Multi-Task Clustering of Human Actions by Sharing Information

Xiaoqiang Yan, Shizhe Hu, Yangdong Ye*
School of Information Engineering
Zhengzhou University, 450000, Zhengzhou, China
iexqyan@gmail.com, ieszhu@gs.zzu.edu.cn, ieydye@zzu.edu.cn

Abstract

Sharing information between multiple tasks can enhance the accuracy of human action recognition systems. However, using shared information to improve multi-task human action clustering has never been considered before, and cannot be achieved using existing clustering methods. In this work, we present a novel and effective Multi-Task Information Bottleneck (MTIB) clustering method, which is capable of exploring the shared information between multiple action clustering tasks to improve the performance of individual task. Our motivation is that, different action collections always share many similar action patterns, and exploiting the shared information can lead to improved performance. Specifically, MTIB generally formulates this problem as an information loss minimization function. In this function, the shared information can be quantified by the distributional correlation of clusters in different tasks, which is based on a high-level common vocabulary constructed through a novel agglomerative information maximization method. Extensive experiments on two kinds of challenging data sets, including realistic action data sets (HMDB & UCF50, Olympic & YouTube), and cross-view data sets (IXMAS, WVU), show that the proposed approach compares favorably to the state-of-the-art methods.

1. Introduction

Human action recognition is a fundamental research area in computer vision. Recently, with the continuing rapid development of information technology, massive amounts of task-specific human action data are generated everyday. In realistic videos, recognizing action categories from each data collection can be treated as a learning task. Apparently, different video collections usually have a considerable amount of similar actions. For instance, both UCF50 [15] and HMDB [4] contain motion patterns: punching, horse riding, pushup and fencing. Intuitively, the shared pattern information can be exploited to enhance the clustering performance of each task. In cross-view videos, the same actions are captured from different camera viewpoints. We assume that action pattern discovery in each viewpoint is treated as a learning task. Due to the problem of self-occlusions, the single view case can not guarantee robust action recognition. Figure 1 shows the existence of shared information between tasks. Therefore, jointly learning all of the tasks together can leverage the shared knowledge among them to improve the generalization ability of model learning.

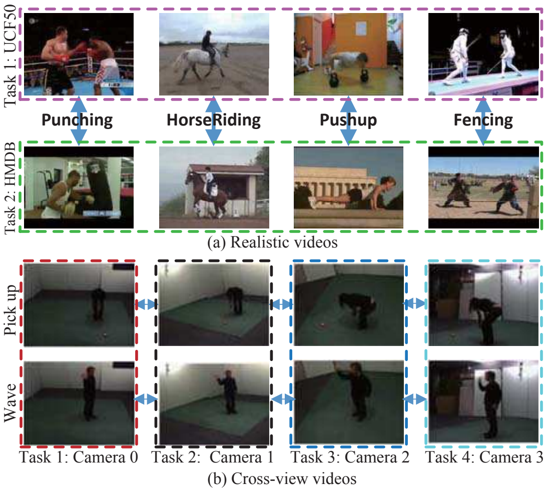


Figure 1. The shared information between tasks. (a) The similar action patterns in UCF50 and HMDB can be treated as shared information. (b) In cross-view videos, the same actions from different viewpoints can be adopted as shared information.

Recently, several multi-task learning (MTL) approaches [6, 10, 13, 26, 28, 34, 9] have been proposed for human action recognition, which capitalize on shared information between related tasks to improve the performance of each task. However, MTL needs to acquire sufficient labeled samples for each task, it is may be impractical for many complicated applications. Moreover, recognizing action patterns is usually challenging just with the human knowledge (label, annotation, etc.), which often invites subject biases or mistakes by human labelers. So it is wise to resort to clustering algorithms for mining the human action in videos.

