Multi-Task Multi-View Clustering for Non-Negative Data

Xianchao Zhang and **Xiaotong Zhang** and **Han Liu**
School of Software
Dalian University of Technology
Dalian 116620, China
xczhang@dlut.edu.cn, zxt.dut@hotmail.com, liu.han.dut@gmail.com

Abstract

Multi-task clustering and multi-view clustering have severally found wide applications and received much attention in recent years. Nevertheless, there are many clustering problems that involve both multi-task clustering and multi-view clustering, i.e., the tasks are closely related and each task can be analyzed from multiple views. In this paper, for non-negative data (e.g., documents), we introduce a multi-task multi-view clustering (MTMVC) framework which integrates within-view-task clustering, multi-view relationship learning and multi-task relationship learning. We then propose a specific algorithm to optimize the MTMVC framework. Experimental results show the superiority of the proposed algorithm over either multi-task clustering algorithms or multi-view clustering algorithms for multi-task clustering of multi-view data.

1 Introduction

Multi-task clustering improves individual clustering performance by learning the relationship among related tasks. Multi-view clustering makes use of the consistency among different views to achieve better performance. Both multi-task clustering and multi-view clustering have severally found wide applications and received much attention in recent years. Nevertheless, there are many practical problems that involve both multi-task clustering and multi-view clustering, i.e., the tasks are closely related and each task can be analyzed from multiple views. For example, the tasks for clustering the web pages from four universities are four related tasks. The four tasks all have word features in the main texts, they also have many other features, such as the words in the hyperlinks pointing to the web pages, and the words in the titles of the web pages. For another example, the tasks for clustering the web images collected from Chinese web sites and English web sites are two related tasks. The two tasks both have visual features in the images, they also have word features in the surrounding texts in Chinese and English respectively. To tackle the clustering problem of such data sets, existing algorithms can only utilize limited information, i.e., multi-view clustering algorithms only use the information of the views in a single task, multi-task clustering algorithms only exploit the mutual information shared by all the related tasks from a single view. However, we can get better performance if both the multi-task and multi-view information could be utilized.

Recently, multi-task multi-view learning algorithms, which learn multiple related tasks with multi-view data, have been proposed. The graph-based framework in [He and Lawrence, 2011] takes full advantages of both the feature heterogeneity and task heterogeneity. Within each task, the consistency among different views is obtained by requiring them to produce the same classification function, and across different tasks, the relationship is established by utilizing the similarity constraint on the common views. The inductive learning framework in [Zhang and Huan, 2012] uses co-regularization and task relationship learning, which increases the practicality of multi-task multi-view learning. These methods have demonstrated their superiorities over either multi-task or multi-view learning algorithms. However, they all tackle classification. To the best of our knowledge, there is no existing approach to the multi-task multi-view clustering problem.

In this paper, we aim to deal with the multi-task multi-view clustering of non-negative data, which arises in many applications, such as various types of documents. Based on the observation that the related tasks have both common views and task specific views, we propose a bipartite graph based multi-task multi-view clustering (MTMVC) framework, which consists of three parts. (1) Within-view-task clustering: this part clusters the data of each view in each task. It is the base of the framework and mutually boosts with the other two parts. (2) Multi-view relationship learning: this part uses the consistency among different views to improve the clustering performance. (3) Multi-task relationship learning: this part learns the relationship among related tasks to improve the clustering performance. We integrate the three parts into one objective function and optimize it with a gradient ascent method. Because of the unitary constraints, we further solve the optimization problem by mapping the variables to the Stiefel manifold [Manton, 2002]. Experimental results on several real data sets show the superiority of the proposed algorithm over either multi-task clustering algorithms or multi-view clustering algorithms for multi-task clustering of multi-view data.

2 Related Work

***Multi-task Clustering*:** Multi-task clustering improves the clustering performance by learning the information shared among multiple related tasks. The approach in [Gu and Zhou, 2009] learns a subspace shared by the related tasks. The method in [Gu *et al.*, 2011] handles the multi-task clustering problem by learning a kernel. Unlike the above methods which focus on cross-domain multi-task clustering, some multi-task clustering methods are proposed to deal with the case of the related tasks from a same distribution or similar distributions [Zhang and Zhang, 2010; 2013].

***Multi-view Clustering*:** Multi-view clustering achieves better performance by using the consistency among different views. The method in [de Sa, 2005] is based on the minimizing-disagreement idea. The method in [Kumar *et al.*, 2011] co-regularizes the clustering hypotheses across views. The method in [Kumar and Daumé, 2011] applies the idea of co-training, which uses the spectral embedding from one view to constrain the similarity graph used for the other view.

***Multi-task Multi-view Learning*:** Multi-task multi-view learning deals with the learning problem of multiple related tasks with multiple views. There are mainly two algorithms proposed recently [He and Lawrence, 2011; Zhang and Huan, 2012]. As far as we know, there is no existing approach to the multi-task multi-view clustering problem.

***Co-clustering*:** Co-clustering has received a lot of attention in several practical applications such as text mining [Dhillon, 2001], genes [Cho *et al.*, 2004] and recommender systems [George and Merugu, 2005]. Co-clustering clusters the samples and features simultaneously so that the clustering performance of samples can be improved by the clustering of features, and vice versa [Banerjee *et al.*, 2007].

