Multidimensional Scaling on Multiple Input Distance Matrices

Song Bai[1], **Xiang Bai**[1*], **Longin Jan Latecki**[2], **Qi Tian**[3]
[1]Huazhong University of Science and Technology
[2]Temple University, [3]University of Texas at San Antonio
{songbai, xbai}@hust.edu.cn, latecki@temple.edu, qitian@cs.utsa.edu

Abstract

Multidimensional Scaling (MDS) is a classic technique that seeks vectorial representations for data points, given the pairwise distances between them. In recent years, data are usually collected from diverse sources or have multiple heterogeneous representations. However, how to do multidimensional scaling on multiple input distance matrices is still unsolved to our best knowledge.

In this paper, we first define this new task formally. Then, we propose a new algorithm called Multi-View Multidimensional Scaling (MVMDS) by considering each input distance matrix as one view. The proposed algorithm can learn the weights of views (*i.e.*, distance matrices) automatically by exploring the consensus information and complementary nature of views. Experimental results on synthetic as well as real datasets demonstrate the effectiveness of MVMDS. We hope that our work encourages a wider consideration in many domains where MDS is needed.

Introduction

Multidimensional scaling (MDS) (Borg and Groenen 2005; Torgerson 1958) is a fundamental and important technique with a wide range of applications to data visualization, artificial intelligence, network localization, robotics, cybernetics, social science, *etc*. For example, researchers in bioinformatics apply MDS to unravel relational patterns among genes (Taguchi and Oono 2005). As another example, MDS is also used by computer vision community (Bronstein et al. 2008). A typical application is to approximate geodesic distances of mesh points (Elad and Kimmel 2003) or planar points (Ling and Jacobs 2007) in Euclidean space so that the non-rigid intrinsic structure of shapes can be captured.

Given pairwise distances between N data points, MDS aims at projecting these data into P dimensional space, such that the between-object distances can be preserved as well as possible. In recent years, data are often collected from diverse domains or have various heterogeneous representations (Amid and Ukkonen 2015; Xu, Tao, and Xu 2013). That is to say, each data may have multiple views. For instance, an image can be described by multiple visual features, such as Scale Invariant Feature Transform (SIFT) (Lowe 2004), Histogram of Oriented Gradients (HOG) (Dalal and Triggs 2005), Local Binary Patterns (LBP) (Ojala, Pietikäinen, and Mäenpää 2002), *etc*. A web page can be described by the document text itself and the anchor text attached to its hyperlinks. It has been extensively demonstrated that a fusion of those multi-view representations by leveraging the interactions and complementarity between them is usually beneficial to obtain more faithful and accurate information.

In the past decades, numerous efforts have been devoted to the formulation, optimization and application of MDS (see (Borg and Groenen 2005; France and Carroll 2011) for a survey). However, the problem of multidimensional scaling on multiple input distance matrices has not been addressed. Nevertheless, this new topic has gradually become important in practical applications. Consider a toy example (also presented in experiments) where one wants to illustrate the relative positions of six cities in a planar map but he/she receives more than one distance matrix. How to project the six cities to $P = 2$ dimensional space given the multiple input matrices? Meanwhile, MDS can also act as a dimensionality reduction algorithm if the embedding dimension P is smaller than the input dimension. This arises another question, that is, how to conduct multi-view dimensionality reduction (Han et al. 2012; Xia et al. 2010; Foster, Kakade, and Zhang 2008) with the rapid growth of high dimensional data.

In this paper, we begin to investigate this new task, *i.e.*, multidimensional scaling on multiple input distance matrices. Our contributions can be divided into three folds:

1. We formally put forward the idea of performing MDS on multi-view data, and discuss the basic difficulties that need to be addressed carefully in this framework.

2. In addition to the novelty of our problem formulation, a new algorithm called Multi-View Multidimensional Scaling (MVMDS) is proposed to solve it. Inspired by (Xia et al. 2010) and its related works in multi-view learning (Sun 2013), a weight learning paradigm is imposed to attach more importance to discriminative views and suppress the negative influences of noisy views. Accordingly, an iterative solution is derived with proven convergence so that view weights can be updated automatically con-

*Corresponding author

trolled with only one parameter.

3. Extensive experimental evaluations on synthetic and real datasets manifest the effectiveness of the proposed method. Besides, we also give a comprehensive summary about promising future works that can be studied within this framework.

Task Definition

Given the pairwise distances $\delta = \{\delta_{ij}\}_{1\leq i,j\leq N}$ between N data points, Multidimensional Scaling (MDS) seeks for N configuration points $\mathcal{X} = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{N\times P}$ such that δ_{ij} can be well approximated by the Euclidean distance $d_{ij}(\mathcal{X}) = \|x_i - x_j\|_2$. The most widely-used definition of metric MDS is called "Stress", defined as

$$\min_{\mathcal{X}} \sum_{i<j} w_{ij} \left(\delta_{ij} - d_{ij}(\mathcal{X})\right)^2, \tag{1}$$

where w_{ij} are some pre-fixed weighting coefficients and P is the embedding dimension. In some specific situations, we have to deal with missing values, *i.e.*, δ_{ij} is not well defined. Therefore, one can set w_{ij} to 0 for those missing values, and set w_{ij} to 1 if δ_{ij} is known.