Human action clustering is crucial to many practical applications, such as fast content based video retrieval or automatic annotation of video databases. However, although the current single task methods have demonstrated superior performance on human action clustering, there still exist the following challenges: 1) Neglecting the shared information among actions. Most of current methods focus on designing features to distinguish actions in the single task setting. For instance, Niebles *et al*. [12] use pLSA and LDA to cluster the actions based on local spatial-temporal feature. Yang *et al*. [27] present a meaningful global action descriptor by hierarchical clustering of optical flow feature. However, the feature representation is not discriminative enough to differentiate actions in more complicated scenarios, such as multi-camera [22], cross-domain [35], etc. It will be helpful if we employ the shared information from other tasks for the more challenging action recognition. 2) Difficulty in shared information measurement. In real applications, although many tasks contain similar action patterns, there are some tasks still mutually partially related, dissimilar even reverse. For instance, both UCF50 and HMDB have same action patterns of punching and fencing, but they also have many completely different patterns, e.g. kissing in UCF50 and biking in HMDB. So it is quite challenging to measure the shared information in realistic tasks.

In this paper, to perform multi-task clustering of human actions by sharing the related information between multiple tasks, we propose a novel multi-task information bottleneck (MTIB) clustering method. MTIB is capable of exploring the shared information between multiple human action clustering tasks to improve the performance of each task. Specifically, to bridge the distributional gap between multiple tasks, as well as local features and action concepts, we first present an agglomerative information maximization (AIM) method to construct a high-level common vocabulary between multiple tasks based on bag-of-visual-words model (See Figure 2). The common vocabulary of multiple tasks is more discriminative than the vocabulary from individual task. For instance, the common vocabulary may contain phrase “raising your hand” that implies high-level concept, other than separate words “rasing” and “hand”. Then, MTIB generally formulates the multi-task human action clustering as minimizing an information loss function, in which the shared information between any two tasks can be quantified by the distributional correlation based on the co-occurrence words from the common vocabulary. To solve the optimization of MTIB function, a rotational draw-and-merge solution is proposed to update the action partition. Extensive experiments are conducted on two kinds of challenging data sets, including realistic action data sets (HMDB [4] & UCF50 [15], Olympic [11] & YouTube [7]), and cross-view data sets (IXMAS [21], WVU [14]).

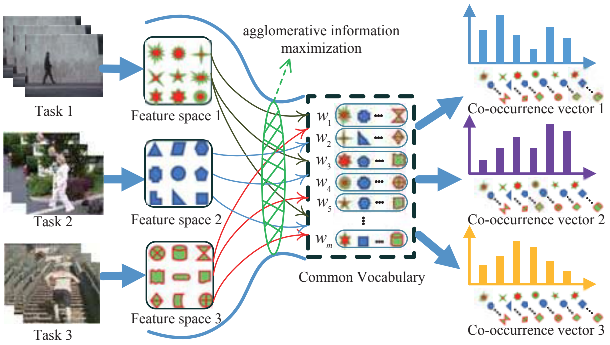


Figure 2. The action representation from multiple tasks based on common vocabulary.

The major contributions of this paper are summarized as follows: 1) A novel and effective multi-task information bottleneck method is proposed for human action clustering. To our knowledge, this is the first work proposing multi-task framework for human action clustering. 2) An agglomerative information maximization method is proposed to bridge the gap between multiple tasks, which is general and can be beneficial to many other fields, such as cross-domain, multi-view, transfer learning, etc. 3) The multi-task human action clustering is generally formulated as an information loss minimization function, in which the task relatedness can be quantified by the distributional correlation of clusters between different tasks. 4) A novel rotational draw-and-merge solution is proposed to update the data partition, which can guarantee to converge to a stable solution.