3 MTMVC Framework

3.1 Problem Formulation

We are given T clustering tasks, each with V_t views, i.e., $X_t^{(v)} = \{x_1^{(v)}, x_2^{(v)}, ..., x_{n_t}^{(v)}\} \in R^{d_t^{(v)} \times n_t}$, $t = 1, \ldots, T$, $v = 1, \ldots, V_t$, where n_t is the number of samples in the t-th task, $d_t^{(v)}$ is the feature number of the v-th view in task t. Each task t is to be partitioned into c_t clusters. In multi-task multi-view applications, the related tasks can be analyzed from multiple views. It can be observed that from some views, the related tasks share a lot of features [He and Lawrence, 2011], we call such views common views, and call the other views task specific views. S is the index collection of common views. T_v is the index collection of tasks under the common view v. The common view v consists of the features in the tasks belonging to T_v under view v. We assume the related tasks share at least one common view, and the number of clusters in each task is the same, i.e., $c_1 = c_2 = \ldots = c_T = c$. The first c eigenvectors represent the eigenvectors corresponding to the c largest eigenvalues.

3.2 Framework Overview

Based on the characteristics of the multi-task multi-view applications, we integrate the features in the common view of each task to link the related tasks together, and propose a bipartite graph based MTMVC framework (Figure 1), which consists of three components: within-view-task clustering, multi-view relationship learning and multi-task relationship learning. For within-view-task clustering, we construct a bipartite graph for each view of each task, and apply the bipartite graph co-clustering (BiCo) algorithm [Dhillon, 2001]. For multi-view relationship learning, we minimize the disagreement between the clustering of samples under each pair of views in each task. For multi-task relationship learning, we construct a bipartite graph between the samples and the shared features in the common view for each task, and perform the BiCo method to learn a shared subspace among the related tasks under each common view.

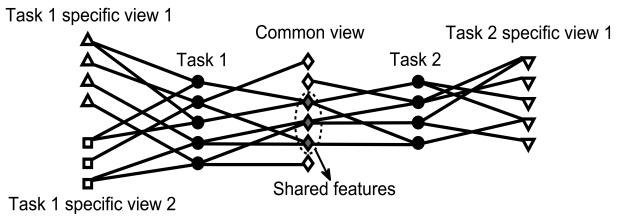

Figure 1: The MTMVC framework. The samples (black circles) of task 1 and task 2 have a common view which consists of task shared features (gray-filled diamonds) and task specific features (hollow diamonds). Task 1 also has two task specific views (upper triangles and squares), task 2 has one task specific view (lower triangles).

3.3 Objective Function

Within-view-task Clustering

This component clusters the data in each view of each task. It accomplishes the essential task of the whole algorithm and mutually boosts with the other two components. It also ensures the preservation of the knowledge available locally at each view of each task to avoid negative transfer [Pan and Yang, 2010]. We use BiCo, which clusters the samples and features through a bipartite graph, and the two phases boost each other. Given a data set $X \in R^{d \times n}$, we have $W = \begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix}$ and $D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$, where W is the matrix of bipartite graph, D is the degree matrix, $D_1(i,i) = \sum_j X_{ij}$, $D_2(j,j) = \sum_i X_{ij}$. The objective function of BiCo can be expressed as

$$\min_Z tr(Z^T L Z) \quad s.t. U^T U = I, M^T M = I \qquad (1)$$

where $L = D^{-1/2}(D - W)D^{-1/2}$, $Z = [U; M], U \in R^{d \times c}$ is composed of the first c eigenvectors of features, which indicates a partitioning of features, $M \in R^{n \times c}$ is composed of the first c eigenvectors of samples, which indicates a partitioning of samples. Eq.(1) can also be expressed as

$$\max_{U,M} tr(U^T A M) \quad s.t. U^T U = I, M^T M = I \qquad (2)$$

where $A = D_1^{-1/2} X D_2^{-1/2}$.

Multi-view Relationship Learning

To meet the requirement of consistency among different views within each task, we need a way to compare the eigenvectors of samples under view v and view q in each task t, say $M_t^{(v)}$ and $M_t^{(q)}$. Since $M_t^{(v)}$ and $M_t^{(q)}$ do not represent the true clustering result, we cannot compute the disagreement between them directly by squared Euclidean distance. But we can compute the similarity matrices for the partitioning of samples under view v and view q, say $Sim_{M_t^{(v)}}$ and $Sim_{M_t^{(q)}}$, then compute the disagreement between them by squared Euclidean distance as follows.

$$\begin{aligned} Dis(M_t^{(v)}, M_t^{(q)}) &= \left\| Sim_{M_t^{(v)}} - Sim_{M_t^{(q)}} \right\|_F^2 \\ s.t. M_t^{(v)T} M_t^{(v)} &= I, M_t^{(q)T} M_t^{(q)} = I \end{aligned} \tag{3}$$

Considering the feasibility of optimization, we use the common measure inner product to compute the similarity matrix, and get $Sim_{M_t^{(v)}} = M_t^{(v)} M_t^{(v)T}$. Under the constraint in Eq.(3), minimizing Eq.(3) can be simplified as

$$\begin{aligned} &\max_{M_t^{(v)}, M_t^{(q)}} tr(M_t^{(v)} M_t^{(v)T} M_t^{(q)} M_t^{(q)T}) \\ &s.t. M_t^{(v)T} M_t^{(v)} = I, M_t^{(q)T} M_t^{(q)} = I \end{aligned} \tag{4}$$