As discussed above, though numerous efforts have been devoted to its optimization and application, all the variants of MDS can only deal with single view data. Performing MDS in multi-view data has not been addressed. In this paper, we first give a formal definition of performing MDS on multi-view data. Given N abstract points $\mathcal{Y} = \{y_1, y_2, \ldots, y_N\}$ and their pairwise distances in M views $\delta^{(v)} \in \mathbb{R}^{N\times N}, 1 \leq v \leq M$, where our goal is to learn a function $\mathcal{F}$ defined as

$$\mathcal{F} : (\mathcal{Y}, \delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(M)}) \rightarrow \mathcal{X}, M > 1, \tag{2}$$

where $\mathcal{X} \in \mathbb{R}^{N\times P}$ is a configuration of N in P dimensional Euclidean space.

In this framework, some basic difficulties should be solved carefully: (1) The most fundamental issue is how to ensemble these distance matrices. A very naive solution is to use a linear combination of them. However, it is likely to achieve unsatisfactory results since informative and noisy views are all treated equally. Another possible solution is to apply co-training (Zhou and Li 2005) to MDS. Unfortunately, many co-training algorithms cannot guarantee the convergence. (2) How to judge the importance of different views? In most cases, MDS is defined as an unsupervised algorithm. It is problematic to determine the view weights automatically in an unsupervised manner, since no prior knowledge is available. (3) How to derive the optimal solution from $\mathcal{F}$ which guarantees to provide meaningful results?

Proposed Solution

To do multidimensional scaling on multiple input distance matrices, we propose a new objective function called Multi-View Multidimensional Scaling (MVMDS), formulated as

$$\begin{aligned} &\min_{\alpha^{(v)},\mathcal{X}} \sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} \sum_{i<j} w_{ij} \left(\delta_{ij}^{(v)} - d_{ij}(\mathcal{X})\right)^2, \\ &s.t. \sum_{v=1}^{M} \alpha^{(v)} = 1, 0 \leq \alpha^{(v)} \leq 1, \end{aligned} \tag{3}$$

where $\alpha^{(v)}$ measures the importance of v-th view, and the exponent $\gamma > 1$ is the weight controller that determines the distribution of $\alpha = \{\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(M)}\}$.

The weight learning mechanism is imposed by adding ${\alpha^{(v)}}^{\gamma}$ to the stress. The reason behind this choice is that if using $\alpha^{(v)}$ directly, the solution of α is that the view with the smallest stress value has the weight $\alpha^{(v)} = 1$ and all other views have $\alpha^{(v)} = 0$. This is not a good behavior since only one view is selected and the complementary nature among multiple views is ignored. The proposed adaptive weight learning paradigm is a primary advantage over the naive solution of using a weighted linear combination of multiple distance matrices, where it is nontrivial to determine the weights since at least $M - 1$ values should be specified. Hence the computational complexity is unbearable when $M > 2$.

Meanwhile, we only set a consensus embedding $\mathcal{X}$, instead of defining an individual embedding $\mathcal{X}^{(v)}$ for each view. It can be understood as minimizing disagreement of multiple views. Nevertheless, we force the embedding $\mathcal{X}$ to be the same across multiple views so that the aggregation of multiple embedding $\mathcal{X}^{(v)}$ is done implicitly. With this setting, one can easily identify the disagreement degree of different views and tune their weights via the weight learning paradigm.

Considering there are two types of variables to determine in Eq. (3): the configuration points $\mathcal{X}$ and the view weight $\alpha^{(v)}$, we adopt an alternative way to iteratively solve the above optimization problem. By doing so, we decompose it into two sub-problems.

I. Update $\mathcal{X}$ when α is fixed. In this situation, Eq. (3) is equivalent to the following optimization problem:

$$\min_{\mathcal{X}} \mathcal{J}_1 + \mathcal{J}_2 - 2\mathcal{J}_3, \tag{4}$$

where

$$\begin{aligned} \mathcal{J}_1 &= \sum_{v=1}^{M} \sum_{i<j} {\alpha^{(v)}}^{\gamma} w_{ij} {\delta_{ij}^{(v)}}^2, \\ \mathcal{J}_2 &= \sum_{v=1}^{M} \sum_{i<j} {\alpha^{(v)}}^{\gamma} w_{ij} d_{ij}^2(\mathcal{X}), \\ \mathcal{J}_3 &= \sum_{v=1}^{M} \sum_{i<j} {\alpha^{(v)}}^{\gamma} w_{ij} \delta_{ij}^{(v)} d_{ij}(\mathcal{X}). \end{aligned} \tag{5}$$

To optimize this sub-problem, we adopt majorization approach.