2. Related Work

2.1. Multi-task Scenario

Several MTL approaches [28, 34, 13, 10, 26, 6] have been proposed for human action recognition by jointly learning multiple tasks using shared information among them. For instance, Yuan *et al*. [28] treat learning the sparse representation under each feature modality as a task. Since the multiple features are generated from same input, they are inter-related. Pentina *et al*. [13] propose to solve task in a sequential manner by transferring information from a previously learned task to the next one, instead of solving all of them simultaneously. Mahasseni *et al*. [10] and Yan *et al*. [26] find that multi-task learning is suitable for achieving view invariance in recognition when each viewpoint of action set is specified as a learning task. Liu *et al*. [6] propose to discover the latent task correlation as well as to learn the action model simultaneously. However, MTL needs to acquire sufficient labeled samples for each task, which may be impractical for massive action data.

Recently, several multi-task clustering (MTC) methods have been designed in the domain of machine learning. Gu *et al.* [2] first address multi-task clustering by learning a shared subspace representation of all tasks, through which the knowledge of the tasks can be transferred to each other. After that, the works in [25, 29, 30, 31, 32, 33] have obtained promising results for different multi-task settings. For instance, Zhang *et al.* [31] propose a multi-task multi-view clustering algorithm, which integrates the features in the common view of each task to link the related tasks together. Zhang [33] proposes two convex multi-task clustering objectives, which aim to learn a shared feature representation and the task relationship, respectively. However, all previous works are designed for document analysis. Recently, Jones *et al.* [3] estimate the correlation between two human action clusterings and use it to improve the results of both clusterings, but they just focus on two tasks. Yan *et al.* [25] propose multi-task clustering based on earth movers distance for first-person vision activity analysis, which concentrates on long time video sequence and can not be applied to large volume of action collections.

2.2. Information Bottleneck

Information Bottleneck (IB) [20] is an information-theoretic framework, which has been applied to action recognition effectively [8]. Given the joint distribution of a source variable X and another relevant variable Y, IB tries to extract a compressed representation T of X, while preserving information about Y. Formally, the IB objective function is suggested in [20] as follows:

$$\mathcal{L}_{IB}[p(t|x)] = I(T;X) - \beta I(T;Y), \qquad (1)$$

where the tradeoff parameter β is the positive Lagrange multiplier controlling compression and informativeness, $I(T;X)$ is the mutual information defined in Eq. 2. IB has been extended successfully to multivariate scenario [17], such as multi-view [23], consensus clustering [24], etc. So it is natural to consider using IB principle to tackle multiple tasks. To the best of our knowledge, this is the first work addressing multi-task clustering by information bottleneck principle.

3. Multi-task Clustering by Sharing Information

In this section, we first describe the problem of multi-task clustering of human actions by sharing information. Then, we present an agglomerative information maximization (AIM) to construct common vocabulary to bridge the gap of multiple tasks. Finally, the objective function of MTIB and its optimization are given in details.

Given multiple collections of unlabeled videos including various human actions, we intend to cluster each video collection into discrete groups of videos with similar actions. In realistic applications, the action patterns in different collections are always similar to each other. For instance, Olympic [11] and YouTube [7] are sports data corpus, and both of them contain various similar sports action patterns. If recognizing action categories in each collection is treated as a learning task, we are curious about whether we can maintain the shared pattern information to enhance the clustering performance of each task.

3.1. Agglomerative Information Maximization

One key issue of human action recognition in multi-task scenario is how to represent the action. Recently, bag-of-visual-words (BoVW) [16] model represents a video as an orderless set of local features and has been demonstrated impressive levels of performance. Traditional BoVW utilizes k-means to quantify the local features into visual words, which generates vocabulary for each action collection independently. However, the independent vocabularies of different tasks are heterogeneous to each other, and can not be used to measure the shared information of multiple tasks. To bridge the gap between multiple tasks, as well as low-level features and action concepts, we present an agglomerative information maximization (AIM) method to construct common vocabulary W^{com} of multiple tasks, which is suitable to describe multiple tasks as demonstrated in our experiments. In this regard, the common vocabulary may contain "phase" other than separate words. Next, we give the AIM method in details.