Multi-task Relationship Learning

To establish the relationship among related tasks, we hope to learn a subspace shared by the related tasks under the common view, in which the clustering result of each task is considered as the same as that in the original feature space. The shared subspace can be seen as a new feature space, in which the distributions of related tasks are close to each other. More specifically, if there are some tasks related to each other, there may exist some common latent features that cause the distributions of the related tasks to be close to each other [Gu and Zhou, 2009]. Therefore the task information can be transferred through the shared subspace. Based on the analysis above, we construct a bipartite graph between the samples and the shared features in the common view for each task, and perform the BiCo method to learn a shared subspace from the shared features in the common view. In BiCo, the eigenvectors of features in the v-th view of the t-th task $U_t^{(v)}$ can be thought of as dimensionality reduction with a linear combination of associated features, which can also be referred to as subspace basis [Vidal, 2011]. Considering there is a subspace $U^{(v)}$ shared by the related tasks under the common view v, as the BiCo method contains a component $U_t^{(v)}$ which can be seen as a subspace, and the eigenvectors of features $U_t^{(v)}$ can be boosted by the eigenvectors of samples $M_t^{(v)}$, thus we use the BiCo method to learn the shared subspace $U^{(v)}$ which can be boosted by $M_t^{(v)}$ in all the related tasks.

$$\max_{U^{(v)}} \sum_{t \in T_v} tr(U^{(v)T} \tilde{A}_t^{(v)} M_t^{(v)}) \; s.t. U^{(v)T} U^{(v)} = I \tag{5}$$

where $\tilde{A}_t^{(v)} = D_1^{-1/2} \tilde{X}_t^{(v)} D_2^{-1/2}$, $\tilde{X}_t^{(v)}$ is a data matrix between the samples in task t and the features shared by the tasks in the common view v.

The Overall Objective Function

We integrate within-view-task clustering, multi-view relationship learning and multi-task relationship learning into the overall objective function as follows.

$$\begin{aligned} &\max_{U_t^{(v)}, U^{(v)}, M_t^{(v)}} \sum_{t=1}^{T} (\sum_{v=1}^{V_t} J_1 + \lambda \sum_{v=1}^{V_t} \sum_{q \neq v}^{V_t} J_2) + \mu \sum_{v \in S} \sum_{t \in T_v} J_3 \\ &s.t. U_t^{(v)T} U_t^{(v)} = I(t = 1, \ldots, T, v = 1, \ldots, V_t) \\ &U^{(v)T} U^{(v)} = I(v \; is \; the \; index \; of \; common \; view) \\ &M_t^{(v)T} M_t^{(v)} = I(t = 1, \ldots, T, v = 1, \ldots, V_t) \end{aligned} \tag{6}$$

where $J_1 = tr(U_t^{(v)T} A_t^{(v)} M_t^{(v)})$, $J_2 = tr(M_t^{(v)} M_t^{(v)T} M_t^{(q)} M_t^{(q)T})$ and $J_3 = tr(U^{(v)T} \tilde{A}_t^{(v)} M_t^{(v)})$.

In Eq.(6), $\sum_{v=1}^{V_t} J_1$ is to co-cluster samples and features of all the views in each task t, $\sum_{v=1}^{V_t} \sum_{q \neq v}^{V_t} J_2$ is to maximize the agreement between the cluster assignments of any two different views in each task t, $\sum_{v \in S} \sum_{t \in T_v} J_3$ is to learn the shared subspace under each common view. $\lambda \geq 0, \mu \geq 0$ are parameters.

Discussion

Note that the objective of multi-task multi-view clustering is very complicated since different missions are involved. We divide the problem into three parts to make it easier to solve. There are other ways to combine the components. However, our way of using components in similar forms and linear combination leads to a simple solution of the problem.

We use the BiCo method for the first component, since it can boost $M_t^{(v)}$ by $U_t^{(v)}$ and vice versa, which can further promote multi-view and multi-task relationship learning. Specifically, during multi-view relationship learning, $M_t^{(v)}$ can take advantage of $M_t^{(q)} (q \neq v)$ because the boosted $M_t^{(q)}$ by BiCo in the first component can indicate the partition of samples more accurately. During multi-task relationship learning, the shared subspace $U^{(v)}$ can be boosted by the $M_t^{(v)}$ which has been improved by BiCo in the first component. However, despite these advantages, the use of BiCo limits our framework to work only for non-negative data.

The BiCo methods used in the first component and the third component play different roles. The BiCo method in the first component is to establish associations between the samples and any view within each task, and is essential to the multi-view relationship learning. The BiCo method in the third component is to learn the shared subspace among the related tasks, as BiCo can cluster features besides clustering samples, and the clusters of features (the eigenvectors of features) can be seen as subspace basis [Vidal, 2011].

Algorithm 1 MTMVC

Input: T tasks, each with V_t views $\{X_t^{(v)}\}_{t=1}^T$, $v = 1, \ldots, V_t$. The index collection of common views S. The index collection of tasks under the common view v T_v. Cluster number c. $\lambda \geq 0$, $\mu \geq 0$, the step length γ.
Output: Partitions $\{P^{(t)}\}_{t=1}^T$.
Initialization: Compute $A_t^{(v)}$ in Eq.(2). $U_t^{(v)}$ is formed by the first c left singular vectors of $A_t^{(v)}$, and $M_t^{(v)}$ is formed by the first c right singular vectors of $A_t^{(v)}$. Select the shared features under the common view v, then compute $\tilde{X}_t^{(v)}$ and $\tilde{A}_t^{(v)}(t \in T_v)$ in Eq.(5). $U^{(v)}$ is formed by the first c left singular vectors of $\tilde{A}$, where $\tilde{A} = D_1^{-1/2}\tilde{X}D_2^{-1/2}$, and $\tilde{X}$ is the combination of $\tilde{X}_t^{(v)}(t \in T_v)$.
repeat
Update $U_t^{(v)}$ by Eq.(9).
Update $U^{(v)}$ by Eq.(12).
Update $M_t^{(v)}$ by Eq.(17).
until Eq.(6) is convergent.
Run the k-means algorithm on $M_t^{(v)}(t = 1, \ldots, T)$, where v is the most informative view a priori. If there is no prior knowledge on the view informativeness, run the k-means algorithm on the column-wise concatenation of $M_t^{(v)}(v = 1, \ldots, V_t)$.