As can be drawn, the first term $\mathcal{J}_1$ in Eq. (4) is a constant. Thus it can be omitted in the procedure of optimization.

We now come to the second term in Eq. (4), which calculates a sum of the weighted squared distances on all views. We can derive that

$$\mathcal{J}_2 = trace(\mathcal{X}'\mathcal{V}\mathcal{X}), \tag{6}$$

where $\mathcal{V} \in \mathbb{R}^{N\times N}$ has elements

$$v_{ij} = \begin{cases} -\sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} w_{ij} & \text{if } i \neq j, \\ \sum_{j=1, j\neq i}^{N} \sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} w_{ij} & \text{if } i = j. \end{cases} \tag{7}$$

The last term in Eq. (4) computes a weighted sum of the distances on all views. Assume $\mathcal{Z}$ denotes the configuration points $\mathcal{X}$ in the previous iteration. According to Cauchy-Schwartz inequality $d_{ij}(\mathcal{X})d_{ij}(\mathcal{Z}) \geq \sum_{p=1}^{P}(x_{ip} - x_{jp})(z_{ip} - z_{jp})$ with equality if $\mathcal{Z} = \mathcal{X}$, we can obtain

$$\mathcal{J}_3 = \sum_{i<j} \left(\sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} w_{ij} \delta_{ij}^{(v)} \right) d_{ij}(\mathcal{X}) \geq trace(\mathcal{X}'\mathcal{B}\mathcal{Z}), \tag{8}$$

where $\mathcal{B} \in \mathbb{R}^{N\times N}$ has elements

$$\begin{aligned} b_{ij} &= \begin{cases} -\frac{\sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} w_{ij} \delta_{ij}^{(v)}}{d_{ij}(\mathcal{Z})} & \text{if } i \neq j \text{ and } d_{ij}(\mathcal{Z}) \neq 0 \\ 0 & \text{if } i \neq j \text{ and } d_{ij}(\mathcal{Z}) = 0 \end{cases} \\ b_{ii} &= -\sum_{j=1, j\neq i}^{N} b_{ij}, \end{aligned} \tag{9}$$

Based on the analysis above, the objective function in Eq. (4) is upper-bounded by

$$\mathcal{J} \leq \mathcal{J}_{\triangle} = \mathcal{J}_1 + trace(\mathcal{X}'\mathcal{V}\mathcal{X}) - 2trace(\mathcal{X}'\mathcal{B}\mathcal{Z}). \tag{10}$$

The partial derivative of $J_{\triangle}$ with regard to $\mathcal{X}$ is

$$\frac{\partial \mathcal{J}_{\triangle}}{\partial \mathcal{X}} = 2\mathcal{V}\mathcal{X} - 2\mathcal{B}\mathcal{Z}. \tag{11}$$

By setting Eq. (11) to zero, we have

$$\mathcal{X} = \mathcal{V}^{+}\mathcal{B}\mathcal{Z}, \tag{12}$$

where $\mathcal{V}^{+}$ is the Moore-Penrose inverse of $\mathcal{V}$. In usual cases, there are no missing values in the input distance matrix δ (*i.e.*, $\forall i, j, w_{ij} = 1$). Consequently, Eq. (12) can be simplified to

$$\mathcal{X} = \frac{1}{N\sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma}} \mathcal{B}\mathcal{Z}, \tag{13}$$

II. Update $\alpha^{(v)}$ when $\mathcal{X}$ is fixed. For the sake of notation convenience, we re-write the objective function in Eq. (3) as

$$\mathcal{J} = \sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} \mathcal{J}^{(v)}, \tag{14}$$

where $\mathcal{J}^{(v)} = \sum_{i<j} w_{ij} \left(\delta_{ij}^{(v)} - d_{ij}(\mathcal{X}) \right)^2$ denotes the counterpart of the v-th view. To get the optimal solution of this sub-problem, we utilize Lagrange Multiplier Method.

