods	C_1&C_2 V_1&V_2	C_1&C_3 V_1&V_2	C_1&C_2 V_1&V_3	C_2&C_3 V_1&V_3	C_1&C_2 V_2&V_3
Original (all)	0.9096	0.9148	0.8995	0.9192	0.8563
Original (View1)	0.8571	0.8942	0.8368	0.8923	0.8189
Original (View2)	0.5918	0.5838	0.6421	0.6769	0.6474
PCA (all)	0.9396	0.9348	0.9295	0.9392	0.9000
LDA (all)	0.9133	0.9295	0.9184	0.9231	0.8947
CCA (all)	0.8980	0.8985	0.9211	0.8923	0.8811
SSDR (all)	0.6510	0.6152	0.5132	0.5128	0.5474
SSDR (View1)	0.9374	0.9036	0.9195	0.9282	0.8953
SSDR (View2)	0.6580	0.7843	0.8842	0.7026	0.7300
MVSSDR	**0.9638**	**0.9670**	**0.9553**	**0.9638**	**0.9211**

Table 11
The NMI results of MVSSDR on WebKB-II with different parameter K.

	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$
Cornell	0.4889	0.5064	0.5230	0.5132	0.4674
Wisconsin	0.4130	0.4358	0.4020	0.3708	0.3659
Washington	0.3826	0.4052	0.4176	0.3854	0.3747
Texas	0.3945	0.4059	0.4309	0.4174	0.4094

The third concern is how to select the useful views for dimensionality reduction. It is a common question for multiple view learning and a little similar to feature selection. Since feature selection is NP hard, we have not found a useful rule to determine which feature is important for multiple view dimensionality reduction.

We have also done some experiments on Sonar data set with three views for illustration. Table 9 listed all the Accs averaged over 50 independent trails.

As seen from Tables 4, 5 and 9, we have not found an effective rule for feature selection. We will continue our work on this problem.

The fourth question is how much does the previous results depend on the clustering method, i.e., K-means? Intuitively, since our method could integrate the domain knowledge and the discriminant information in low dimensional subspaces, MVSSDR should also perform the best when another clustering method is employed. For illustration, we have also done some experiments on the 20-Newsgroups data set with another clustering method: the normalized cut [20]. The results are listed in Table 10.

Comparing with the corresponding results in Table 4, MVSSDR also performs the best, even when we employ a different clustering approach.

The finally concern is to show whether the results are sensitive to the parameter K in K-means. We have also done some experiments on the WebKB-II data set with different K. The NMI results are shown in Table 11. It seems that the accuracies do not heavily depend on K.

6. Conclusions and future works

In this paper, a novel semi-supervised dimensionality reduction approach MVSSDR was proposed. To the best of our knowledge, this is the first approach to address this problem. It aims to derive the consensus pattern for multiple view data with domain knowledge.

In each view, we derive the embedding that preserves the intrinsic structure of the data as well as the must-link and cannot-link constraints. We also provide an iterative algorithm to solve this problem. The linear transformation matrix, which is used to approximate each view pattern by the consensus pattern, can be computed in a closed form. The convergence behavior and out-of-sample-extension of MVSSDR are also provided. SSDR can be regarded as a special case of our method. Empirical evaluation on different kinds of data clustering tasks shows that MVSSDR outperforms other approaches. Some important discussions about multiple view dimensionality reduction are also provided.

In our future work, we will focus on the theoretical analysis and accelerating issues of the MVSSDR algorithm. To deal with nonlinear problem, the kernel extension is also our future work.

Acknowledgments

The authors thank two anonymous reviewers for their constructive suggestions. Thanks to Daoqiang Zhang for providing data. Thanks also to the National Basic Research Program of China under Grant nos. 2005CB321800 and 2009CB320602, National Natural Science Foundation of China, under Grant no. 60673090, the Hunan Provincial Innovation Foundation for Postgraduate for their supports.