Consider multiple tasks $X^1, X^2, \cdots, X^m$, we first extract a set of space-time interest points $D = \{D^1, D^2, \cdots, D^m\}$ for each task with the Harris3D detector and the HoG/HoF descriptor [5], and each task can generate a set of 162-dimensional feature vector $R = \{R^1, R^2, \cdots, R^m\}$. Instead of building vocabulary of each task separately, we wish to find a more compact and yet discriminative common representation W^{com} of interest points $D = \{D^1, D^2, \cdots, D^m\}$ from multiple tasks. In this study, we use mutual information to measure the similarity between two variables, which can be defined as:

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}, \qquad (2)$$

so $I(W^{com}; D^i)$, $1 \leq i \leq m$, signifies how compact the new representation W^{com} is. However, that representation may not be discriminative, because it does not give any information regarding the feature variable R^i from W^{com}. Therefore, this problem can be expressed as an information maximization function:

$$\mathcal{L}_{max}(p(w^{com}|d)) = \\ \sum_{i=1}^{m} I(W^{com}; R^i) - \lambda^{-1} \cdot \sum_{i=1}^{m} I(W^{com}; D^i), \quad (3)$$

where λ is the Lagrange Multiplier controlling the trade-off between the information compression $\sum_{i=1}^{m} I(W^{com}; R^i)$ and the information preservation $\sum_{i=1}^{m} I(W^{com}; D^i)$.

In this study, we employ an agglomerative framework [18, 8] to solve the function 3, in which two elements with the least merger cost will be merged together at each step. The major extension in our method compared with [18, 8] is that all the elements which has the least merger cost in all tasks can be merged together into a new element, instead of merging the pair of elements which are rooted in single task. Let $\hat{w}_1$ and $\hat{w}_2$ be the two elements of W^{com}, the information loss of the merging of $\hat{w}_1$ and $\hat{w}_2$ is then defined as:

$$d(\hat{w}_1, \hat{w}_2) = \sum_{i=1}^{m} [I(\hat{W}^{com}_{bef}; R^i) - I(\hat{W}^{com}_{aft}; R^i)], \quad (4)$$

where $I(\hat{W}^{com}_{bef}; R^i)$ and $I(\hat{W}^{com}_{aft}; R^i)$ are the mutual information before and after $\hat{w}_1$ and $\hat{w}_2$ are merged for all tasks. The probability distributions $p(\hat{w})$, $p(r|\hat{w})$ and $p(\hat{w}|w)$ are calculated as:

$$p(\hat{w}) = p(\hat{w}_1) + p(\hat{w}_2), \quad (5)$$

$$p(R_i|\hat{w}) = \frac{p(\hat{w}_1)}{p(\hat{w})} p(R_i|\hat{w}_1)) + \frac{p(\hat{w}_2)}{p(\hat{w})} p(R_i|\hat{w}_2). \quad (6)$$

After the determination of which pair of elements should be merged, we can give the algorithm of AIM as follows:

1) Initialize all sampling feature points as a singleton cluster.

2) At each step, compute the merger cost $d(\hat{w}_1, \hat{w}_2)$ between all pair of elements from multiple tasks.

3) Select the pair which gives the minimum information loss $argmin\{d(\hat{w}_1, \hat{w}_2)\}$.

4) Update the probability distributions $p(\hat{w})$, $p(r|\hat{w})$ and $p(\hat{w}|w)$, until the number of clusters reaches predefined value.