$\lambda \geq 0$ and $\mu \geq 0$ control the relative importance of the three components. Specifically, Eq.(6) can be seen as the objective function of a multi-view clustering method with $\mu = 0$, and a multi-task clustering method with $\lambda = 0$.

3.4 Optimization

In this subsection, we present an algorithm to optimize Eq.(6) by gradient ascent method. Optimizing Eq.(6) is with respect to variables $U_t^{(v)}$, $U^{(v)}$ and $M_t^{(v)}$. Because of the constraint in Eq.(6), we map the variables to the Stiefel manifold.

Proposition 1 *[Manton, 2002] The complex Stiefel manifold $St(n,p)$ is $St(n,p) = \{X \in C^{n\times p} : X^H X = I\}$. Let $X \in C^{n\times p}$ be a rank p matrix. The projection operator $\pi : C^{n\times p} \to St(n,p)$ is defined to be $\pi(X) = \arg\min_{Q\in St(n,p)} ||X - Q||_F^2$. Moreover, if the SVD of X is $X = U\Sigma V^H$, then $\pi(X) = UI_{n,p}V^H$, where $I_{n,p}$ denotes the $n - by - p$ matrix with ones along the diagonal.*

Computation of $U_t^{(v)}$: Given $M_t^{(v)}$, optimizing Eq.(6) with respect to $U_t^{(v)}$ turns to be

$$\max O_1 = tr(U_t^{(v)T} A_t^{(v)} M_t^{(v)}) \tag{7}$$

Then we get

$$\frac{\partial O_1}{\partial U_t^{(v)}} = A_t^{(v)} M_t^{(v)} \tag{8}$$

According to proposition 1, we can get

$$U_t^{(v)} = \pi(U_t^{(v)} + \gamma \frac{\partial O_1}{\partial U_t^{(v)}} \Big/ \left\| \frac{\partial O_1}{\partial U_t^{(v)}} \right\|_F) \tag{9}$$

Computation of $U^{(v)}$: For the common view v, we get the optimal $U^{(v)}$ by the following computations. Given $M_t^{(v)}$, optimizing Eq.(6) with respect to $U^{(v)}$ turns to be

$$\max O_2 = \mu \sum_{t\in T_v} tr(U^{(v)T} \tilde{A}_t^{(v)} M_t^{(v)}) \tag{10}$$

Then we get

$$\frac{\partial O_2}{\partial U^{(v)}} = \mu \sum_{t\in T_v} \tilde{A}_t^{(v)} M_t^{(v)} \tag{11}$$

According to proposition 1, we can get

$$U^{(v)} = \pi(U^{(v)} + \gamma \frac{\partial O_2}{\partial U^{(v)}} \Big/ \left\| \frac{\partial O_2}{\partial U^{(v)}} \right\|_F) \tag{12}$$

Computation of $M_t^{(v)}$:

1) If $v \in S$ and $t \in T_v$, given $U_t^{(v)}$, $U^{(v)}$, $M_t^{(q)}(q \neq v)$, optimizing Eq.(6) with respect to $M_t^{(v)}$ turns to be

$$\begin{aligned} \max O_3 &= tr(U_t^{(v)T} A_t^{(v)} M_t^{(v)}) \\ &+ \lambda \sum_{q\neq v}^{V_t} tr(M_t^{(v)} M_t^{(v)T} M_t^{(q)} M_t^{(q)T}) \\ &+ \mu tr(U^{(v)T} \tilde{A}_t^{(v)} M_t^{(v)}) \end{aligned} \tag{13}$$

Then we get

$$\begin{aligned} \frac{\partial O_3}{\partial M_t^{(v)}} &= A_t^{(v)T} U_t^{(v)} + \lambda \sum_{q\neq v}^{V_t} 2M_t^{(q)} M_t^{(q)T} M_t^{(v)} \\ &+ \mu \tilde{A}_t^{(v)T} U^{(v)} \end{aligned} \tag{14}$$

2) If $v \notin S$ or $t \notin T_v (v \in S)$, given $U_t^{(v)}$, $M_t^{(q)}(q \neq v)$, optimizing Eq.(6) with respect to $M_t^{(v)}$ turns to be

$$\begin{aligned} \max O_3 &= tr(U_t^{(v)T} A_t^{(v)} M_t^{(v)}) \\ &+ \lambda \sum_{q\neq v}^{V_t} tr(M_t^{(v)} M_t^{(v)T} M_t^{(q)} M_t^{(q)T}) \end{aligned} \tag{15}$$

Then we get

$$\frac{\partial O_3}{\partial M_t^{(v)}} = A_t^{(v)T} U_t^{(v)} + \lambda \sum_{q\neq v}^{V_t} 2M_t^{(q)} M_t^{(q)T} M_t^{(v)} \tag{16}$$

According to proposition 1, we can get

$$M_t^{(v)} = \pi(M_t^{(v)} + \gamma \frac{\partial O_3}{\partial M_t^{(v)}} \Big/ \left\| \frac{\partial O_3}{\partial M_t^{(v)}} \right\|_F) \tag{17}$$

We present the process of optimizing Eq.(6) in Algorithm 1. Our proposed algorithm is typically a gradient ascent algorithm, which is proved to be convergent [Griffin, 2012].