Taking the constraint $\sum_{v=1}^{M} \alpha^{(v)} = 1$ into consideration, the Lagrange function of $\mathcal{J}$ is

$$L(\mathcal{J}, \lambda) = \sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} \mathcal{J}^{(v)} + \lambda \left(\sum_{v=1}^{M} \alpha^{(v)} - 1 \right), \tag{15}$$

whose partial derivative with respect to $\alpha^{(v)}$ is

$$\frac{\partial L(\mathcal{J}, \lambda)}{\partial \alpha^{(v)}} = \gamma {\alpha^{(v)}}^{(\gamma-1)} \mathcal{J}^{(v)} - \lambda. \tag{16}$$

By setting Eq. (16) to zero, we have

$$\alpha^{(v)} = \left(\frac{\lambda}{\gamma \mathcal{J}^{(v)}} \right)^{\frac{1}{\gamma-1}}. \tag{17}$$

After substituting $\alpha^{(v)}$ in Eq. (17) into the constraint $\sum_{v=1}^{M} \alpha^{(v)} = 1$, the multiplier λ is eliminated and the optimal solution of $\alpha^{(v)}$ is obtained finally as

$$\alpha^{(v)} = \frac{\left(\mathcal{J}^{(v)} \right)^{\frac{1}{1-\gamma}}}{\sum_{v'=1}^{M} \left(\mathcal{J}^{(v')} \right)^{\frac{1}{1-\gamma}}}. \tag{18}$$

Note that Eq. (18) encounters "division by zero" when $\gamma = 1$. As discussed above, the optimal solution of α in this situation is

$$\alpha^{(v)} = \begin{cases} 1 & \text{if } v = \arg\min_{v'} \mathcal{J}^{(v')} \\ 0 & \text{otherwise}. \end{cases} \tag{19}$$

In the limit case $\gamma \to \infty$, we will get equal weights $\alpha^{(v)} = \frac{1}{M}$ for all the views (see also Fig. 2). As a result, only one parameter γ is used to control the weight distribution across multiple views in our algorithm. The optimal choice of γ depends the complementarity between the input matrices. If rich complementarity exists among views, large γ is preferred.

In summary, we present the whole algorithm in Algorithm 1. The convergence of the proposed algorithm is guaranteed. According to Alg. 1, when updating $\mathcal{X}$ in the $(t+1)$-th iteration, the objective value of Eq. (3) is decreased by the majorization algorithm compared with that of the t-th iteration. When updating $\alpha^{(v)}$, a global minimum is expected to generate the optimal solution based on Eq. (18). Therefore, by alternatively updating $\mathcal{X}$ and $\alpha^{(v)}$ in an iterative manner, the objective value keeps decreasing. Since Eq. (3) is lower-bounded by 0, convergence can be arrived given enough iterations.

Future Work

Many questions remain to be investigated further in this new task, for example:

Missing values. In the proposed solution, one can set $w_{ij} = 0$ to ignore missing values in the input distance matrices. Some clustering approaches (Wagstaff 2004) usually fill missing values by imputation. Since multiple input distance matrices are available here, maybe it is more effective if we can use the existing values in other views to predict the missing values in a certain view. It deserves a careful investigation in the future, since using the interactions among multiple views to predict missing values have not been exploited before to our best knowledge.

Algorithm 1: Multi-View Multidimensional Scaling.

Input:
$\delta^{(v)} \in \mathbb{R}^{N\times N}, 1 \le v \le M$: the input distance matrix;
P: the embedding dimension;
γ: the weight controller.
Output:
$\mathcal{X} \in \mathbb{R}^{N\times P}$: the configuration points;
begin
  Initialize $\alpha^{(v)} = \frac{1}{M}$;
  repeat
    Update $\mathcal{X}$ using Eq. (12) or Eq. (13);
    Update the weights $\alpha^{(v)}$ using Eq. (18);
    Update $\mathcal{Z} = \mathcal{X}$;
  until *convergence*
  return $\mathcal{X}$

Intrinsic dimension. For the sake of data visualization, the embedding dimension of MDS is usually $P = 2$ or $P = 3$. In a general situation, P should be specified by the users. Some studies (Levina and Bickel 2004) aim at learning an estimator of intrinsic dimension that can sufficiently describe the data distribution. In this paper, different input distance matrices tend to have different intrinsic dimensions. Therefore, the optimal embedding dimension should not only be "intrinsic", but also "consensus", *i.e.*, shared by multiple views. It is probably a data-driven problem. Nevertheless, it is still worthy studying.

Parameter-free. Despite the embedding dimension P, standard MDS can be deemed as a parameter-free algorithm. When dealing with multiple input distance matrices, the solution given in this paper introduces an additional parameter γ to tune their weight distribution. In our experiments, γ has to be specified manually or determined by cross validation. It remains an open issue for researchers to design parameter-free algorithms which can fit into various applications.

Applications. MDS has a wide range of applications in many domains (Lin et al. 2016; Lindenbaum et al. 2015). These applications can be mostly reconsidered in this newly-defined framework. For example, MDS can be used to draw perceptual maps (Bijmolt and Wedel 1999) in marketing, where each brand has thousands of attributes. Traditionally in MDS, these attributes are treated equally. While with the solution given in this framework (*e.g.*, MVMDS proposed in this paper), the importance of these attributes can be identified simultaneously. In robots localization (Jenkins and Matarić 2004), distances between items are usually captured by multiple sensors and multiple time periods. It is badly required to do MDS on multiple distance matrices. These practical applications can be further investigated by researchers in specific domains.