Once the common vocabulary is determined, the shared information of multiple tasks can be discovered by the cooccurrence words in the the common vocabulary. Intuitively, one action collection can be interpreted by a set of action clusters, and similar collection consists of similar clusters. So we can utilize mutual information of clusters in different tasks to measure the distributional correlation between two tasks. Let $C^s_i = \{x^s_1, x^s_2, \cdots, x^s_{n_i}\}$ and $C^t_j = \{x^t_1, x^t_2, \cdots, x^t_{n_j}\}$ be the clusters in task T^s and T^t, respectively, where n_i and n_j are the number of instances in the clusters C^s_i and C^t_j. Then we can obtain the similarity matrix $Z^{i,j}$ between the two clusters C^s_i and C^t_j, where each entry is the co-occurrence of key word between the two clusters. So the mutual information $I(C^s_i; C^t_j)$ can be calculated now. Then, the mutual information between any two tasks T^s and T^t can be defined as follows:

$$I(T^s; T^t) = \sum_{i=1}^{n_s} \max_{j=1}^{n_t} I(C^s_i; C^t_j), \quad (7)$$

where the n_s and n_t are the number of clusters in task T^s and T^t. Now, given two clusters from different tasks, we can calculate their mutual information according to Eq. 7. Next, we will give the objective function of our multi-task human action clustering method, which involves data compressing in individual task and the measurement of shared information cross tasks.

3.2. Objective Function of MTIB

Once the shared information of multiple tasks is discovered, we can build the objective function of our multi-task human action clustering method: MTIB. Suppose there are m human action clustering tasks $X^1, X^2, \cdots, X^m$, each task X^k $(1 \leq i \leq m)$ takes value from a video collection $\mathcal{X}^k = \{x^k_1, x^k_2, \cdots, x^k_{n_k}\}$, where n_k is the number of videos in the k-th task. Accordingly, there are m discrete random variables $\{Y^1, Y^2, \cdots, Y^m\}$ on behalf of the m feature variables of the tasks, which are mapped from the common vocabulary $W^{com} = \{w_1, w_2, \cdots, w_d\}$ of multiple tasks. Then, we can build corresponding joint distributions $p(X^1, Y^1), \cdots, p(X^m, Y^m)$ for each task. So the goal of our multi-task clustering method is to learn a good compressed representation $p(t^k|x^k)$ of X^k to T^k from its own feature variable Y^k.

The objective function of MTIB is built in twofold setting: 1) Data compression. In this part, the source human action collection X^k is compressed into a compact representation T^k (we also call it "bottleneck variable"). 2) Relevant information preservation. This part means each bottleneck variable T^k attempts to preserve the maximum information in terms of its own feature variable Y^k and the shared information with other tasks. The objective function of MTIB can be formulated as follows:

$$\mathcal{L}_{max}[p(t^k|x^k)] = -\beta^{-1} \cdot \sum_{k=1}^{m} I(T^k; X^k) + \\ [\sum_{k=1}^{m} I(T^k; Y^k) + \sum_{s=1}^{m} \lambda_s \cdot \sum_{t=1, t \neq s}^{m} I(T^s; T^t)], \quad (8)$$

where $\sum_{k=1}^{m} I(T^k; X^k)$ measures the compactness between X^k and its new representation T^k, $\sum_{k=1}^{m} I(T^k; Y^k)$ measures how much relevant information each bottleneck variable T^k preserves about the relevant variable Y^k,

$I(T^s;T^t)$ quantifies the correlation among tasks by calculating the mutual information between pairwise clusters from any two tasks. β is the balance parameter controlling the trade-off between information compression and preservation. $\lambda_s \geq 0$ $(1 \leq s \leq m)$ controls the influence of other tasks.

In clustering scenario, the number of categories M is much less than the size of each video collection $|X^k|$, i.e. $M \ll |X^k|$, which implies a significant compression. Therefore, to maximally preserve the relevant information and fully explore the correlation among tasks, we set the value of β as ∞. Now, the objective function of MTIB can be rewritten as:

$$\begin{aligned}&\mathcal{L}_{max}[p(t^k|x^k)] = \\ &\sum_{k=1}^{m} I(T^k;Y^k) + \sum_{s=1}^{m} \lambda_s \cdot \sum_{t=1, t\neq s}^{m} I(T^s;T^t).\end{aligned} \tag{9}$$

In this paper, we consider the hard clustering, which means the value of $p(t^k|x^k)$ is either 0 or 1. Now, the remaining task is to optimize objective function Eq. 9.