3.5 Time Complexity

Denote d as the feature number, n as the sample number, $iter$ as the iterations of MTMVC and I as the iterations of k-means. The time complexity of the initialization process is $O(d^2n + dn^2)$. The time complexity during iterations is $O(iter(dnc + d^2c + dc^2 + n^2c + c^2n))$. The time complexity of the final clustering part in MTMVC k-means is $O(Icnd)$. The overall time complexity of MTMVC is $O((d^2 + iter(cd + c^2) + Icd)n + (d + iterc)n^2 + iterd^2c + iterdc^2) = O(d^2n + dn^2)$.

Table 1: Clustering Results on WebKB

Method	Task 1		Task 2		Task 3		Task 4	
	Acc(%)	NMI(%)	Acc(%)	NMI(%)	Acc(%)	NMI(%)	Acc(%)	NMI(%)
k-means	61.41± 1.81	16.56± 3.29	52.07± 8.89	13.75± 3.91	52.23± 3.51	10.82± 4.31	61.36± 5.35	20.43± 8.96
NSC	53.23± 0.06	25.03± 0.01	46.83± 0.00	26.99± 0.00	60.67± 0.07	40.27± 0.01	64.17± 0.01	35.58± 0.00
kk-means	48.04± 2.17	13.62± 2.34	44.72± 2.31	19.04± 0.73	46.47± 1.20	14.64± 1.90	56.22± 0.97	29.42± 0.65
BiCo	65.39± 9.36	34.01± 4.76	62.29± 5.64	28.52± 2.36	65.31± 5.79	31.75± 4.30	72.78± 9.22	51.62± 6.33
CoRe	75.66± 0.00	48.33± 0.00	68.65± 0.03	31.84± 0.10	67.25± 0.06	50.62± 0.04	70.75± 0.04	41.78± 0.05
CoTr	65.35± 6.32	45.26± 7.46	62.50± 8.23	30.07± 4.80	80.98± 8.39	60.05± 7.41	76.07± 2.04	54.81± 2.69
LSSMTC	62.18± 8.22	25.61± 4.86	62.03± 5.14	29.66± 2.77	58.62± 6.31	25.66± 2.29	66.22± 9.84	33.57± 5.94
LSSMTC(CF)	63.36± 7.89	26.73± 4.65	64.20± 3.86	24.89± 3.42	63.09± 8.44	33.27± 4.21	67.71± 6.97	36.79± 6.10
LNKMTC	63.02± 6.09	29.99± 7.94	60.32± 7.37	30.40± 7.19	55.52± 10.33	29.04± 6.36	66.35± 5.75	38.16± 6.27
LNKMTC(CF)	55.08± 5.99	27.61± 9.44	60.99± 3.98	31.88± 10.84	65.37± 11.45	35.82± 9.49	68.63± 8.62	40.82± 8.23
MTMVC-MT	70.26± 11.12	38.59± 4.82	69.80± 5.65	32.77± 3.83	68.03± 5.49	35.32± 2.62	80.39± 4.27	57.33± 3.47
MTMVC-MT(CF)	74.64± 3.09	42.14± 2.32	69.20± 1.72	34.52± 4.33	66.43± 5.31	32.59± 2.84	80.87± 2.76	58.74± 1.37
MTMVC-MV	82.03± 4.91	67.14± 2.42	76.54± 4.95	50.88± 3.17	81.62± 5.84	59.02± 3.80	80.42± 3.79	62.26± 2.25
MTMVC-CV	**84.46± 4.41**	**70.21± 3.49**	**80.07± 0.16**	**55.60± 1.54**	82.58± 5.68	61.87± 6.94	83.32± 0.13	64.44± 0.49
MTMVC	83.31± 3.58	67.72± 3.27	79.12± 3.45	53.21± 3.47	**83.41± 1.16**	**64.49± 0.41**	**87.03± 1.17**	**65.21± 0.27**