Experiments

MDS usually acts as a fundamental tool for preprocessing (Ling and Jacobs 2007) or visualization (Buja et al. 2008). For a long time, the only principled way to evaluate the effectiveness of MDS-related algorithms is to compare the stress value defined in Eq. (1). However, it is not applicable in this paper, owing to the use of multiple groundtruth distances. In this section, we first demonstrate the effectiveness of MVMDS using a synthetic example where multiple views are imitated from a unique groundtruth. Thus, it becomes feasible to compare the stress value using Eq. (1). Then following (Han et al. 2012), we assess the discriminative power of the embedding $\mathcal{X}$ obtain by MVMDS on three image datasets in the applications of retrieval and clustering.

Since the weight controller γ needs to be determined empirically, we conduct an exhaustive search in the interval $(1, 10]$ with step size 0.5 to find its optimal value.

	LA	SFO	CHI	HOU	NY	WC
LA	0	-	-	-	-	-
SFO	380	0	-	-	-	-
CHI	2034	2148	0	-	-	-
HOU	1566	1945	1085	0	-	-
NY	2824	2946	821	1653	0	-
WC	2689	2840	715	1414	237	0

Table 1: The pairwise distances among six cities in the USA.

Synthetic Example

We consider a synthetic example where 4 participants are asked to estimate the distances between six cities in the USA, including Los Angeles (LA), San Francisco (SFO), Houston (HOU), Washington D.C. (WC), Chicago (CHI) and New York (NY). Table 1 gives the true pairwise distances among them. Due to the differences in skill and character, different participants generate different estimating results, serving as multiple views. Specially, more professional and careful participants are more likely to attain faithful results.

The procedure of generating multiple view input is as follow. To generate the v-th view, we first randomly select K pairs of city distances δ_{ij}. Then for each δ_{ij}, Gaussian noise with mean δ_{ij} and standard derivation $\sigma \cdot \delta_{ij}$ is added. Finally, 4 views are generated and Table 2 lists the values of K and σ. As we can see, View 1 imitates the most proficient and careful participant, since it has the fewest perturbed distance pairs and the smallest derivation. By contrast, View 4 is the most unskilled and careless participant with lots of mistakes during estimating the distances.

Fig. 1(a) to Fig. 1(d) give the relative positions of the six cities, marked in orange points, in $P = 2$ dimensional space by applying MDS to each view. The result of a linear combination of the 4 views with equal weights, denoted as LC_MDS, is presented in Fig. 1(e), and the results of the proposed MVMDS with different γ are presented in Fig. 1(f) to Fig. 1(h). We apply MDS to the distance matrix given in Table 1 to produce the groundtruth, marked in gray color in Fig. 1. As can be drawn from the figure, MVMDS can yield near perfect results.

Moreover, since the true distance matrix is accessible in Table 1, we can directly compute the stress values of different methods using Eq. (1). The quantitative comparison of