3.3. Optimization of MTIB

In this section, a rotational draw-and-merge optimization solution is proposed to obtain the partition of each task. To begin with, the solution partitions each tasks $X^1, X^2, \ldots, X^m$ into M clusters and obtains an initialization. Then, for the task T^k, we perform the following two procedures at each step, while remaining the other tasks stationary. 1) Draw each data point x^k from the current cluster $t_k(x^k)$ and treat it as a singleton cluster $\{x^k\}$, thus the current task has $M+1$ clusters. 2) The singleton cluster $\{x^k\}$ must be merged into a new clusters t_k^{new} to ensure the total number of clusters is M. After these two steps, we do the same procedure as the task T^k for the next task. So we guarantee that each data point of all tasks is gradually merged into a better cluster.

In the rotational draw-and-merge procedure, we attempt to merge each data point $\{x^k\}$ of every task into an optimal cluster t_k^{new} at each step. For clarity, the value of objective function 9 before and after drawing $\{x^k\}$ are denoted as $\mathcal{L}_{before}$ and $\mathcal{L}_{after}$ respectively. The value of objective function 9 after the merger of $\{x^k\}$ into some cluster t_k^{new} is indicated by $\mathcal{L}_{new}$. In the Merge step, how to select an optimal cluster t_k^{new} for $\{x^k\}$ is equivalent to choose the minimum value change between $\mathcal{L}_{after}$ and $\mathcal{L}_{new}$, i.e. $t_k^{new} = \arg\min(\mathcal{L}_{after} - \mathcal{L}_{new})$. Here, we call the value change "merger cost", denoted by $d_{\mathcal{L}}$, which consists of two parts: the value change of within-task compression and cross-task regularization denoted by ΔI_1 and ΔI_2 separately. Thus, we write the total merger cost as: $d_{\mathcal{L}} = \Delta I_1 + \Delta I_2$.

Let each singleton cluster $\{x^k\}$ be merged into some cluster t_k and become a new cluster, i.e. $\{\{x^k, t_k\}\} \Rightarrow \tilde{t}_k$. Then we obtain

$$\begin{cases} p(\tilde{t}_k) = p(x^k) + p(t_k), \\ p(y|\tilde{t}_k) = \pi_1 \cdot p(y|x^k) + \pi_2 \cdot p(y|t_k), \\ \pi_1 = \dfrac{p(x^k)}{p(\tilde{t}_k)}, \pi_2 = \dfrac{p(t_k)}{p(\tilde{t}_k)} \end{cases} \tag{10}$$

where $1 \leq k \leq m$. Then, we can calculate the merger cost with respect to feature as follows.

$$\begin{aligned}&\Delta I_1 = \mathcal{L}_{after} - \mathcal{L}_{new} = \\ &\sum_{k=1}^{m} I(T_{after}^k;Y^k) - \sum_{k=1}^{m} I(T_{new}^k;Y^k) = \\ &\sum_{k=1}^{m} [I(T_{after}^k;Y^k) - I(T_{new}^k;Y^k)] = \sum_{k=1}^{m} \Delta I_{rel}.\end{aligned} \tag{11}$$

According to Eq. 10, we can get

$$\begin{aligned}&\Delta I_{rel} = p(x^k) \sum_y p(y|x^k) \log \frac{p(y|x^k)}{p(y)} + \\ &p(t^k) \sum_y p(y|t^k) \log \frac{p(y|t^k)}{p(y)} - \sum_y p(x^k)p(y|x^k) \log \frac{p(y|\tilde{t}^k)}{p(y)} \\ &- \sum_y p(t^k)p(y|t^k) \log \frac{p(y|\tilde{t}^k)}{p(y)} \\ &= p(x^k) \sum_y p(y|x^k) \log \frac{p(y|x^k)}{p(y|\tilde{t}^k)} + p(t^k) \sum_y p(y|t^k) \log \frac{p(y|t^k)}{p(y|\tilde{t}^k)} \\ &= p(x) D_{KL} \left[p(y|x^k)||p(y|\tilde{t}^k)\right] + p(t^k) D_{KL} \left[p(y|t^k)||p(y|\tilde{t}^k)\right] \\ &= \left[p(x^k) + p(t^k)\right] \cdot JS_{\Pi} \left[p(y|x^k), p(y|t^k)\right],\end{aligned} \tag{12}$$