Table 2: Clustering Results on NG

Method	Task 1		Task 2		Task 3		Task 4	
	Acc(%)	NMI(%)	Acc(%)	NMI(%)	Acc(%)	NMI(%)	Acc(%)	NMI(%)
k-means	28.60± 2.98	6.78± 3.30	29.51± 3.00	11.28± 4.23	28.46± 1.25	5.89± 2.69	28.85± 2.67	4.97± 4.87
NSC	26.30± 0.00	2.95± 0.00	28.64± 3.34	3.73± 1.70	28.32± 0.81	2.77± 0.44	28.50± 0.00	1.93± 0.15
kk-means	32.26± 0.06	3.41± 0.03	34.27± 0.06	9.61± 0.03	29.65± 0.11	2.07± 0.05	35.76± 0.14	3.14± 0.04
BiCo	61.66± 4.73	41.53± 4.36	63.94± 1.60	42.61± 3.81	63.03± 3.60	54.43± 8.26	70.10± 8.66	59.66± 6.53
CoRe	27.05± 0.60	2.23± 0.38	32.51± 2.64	5.29± 1.32	29.16± 0.45	2.86± 0.38	28.69± 0.00	2.16± 0.00
CoTr	43.72± 5.96	14.29± 4.68	47.42± 8.46	18.78± 9.11	39.97± 4.91	10.84± 3.71	48.14± 7.31	18.86± 6.95
LSSMTC	31.26± 3.05	5.56± 2.74	42.79± 3.97	21.37± 4.84	35.63± 4.53	7.23± 4.68	40.95± 4.46	11.05± 4.05
LSSMTC(CF)	30.47± 3.80	8.74± 2.88	45.04± 4.65	25.27± 4.37	33.83± 2.51	10.12± 3.08	40.12± 9.30	17.46± 7.84
LNKMTC	69.62± 9.08	45.85± 13.19	59.01± 9.22	48.20± 8.67	48.44± 6.42	44.86± 8.58	59.72± 8.80	45.55± 5.59
LNKMTC(CF)	69.88± 14.25	51.50± 7.84	59.69± 5.91	51.25± 7.97	54.23± 12.42	50.70± 8.53	70.18± 9.14	52.45± 8.05
MTMVC-MT	74.43± 9.51	60.57± 7.65	67.32± 1.51	49.70± 2.31	71.43± 7.26	59.13 ± 7.54	74.11± 1.94	61.95± 2.39
MTMVC-MT(CF)	75.19± 7.55	62.04± 8.51	66.43± 1.89	48.57± 5.09	70.64± 5.23	61.27 ± 6.68	79.96± 7.37	66.55± 7.10
MTMVC-MV	75.52± 9.36	62.31± 2.99	71.82± 5.69	57.96 ± 4.99	70.33± 7.31	61.42± 6.22	79.01± 5.30	67.42± 3.33
MTMVC-CV	79.08± 8.94	64.50± 3.71	73.36± 3.22	58.86± 5.26	72.12 ± 6.10	62.51± 4.17	81.09± 2.84	69.36± 3.15
MTMVC	**80.95± 7.29**	**65.35± 2.63**	**74.29± 3.88**	**62.03± 4.02**	**74.61± 5.74**	**64.77± 2.62**	**82.28± 3.13**	**70.26± 2.53**

4 Experiments

4.1 Data Sets

WebKB[1]: The WebKB data set contains web pages collected from computer science department websites at 4 universities. They are divided into 7 categories, we choose 4 most populous categories such as course, faculty, project and student for clustering. The data set has 4 tasks, each is to cluster the web pages of an university. There are three views for each task: the words in the main texts of all the 4 universities constitute the common view, since the tasks share lots of words from the main text view; the words in the hyperlinks pointing to the web pages of this university and the words in the titles of the web pages of this university constitute two task specific views respectively, since the tasks share few words from the hyperlink view and title view.

20NewsGroups[2]: The 20NewsGroups data set is composed of 6 root categories, under which are 20 sub categories. We generate the NG data set from the samples of 4 root categories (comp, rec, sci and talk), each with 4 sub categories. NG has 4 tasks, each containing samples from 4 root categories.

Email[3]: The Email data set contains the emails of 3 inboxes from different users. The data set has 3 tasks, with each task formed by an inbox from a specific user. We aim to divide each task into 2 clusters, one is spam, the other is non-spam.

For NG and Email, we adopt the way of constructing multi-task multi-view data in [He and Lawrence, 2011; Zhang and Huan, 2012], which has been effectively used to validate the multi-task multi-view learning methods. Based on the way in [He and Lawrence, 2011; Zhang and Huan, 2012], for each task, the common view consists of the words appearing in the main texts of all the tasks, the task specific view consists of the words that only appear in this task. We use Rainbow[4] for data preprocessing: removing the header lines and stop words, selecting words by mutual information.

4.2 Baseline Methods

To the best of our knowledge, there is no existing work for multi-task multi-view clustering. We compare the MTMVC algorithm with (1) typical single-task single-view clustering methods: *k*-means, normalized spectral clustering algorithm (NSC) [Ng *et al.*, 2001], kernel *k*-means (*kk*-means), bipartite graph co-clustering algorithm (BiCo) [Dhillon, 2001]; (2) multi-view clustering methods: co-regularized multi-view spectral clustering algorithm (CoRe) [Kumar *et al.*, 2011], co-trained multi-view spectral clustering algorithm (CoTr) [Ku-