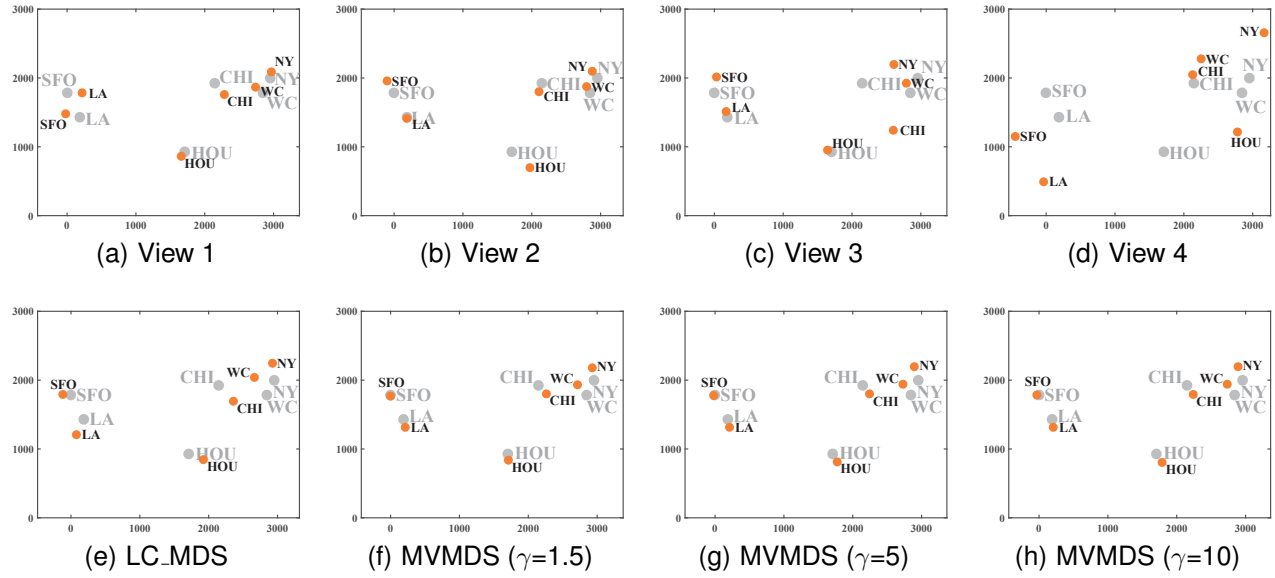

Figure 1: The illustration of relative positions of the six cities. The gray points are obtained by applying MDS to the true distance matrix shown in Table 1, serving as groundtruth. The orange points are obtained by applying MDS or MVMDS to simulated views.

	K	σ	Stress value ($\times 10^5$)
View 1	4	0.3	2.18
View 2	4	0.7	4.40
View 3	8	0.3	7.50
View 4	8	0.7	74.11
LC_MDS	-	-	6.15
MVMDS (γ=1.5)	-	-	1.61
MVMDS (γ=5)	-	-	**1.35**
MVMDS (γ=10)	-	-	1.36

Table 2: The parameter setup to generate multi-view input and the comparison of stress.

stress values is listed in Table 2. As we can see, the stress of MVMDS is not only lower than each single view but also lower than LC_MDS. The reason behind the superiority of MVMDS is the weight learning mechanism imposed on multiple views. To support our claim more clearly, we plot the learned weight α as a function of γ in Fig. 2. It suggests that in all the cases, MVMDS can give prominence to View 1 which is the most reliable participant. When $\gamma < 1.5$, the influence of View 4 is eliminated totally. When $\gamma > 35$, we will get equal weights for all the views.

Fig. 3 presents the curve of convergence of MVMDS, which testifies its convergence property experimentally. It is also observed that MVMDS converges quickly within less than 10 iterations.

Image Retrieval

Two image benchmark datasets, *i.e.*, Microsoft Research Cambridge Volume 1 (MSRC-v1) (Winn and Jojic 2005),

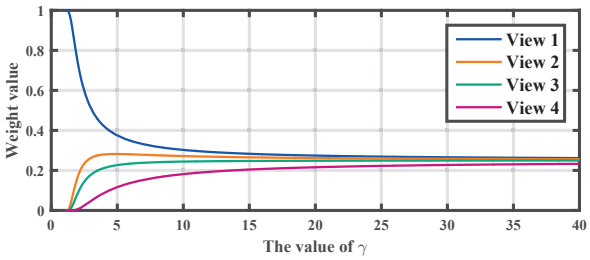

Figure 2: The learned weight of different views.

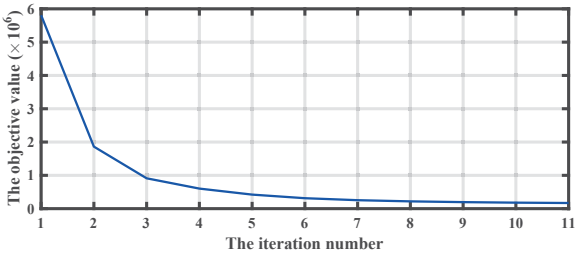

Figure 3: The curve of convergence.

Caltech-101 dataset (Fei-Fei, Fergus, and Perona 2007), are selected for performance comparisons. The details of those datasets are listed below:

1. MSRC-v1: it is a scene image dataset composed of 240 images and 9 categories. Following (Lee and Grauman 2009), 7 categories (tree, building, airplane, cow, face, car, bicycle) are used with 30 images per category.
2. Caltech-101: it consists of 101 object categories, with

Methods	MSRC-v1 dataset				Caltech101-7 dataset				Caltech101-20 dataset			
	NN	FT	ST	DCG	NN	FT	ST	DCG	NN	FT	ST	DCG
SIFT	0.719	0.465	0.673	0.783	0.716	0.432	0.633	0.794	0.261	0.169	0.283	0.577
HOG	0.728	0.481	0.681	0.790	0.788	0.515	0.700	0.832	0.417	0.261	0.384	0.636
LBP	0.733	0.477	0.681	0.793	0.671	0.446	0.604	0.779	0.354	0.222	0.344	0.609
HSV	0.518	0.307	0.489	0.675	0.446	0.283	0.479	0.683	0.286	0.202	0.305	0.585
GIST	0.742	0.460	0.663	0.786	0.721	0.496	0.677	0.810	0.425	0.262	0.373	0.636
LC_MDS	0.796	0.501	0.705	0.816	0.711	0.460	0.651	0.797	0.304	0.216	0.314	0.594
MVMDS	**0.806**	**0.530**	**0.728**	**0.827**	**0.805**	**0.554**	**0.734**	**0.847**	**0.429**	**0.267**	**0.392**	**0.641**

Table 3: The retrieval performance comparison on MSRC-v1 dataset, Caltech101-7 dataset and Caltech101-20 dataset.