where the JS_{Π} is the *Jensen-Shannon* divergence [19]. Because JS_{Π} is non-negative, we get $\Delta I_{com} \geq 0$. Next, we give the computation of the merger cost ΔI_{reg}.

$$\begin{aligned}&\Delta I_2 = \\ &\sum_{s=1}^{m} \lambda_s \cdot \sum_{t=1, t\neq s}^{m} [I(T_{after}^s;T_{after}^t) - I(T_{new}^s;T_{new}^t)].\end{aligned} \tag{13}$$

At each draw-and-merge step, we merge each data point x^k into some cluster t_k^{new} with the purpose of minimizing the information loss, i.e. $t_k^{new} = \arg\min d_{\mathcal{L}}$. It should be noted that there must be some information losses when x^k is merged into a new cluster, that is, $\Delta I_{reg} \geq 0$ and $d_{\mathcal{L}} \geq 0$. The details of MTIB are shown in Algorithm 1.

3.4. Complexity Analysis

Now, we focus on the complexity analysis of the proposed MTIB method, which consists of time complexity and space complexity. 1) Time complexity: at the step 9

Algorithm 1 The Multi-Task Information Bottleneck

```
Input: $m$ joint distributions $\{p(X^k, Y^k)\}_{k=1}^m$; Clustering number of each task $M$; The trade-off parameter $\lambda_{st}(1 \le s, t \le m)$.
Output: Partitions $\{T^k\}_{k=1}^m$.
Initialize: Random partitions of $\{\mathcal{X}^k\}_{k=1}^m$ into $M$ clusters $\{T^k\}_{k=1}^m$.
repeat
    $k \leftarrow 1$
    while $k \le m$ do
        for all $x^k \in X^k$ do
            Remove $x^k$ from current cluster $t_k(x^k)$;
            Reassign $x^k$ into different clusters in current task, and compute the merger cost $d_{\mathcal{L}}(\{x^k, t_k\})$ according to Eq. 9;
            Merge $x^k$ into cluster $t_k^{new}$ to such that $t_k^{new} = \arg\min_{t_k \in T^k} d_{\mathcal{L}}(\{x^k\}, t_k)$;
        end for
        $k \leftarrow k + 1$
    end while
until Convergence
```

of Algorithm 1, we compute the merger cost $d_{\mathcal{L}}$ for each t_k in every task which takes $O(lmM(|X^1|+\cdots+|X^m|)|Y|)$, where l is the number of iterations until MTIB converges to a stable solution, m and M are the number of tasks and clusters, respectively, which can be seen as constants. Since we construct a common vocabulary of all the tasks, the dimension of the relevant variables is same to each other, i.e. $|Y| = |Y^1| =, \cdots, = |Y^m|$. Note that the computation of mutual information between pairwise tasks takes $O(1)$. Therefore, the total time complexity of MTIB is $O(lmM(|X^1| + \cdots + |X^m|)|Y|)$. 2) Space complexity: the MTIB has to store the joint distributions of all tasks, so the space complexity is $O(|X^1||Y| + \cdots + |X^m||Y|)$.