[1]http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

[2]http://qwone.com/ jason/20Newsgroups/

[3]http://www.ecmlpkdd2006.org/challenge.html

[4]http://www.cs.cmu.edu/∼mccallum/bow/

Table 3: Clustering Results on Email

Method	Task 1		Task 2		Task 3	
	Acc(%)	NMI(%)	Acc(%)	NMI(%)	Acc(%)	NMI(%)
k-means	55.32± 1.59	3.30± 0.68	50.60± 0.61	0.38± 0.18	54.68± 1.68	6.08± 1.99
NSC	50.20± 0.00	1.16± 0.51	51.67± 1.97	1.36± 0.53	50.34± 0.00	1.17± 0.39
kk-means	55.94± 0.00	1.28± 0.00	54.10± 0.00	0.57± 0.00	60.08± 0.00	7.75± 0.00
BiCo	61.30± 0.08	16.84± 0.09	72.16± 0.02	32.17± 0.04	70.68± 0.00	31.92± 0.00
CoRe	55.03± 4.19	3.71± 2.16	54.48± 3.05	2.04± 0.96	50.84± 1.90	2.11± 1.81
CoTr	54.01± 0.73	8.35± 1.09	58.73± 1.59	3.13± 0.93	68.34± 0.30	14.88± 0.32
LSSMTC	67.24± 0.65	22.58± 2.15	50.42± 0.67	0.18± 0.41	53.32± 2.27	0.91± 1.19
LSSMTC(CF)	68.26± 0.92	23.12± 3.44	50.50± 0.89	0.71± 1.29	52.45± 1.63	0.75± 0.52
LNKMTC	62.96± 6.88	23.87± 5.73	64.50± 6.38	28.46± 6.26	71.76± 4.50	28.52± 6.17
LNKMTC(CF)	62.04± 6.78	18.81± 11.42	66.39± 7.44	16.87± 12.62	65.22± 9.39	18.11± 11.50
MTMVC-MT	69.38± 0.18	26.61± 0.10	75.36± 0.23	36.01± 0.34	79.52± 0.29	38.26± 0.43
MTMVC-MT(CF)	64.44± 0.35	20.55± 0.30	72.98± 0.02	32.63± 0.03	73.20± 0.04	33.71± 0.08
MTMVC-MV	62.82± 0.02	18.64± 0.03	76.64± 0.13	36.94± 0.29	76.00± 0.47	35.99± 0.62
MTMVC-CV	70.24± 0.28	27.67± 0.31	76.82± 0.08	37.20± 0.12	80.32± 0.16	41.98± 0.38
MTMVC	**70.32± 0.11**	**27.81± 0.14**	**77.20± 0.05**	**37.87± 0.08**	**81.16± 0.35**	**43.12± 0.42**

mar and Daumé, 2011]; (3) multi-task clustering methods: the shared subspace learning multi-task clustering algorithm (LSSMTC) [Gu and Zhou, 2009] and the kernel learning multi-task clustering algorithm (LNKMTC) [Gu *et al.*, 2011].

In addition, we evaluate MTMVC without the second component (MTMVC-MT), regressing to a multi-task algorithm, and without the third component (MTMVC-MV), regressing to a multi-view algorithm. In general, we use the clustering result of MTMVC from the most informative view (as all multi-view clustering methods do). But sometimes we may not know the most informative view, so we also evaluate MTMVC by running *k*-means on column-wise concatenation of $M_t^{(v)}(v = 1, \ldots, V_t)$, we denote this version as MTMVC-CV.

As existing multi-task clustering methods can only work in a view which contains the features shared by all the tasks, we perform LSSMTC, LNKMTC and MTMVC-MT on the common view. To make the three methods exploit the information of task specific views, we also perform them on a concatenation of the features of all the views, and call them LSSMTC(CF), LNKMTC(CF) and MTMVC-MT(CF).

4.3 Settings

For NSC, *kk*-means, CoRe and CoTr, the Gaussian kernel width δ is set with the median Euclidean distance between samples of each task, which can self-adapt to the data and has been commonly used in the literatures [Gretton *et al.*, 2006; Kumar *et al.*, 2011]. For the other parameters, we apply grid searching [Zhang and Huan, 2012] to identify the optimal values. For the baseline methods, we extend the grid searching range largely upon the original authors's settings. For CoRe, the parameter λ is set by searching the grid $\{0.01, 0.02, \ldots 0.99, 1\}$. For CoTr, the number of selected eigenvectors is set by searching the grid $\{1, 1.1, \ldots, 1.5\} \times c$, where c is the cluster number. For LSSMTC and LSSMTC(CF), the parameter λ is set by searching the grid $\{0.1, 0.2, \ldots 0.9\}$, the dimensionality of the shared subspace l is set by searching the grid $\{2, 4, 6, 8, 10\}$. For LNKMTC and LNKMTC(CF), the neighborhood size is set by searching the grid $\{10, 20, \ldots 100\}$, the regularization parameter C is set by searching the grid $\{0.1, 1, 10, 100, 500, 1000\}$, the trace parameter b is set by searching the grid $\{10, 20, 30, 40, 50\}$. For MTMVC and MTMVC-CV, we set the parameter of step length $\gamma = 1$, and set λ, μ by searching the grid $\{0.1, 0.2, \ldots, 1\}$. Specifically, we set $\lambda = 0$ for MTMVC-MT and MTMVC-MT(CF), $\mu = 0$ for MTMVC-MV.

To evaluate the clustering results, we adopt two performance measures in [Xu *et al.*, 2003]: clustering accuracy (Acc) and normalized mutual information (NMI).

4.4 Clustering Results

We repeat each algorithm 10 times under each parameter setting. For the algorithms LSSMTC(CF), LNKMTC(CF), MTMVC-MT(CF) and MTMVC-CV, we show the mean result and the standard deviation corresponding to the best parameter setting. For the other algorithms, we show those of the most informative view corresponding to the best parameter setting. For multi-view clustering methods, we cluster each task at a time. From the clustering results in Table 1, 2 and 3, the following observations could be made.

(1) BiCo performs better than the single-task single-view baseline methods (consistent with experiments in previous works [Dhillon, 2001]), and even performs better than the multi-view and multi-task baseline methods (without using co-clustering as building block) sometimes. This is because that for non-negative data such as documents, BiCo can boost the performance of sample clustering by the clustering of features and vice versa. Whereas the baseline methods only cluster the samples without the sample-feature mutual boosting.