References

[1] S. Rüping, T. Scheffer, Learning with multiple views, in: ICML Workshop on Learning with Multiple Views, 2005.
[2] X. Zhu, Semi-supervised learning literature survey, Technical Report 1530, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, 2006.
[3] T. Mullen, R. Malouf, A preliminary investigation into sentiment analysis for informal political discourse, in: Proceedings of the AAAI Workshop on Analysis of Weblogs, 2006.
[4] A.M. Martinez, A.C. Kak, PCA versus LDA, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2) (2001) 228–233.
[5] L. Saul, S. Roweis, Think globally, fit locally: unsupervised learning of low dimensional manifolds, Journal of Machine Learning Research 4 (2003) 119–155.
[6] P. Glenisson, J. Mathys, B.D. Moor, Metaclustering of gene expression data and literature-based information, SIGKDD Explorations Newsletter 5 (2) (2003) 101–112.
[7] H. Hotelling, The most predictable criterion, Journal of Educational Psychology 26 (1935) 139–142.
[8] D. Foster, S. Kakade, T. Zhang, Multi-view dimensionality reduction via canonical correlation analysis, TTI-C Technical Report, TTI-TR-2008-4, 2008.
[9] S. Bickel, T. Scheffer, Multi-view clustering, in: ICDM 04, 2004, p. 1926.
[10] D. Zhou, C. Burges, Spectral clustering and transductive learning with multiple views, in: ICML07, 2007, pp. 1159–1166.
[11] B. Long, P.S. Yu, Z. Zhang, A general model for multiple view unsupervised learning. in: Proceedings of the 8th SIAM International Conference on Data Mining (SDM'08), Atlanta, Georgia, USA, 2008.
[12] X. Yang, H. Fu, H. Zha, J.L. Barlow, Semi-supervised nonlinear dimensionality reduction, in: ICML06, Pittsburgh, PA, 2006, pp. 1065–1072.
[13] D. Zhang, Z. Zhou, S. Chen, Semi-supervised dimensionality reduction, in: SIAM Conference on Data Mining (SDM), 2007.
[14] D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in: Proceedings of the International Conference on Computer Vision, 2007.
[15] A. Blum, T. Mitchell, Combining labeled and unlabeled data with cotraining, in: Annual Conference on Computational Learning Theory (COLT-98), 1998.
[16] J.S. Sobieski, Two alternative ways for solving the coordination problem in multilevel optimization, Structural and Multidisciplinary Optimization 6 (4) (1993) 205–215.
[17] D. Zhang, F. Wang, C. Zhang, T. Li, Multi-view local learning, in: AAAI'08, Chicago, Illinois, USA, 2008.
[18] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Dover, New York, 1998.
[19] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, Journal on Machine Learning Research (JMLR) 3 (2002) 583–617.
[20] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

About the Author—CHENPING HOU received his BS degree in applied mathematics from the National University of Defense Technology, Changsha, China, in 2004. He is now a PhD candidate in the Department of Mathematics and System Science. He worked as a Visiting Student at Tsinghua University since May 2008. His current research interests include dimensionality reduction, manifold learning, and semi-supervised learning.

About the Author—CHANGSHUI ZHANG received his BS degree in Mathematics from Peking University, China, in 1986, and PhD degree from Department of Automation, Tsinghua University, in 1992. He is currently a Professor of Department of Automation, Tsinghua University. He is an Associate Editor of the journal Pattern Recognition. His interests include artificial intelligence, image processing, pattern recognition, machine learning, evolutionary computation and complex system analysis.

About the Author—YI WU is a Professor in the Department of Mathematics and System Science at the National University of Defense Technology in Changsha, China. He earned his Bachelor's and Master's degrees in Applied Mathematics at the National University of Defense Technology in 1981 and 1988. He worked as a Visiting Researcher at New York State University in 1999. His research interests include applied mathematics, statistics, and data processing.

About the Author—FEIPING NIE received his BS degree from the Department of Computer Science, North China University of Water Conservancy and Electric Power, China, in 2000, and MS degree from the Department of Computer Science, Lanzhou University, China, in 2003. He is currently a PhD candidate in the Department of Automation, Tsinghua University. His research interests focus on machine learning and its applications.