Methods	MSRC-v1 dataset			Caltech101-7 dataset			Caltech101-20 dataset		
	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity
SIFT	61.7±3.04	52.8±2.87	63.8±2.85	55.9±2.29	45.6±2.75	62.0±2.19	24.4±1.67	25.6±1.93	29.4±1.87
HOG	63.6±1.72	57.0±2.08	65.6±1.62	63.5±1.40	52.5±1.82	68.9±1.27	37.4±1.56	36.4±1.58	41.1±1.58
LBP	64.3±1.37	57.2±1.12	66.7±1.23	56.5±1.24	43.3±1.23	63.4±1.03	33.0±1.20	33.1±1.11	38.9±1.23
HSV	42.3±1.50	30.7±1.71	44.6±1.46	32.8±0.93	16.8±0.65	42.6±0.69	26.2±0.70	26.5±0.49	30.7±0.63
GIST	60.2±1.74	52.8±2.02	63.1±1.69	59.2±1.28	48.2±1.09	63.4±0.98	37.6±1.12	36.4±0.86	41.0±1.10
LC_MDS	70.3±2.82	61.9±3.61	72.4±2.81	56.7±2.68	44.8±2.98	63.2±2.44	27.9±1.80	28.4±1.84	32.9±1.72
MVMDS	**71.9±1.82**	**64.9±2.68**	**73.8±1.89**	**71.5±1.64**	**63.0±1.97**	**76.1±1.32**	**38.5±1.86**	**37.6±1.59**	**42.3±1.74**

Table 4: The clustering performance comparison (%) on MSRC-v1 dataset, Caltech101-7 dataset and Caltech101-20 dataset.

31 to 800 images per category. Following (Dueck and Frey 2007), we select 7 classes and 20 classes forming Caltech101-7 and Caltech101-20 respectively.

We extract 5 visual features to obtain the multi-view representations for each image, *i.e.*, Scale Invariant Feature Transform (SIFT) (Lowe 2004) with dimension 128, Histogram of Oriented Gradients (HOG) (Dalal and Triggs 2005) with dimension 775, Local Binary Patterns (LBP) (Ojala, Pietikäinen, and Mäenpää 2002) with dimension 1450, HSV color histogram with dimension 1000, GIST (Oliva and Torralba 2001) with dimension 512. All the visual features are L_2 normalized, then Euclidean distance is used to measure the dissimilarity between images. To get a comprehensive quantitative evaluation, we adopt four widely-used metrics in information retrieval, *i.e.*, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST) and Discounted Cumulative Gain (DCG). All the metrics range from 0 to 1 and larger values indicate better performances. Please refer to (Shilane et al. 2004) for their detailed definitions.

We compare the proposed MVMDS against 6 methods, including 5 single view counterparts and LC_MDS. All the comparisons are done by using MDS or the proposed MVMDS to project images into $P = 10$ dimensional space. Table 3 presents the experimental results on all the datasets. The table shows that our proposed MVMDS achieves the best performances consistently in all the evaluation metrics. One can also find that using a linear combination of all the views is not always useful. For example, the performances of LC_MDS are much lower than those of HOG on Caltech101-7 dataset and Caltech101-20 dataset. Our interpretation is that the baseline performances of most views (*e.g.*, SIFT, LBP and HSV) are poor, and they will deprive the discriminative power of informative views (*e.g.*, HOG and GIST) by simply stacking them with equal weights. By contrast, the proposed MVMDS benefits from the weight learning paradigm, thus decreasing the weights of less information views and suppressing their negative effects to a certain extent.

Image Clustering

In this section, we evaluate the performances of MVMDS in clustering task to obtain a more thorough analysis. We also extract 5 visual features and project all images into $P = 10$ dimensional space. Then K-means is applied to divide the images into clusters. The desired number of clusters is set to be equal to the natural number of categories in each dataset. For performance evaluation, we adopt three widely-used evaluation metrics, that is, Clustering Accuracy (ACC), Normalized Mutual Information (NMI) and Purity.

The comparison is presented in Table 4. Consistent to the experimental results above, MVMDS outperforms all the compared methods by a large margin. Especially on Caltech101-7 dataset, MVMDS outperforms the best-performing single view (HOG) by 7.97% in ACC, 10.51% in NMI, 7.18% in Purity and LC_MDS by 14.76% in ACC, 18.16% in NMI, 12.87% in Purity respectively.

We also compare with other multi-view learning algorithms, though they are not MDS-based. For example, the performance of MVMDS is better than Robust Multi-view K-means Clustering (RMKMC) (Cai, Nie, and Huang 2013), which reports ACC 67.9, NMI 68.9 and Purity 75.9. The performance gain is especially valuable when considering that the feature dimension used by MVMDS is only $P = 10$, significantly shorter than 2346 dimensional feature used in RMKMC.