(2) The methods with simple feature concatenation such as LSSMTC(CF), LNKMTC(CF) and MTMVC-MT(CF) do not help much on improving the clustering performance, and they may perform worse than LSSMTC, LNKMTC and MTMVC-MT on a single view sometimes. From the comparison between MTMVC and MTMVC-MT(CF), it can be seen that applying multi-view relationship learning is more effective than simply concatenating features. This is consistent with the conclusions in the multi-view clustering literatures.

(3) MTMVC-MV outperforms the multi-view clustering algorithms CoRe and CoTr in most cases, because whether multi-view relationship learning can help improve the clustering performance highly depends on the performance of the basic clustering. CoRe and CoTr use NSC as the ba-

sic clustering method, while MTMVC-MV uses BiCo. BiCo is shown to perform much better than NSC for non-negative data in most cases in the three tables.

(4) MTMVC-MT outperforms the multi-task clustering algorithms LSSMTC and LNKMTC in most cases, because: 1) the basic clustering method BiCo used by MTMVC-MT performs better than k-means used by LSSMTC and kk-means used by LNKMTC; 2) during the multi-task relationship learning, MTMVC-MT uses BiCo to explicitly learn the shared subspace by combining the associated features, whereas LSSMTC does not; 3) there are also some specific characteristics within each task, MTMVC-MT uses the within-view-task clustering part to preserve the knowledge available locally in each task, while LNKMTC only gets the clustering result through a common kernel space.

(5) MTMVC and MTMVC-CV further improve upon BiCo, MTMVC-MV and MTMVC-MT, thus they perform much better than all the other baseline algorithms. This is because that MTMVC and MTMVC-CV contain a multi-view relationship learning component and a multi-task relationship learning component, thus it can take advantages of both the consistency among different views and the relationship among related tasks, whereas the baselines take advantage of only one (or none) of the two functions. In most cases MTMVC performs better than MTMVC-CV, since it considers the most informative view. In some cases when the most informative view is not so dominant, MTMVC-CV, which takes into account the clustering of all the views and gets a result of trade off, performs a little better than MTMVC. Overall, MTMVC performs better than or comparable to MTMVC-CV.

5 Conclusion

In this paper, we propose a multi-task multi-view clustering (MTMVC) framework which integrates within-view-task clustering, multi-view relationship learning and multi-task relationship learning, and solve the optimization problem by using a gradient ascent method which exploits unitary constraints. As far as we know, this is the first work addressing multi-task multi-view clustering. Experimental results show the superiority of the proposed algorithm over either multi-task clustering or multi-view clustering algorithms for multi-task clustering of multi-view data. Our algorithm uses bipartite graph co-clustering as the basic clustering method, thus it works for non-negative data. For future work, we will solve the general multi-task multi-view clustering problem.

Acknowledgments

This work was supported by National Science Foundation of China (No. 61272374,61300190).

References

[Banerjee *et al.*, 2007] Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. *Journal of Machine Learning Research*, 8:1919–1986, 2007.

[Cho *et al.*, 2004] Hyuk Cho, Inderjit S. Dhillon, Yuqiang Guan, and Suvrit Sra. Minimum sum-squared residue co-clustering of gene expression data. In *SDM*, pages 114–125, 2004.

[de Sa, 2005] Virginia R. de Sa. Spectral clustering with two views. In *ICML*, pages 20–27, 2005.

[Dhillon, 2001] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, pages 269–274, 2001.

[George and Merugu, 2005] Thomas George and Srujana Merugu. A scalable collaborative filtering framework based on co-clustering. In *ICDM*, pages 625–628, 2005.

[Gretton *et al.*, 2006] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two-sample-problem. In *NIPS*, pages 513–520, 2006.

[Griffin, 2012] Christopher Griffin. *Numerical Optimization: Penn State Math 555 Lecture Notes*. Creative Commons, Penn State University, PA, USA, 2012.

[Gu and Zhou, 2009] Quanquan Gu and Jie Zhou. Learning the shared subspace for multi-task clustering and transductive transfer classification. In *ICDM*, pages 159–168, 2009.

[Gu *et al.*, 2011] Quanquan Gu, Zhenhui Li, and Jiawei Han. Learning a kernel for multi-task clustering. In *AAAI*, pages 368–373, 2011.

[He and Lawrence, 2011] Jingrui He and Rick Lawrence. A graph-based framework for multi-task multi-view learning. In *ICML*, pages 25–32, 2011.

[Kumar and Daumé, 2011] Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *ICML*, pages 393–400, 2011.

[Kumar *et al.*, 2011] Abhishek Kumar, Piyush Rai, and Hal Daumé. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

[Manton, 2002] Jonathan H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Transactions on Signal Processing*, 50(3):635–650, 2002.

[Ng *et al.*, 2001] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[Vidal, 2011] René Vidal. Subspace clustering. *IEEE Signal Process. Mag.*, 28(2):52–68, 2011.

[Xu *et al.*, 2003] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *SIGIR*, pages 267–273, 2003.

[Zhang and Huan, 2012] Jintao Zhang and Jun Huan. Inductive multi-task learning with multiple view data. In *KDD*, pages 543–551, 2012.

[Zhang and Zhang, 2010] Jianwen Zhang and Changshui Zhang. Multitask Bregman clustering. In *AAAI*, pages 655–660, 2010.

[Zhang and Zhang, 2013] Xianchao Zhang and Xiaotong Zhang. Smart multi-task Bregman clustering and multi-task Kernel clustering. In *AAAI*, pages 1034–1040, 2013.