Conclusion

In this paper, we focus on a new problem, that is, performing Multidimensional Scaling (MDS) on multi-view data. To address this issue, we propose a new algorithm called Multi-View Multidimensional Scaling (MVMDS), which is optimized in an iterative manner with guaranteed convergence. The proposed method can do discriminative view selection adaptively, thus the contributions of informative views are amplified. As introduced above, there are many interesting problems and applications for following researchers to think deeply in the future.

Acknowledgements

This work was supported in part by NSFC 61573160, NSFC 61429201, NSF IIS-1302164 and China Scholarship Council; and in part to Dr. Qi Tian by ARO grants W911NF-15-1-0290 and Faculty Research Gift Awards by NEC Laboratories of America and Blippar.

References

Amid, E., and Ukkonen, A. 2015. Multiview triplet embedding: Learning attributes in multiple maps. In *ICML*, 1472–1480.

Bijmolt, T. H., and Wedel, M. 1999. A comparison of multidimensional scaling methods for perceptual mapping. *Journal of Marketing Research* 277–285.

Borg, I., and Groenen, P. J. 2005. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.

Bronstein, A. M.; Bronstein, M. M.; Bruckstein, A. M.; and Kimmel, R. 2008. Analysis of two-dimensional non-rigid shapes. *IJCV* 78(1):67–88.

Buja, A.; Swayne, D. F.; Littman, M. L.; Dean, N.; Hofmann, H.; and Chen, L. 2008. Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics* 17(2):444–472.

Cai, X.; Nie, F.; and Huang, H. 2013. Multi-view k-means clustering on big data. In *IJCAI*.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *CVPR*, 886–893.

Dueck, D., and Frey, B. J. 2007. Non-metric affinity propagation for unsupervised image categorization. In *ICCV*, 1–8.

Elad, A., and Kimmel, R. 2003. On bending invariant signatures for surfaces. *TPAMI* 25(10):1285–1295.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVIU* 106(1):59–70.

Foster, D. P.; Kakade, S. M.; and Zhang, T. 2008. Multi-view dimensionality reduction via canonical correlation analysis. In *Technical Report*.

France, S. L., and Carroll, J. D. 2011. Two-way multidimensional scaling: A review. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 41(5):644–661.

Han, Y.; Wu, F.; Tao, D.; Shao, J.; Zhuang, Y.; and Jiang, J. 2012. Sparse unsupervised dimensionality reduction for multiple view data. *IEEE Trans. Circuits Syst. Video Techn* 22(10):1485–1496.

Jenkins, O. C., and Matarić, M. J. 2004. A spatio-temporal extension to isomap nonlinear dimension reduction. In *ICML*, 56.

Lee, Y. J., and Grauman, K. 2009. Foreground focus: Unsupervised learning from partially matching images. *IJCV* 85(2):143–166.

Levina, E., and Bickel, P. J. 2004. Maximum likelihood estimation of intrinsic dimension. In *NIPS*, 777–784.

Lin, G.; Fan, G.; Kang, X.; Zhang, E.; and Yu, L. 2016. Heterogeneous feature structure fusion for classification. *Pattern Recognition* 53:1–11.

Lindenbaum, O.; Yeredor, A.; Salhov, M.; and Averbuch, A. 2015. Multiview diffusion maps. *arXiv preprint arXiv:1508.05550*.

Ling, H., and Jacobs, D. W. 2007. Shape classification using the inner-distance. *TPAMI* 29(2):286–299.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *IJCV* 60(2):91–110.

Ojala, T.; Pietikäinen, M.; and Mäenpää, T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI* 24(7):971–987.

Oliva, A., and Torralba, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV* 42(3):145–175.

Shilane, P.; Min, P.; Kazhdan, M. M.; and Funkhouser, T. A. 2004. The princeton shape benchmark. In *SMI*.

Sun, S. 2013. A survey of multi-view machine learning. *Neural Computing and Applications* 23(7-8):2031–2038.

Taguchi, Y.-h., and Oono, Y. 2005. Relational patterns of gene expression via non-metric multidimensional scaling analysis. *Bioinformatics* 21(6):730–740.

Torgerson, W. S. 1958. Theory and methods of scaling.

Wagstaff, K. 2004. *Clustering with missing values: No imputation required*. Springer.

Winn, J. M., and Jojic, N. 2005. LOCUS: learning object classes with unsupervised segmentation. In *ICCV*, 756–763.

Xia, T.; Tao, D.; Mei, T.; and Zhang, Y. 2010. Multiview spectral embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 40(6):1438–1446.

Xu, C.; Tao, D.; and Xu, C. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*.

Zhou, Z.-H., and Li, M. 2005. Semi-supervised regression with co-training. In *IJCAI*, volume 5, 908–913.