4. Experiments

In this section, we will compare the proposed MTIB algorithm with 10 clustering algorithms on two kinds of data sets—*realistic* and *cross-view*. The competitive algorithms can be categorized into three classes. They are 1) Single-task clustering: K-Means (KM), Information Bottleneck (IB) [20]. All-KM and All-IB imply that KM and IB group all tasks into a single task respectively. 2) Multi-task clustering: Learn a Shared Subspace for MTC (LSSMTC) [2], Multi-task Bregman Clustering with Pairwise task regularization (MBC-P) [29], Multi-Task Multi-View Clustering (MTMVC) [31], convex Discriminative Multi-Task Relationship Clustering (DMTRC) [33]. 3) Human action clustering: Latent Dirichlet Allocation (LDA) [12], Dual Assignment K-Means (DAKM) [3]. The experiments of the competitive algorithms are run exactly with the authors' experimental settings. For the convex algorithm DMTRC, we perform it once under each parameter to select the best result, while all the other algorithms are executed 10 times to alleviate the influence caused by random initialization. We report the average evaluation with the metrics of Clustering Accuracy (ACC) and Normalized Mutual Information (NMI), as they are widely used in the literature [1].

4.1. Experimental Setup

To extract motion representation of the actions, we utilize the STIP with the detector of Harris 3D and the descriptor of HoG/HoF [5] for space-time interest point extraction and description. Then the popular BoVW framework is leveraged for feature representation. Differently, we implement the proposed agglomerative information maximization for common vocabulary generation. The dimensions of BoVW for all data sets are set as 1000. The λ_s $(1 \le s \le m)$ of MTIB is selected from the grid $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.

4.2. Results on Realistic Data Sets

In the realistic scenario, we utilize 4 data sets, separated into 2 groups of multi-task clustering evaluation, to verify the effectiveness of MTIB. 1) UCF50 [15] & HMDB [4]. UCF50 is an action recognition data set with 50 action categories, consisting of 6,000 realistic videos from the web. HMDB consists of 51 human actions with 6,766 videos, which has been collected from various sources, mostly movies. 2) Olympic [11] & YouTube [7]. Olympic contains 16 sports classes, with 50 sequences per class. YouTube contains 11 sports categories, with 1,168 sequences in total. All the realistic data sets are quite challenging due to large variations in camera motion, cluttered background, illumination conditions, etc. In this study, action clustering on each data set is treated as a learning task.

We show the performance of MTIB on realistic human action data set compared with different clustering methods in Table 1. From this table, several observations can be made. 1) The performances of ALL-KM and ALL-IB are not always better than their single-task version (KM and IB). This phenomenon illustrates that simply merging all tasks together for clustering may be harmful to each task and degrade the clustering quality. It is wise to characterize the shared information between tasks for improved clustering. 2) Most multi-task clustering methods obtain better performance than single-task algorithm. For instance, the DMTRC algorithm gets improvement 14.24% and 8.4% on ACC compared with KM. It demonstrates that exploiting the shared information among tasks can boost the clustering performance of each task. 3) MTIB can not only beat single-task clustering algorithms and their all-task versions, but also perform much better than all the multi-task clus-

Table 1. Clustering results on realistic data

	HMDB & UCF50				Olympic & YouTube			
	HMDB		UCF50		Olympic		YouTube	
	ACC(%)	NMI(%)	ACC(%)	NMI(%)	ACC(%)	NMI(%)	ACC(%)	NMI(%)
KM	19.25	33.36	35.26	56.49	31.82	33.36	36.15	36.29
IB	25.93	47.19	40.86	63.53	39.58	40.63	42.76	43.60
ALL-KM	18.09	30.29	28.68	47.76	30.21	30.40	26.12	22.06
ALL-IB	23.68	43.52	28.23	46.33	35.03	34.79	37.24	33.26
LSSMTC	17.00	34.85	19.80	40.25	32.27	30.47	33.94	31.13
MBC-P	20.08	39.02	25.99	48.14	35.21	34.93	34.52	34.27
MTMVC	23.05	42.07	34.14	56.90	38.49	37.05	36.76	36.97
DMTRC	26.30	48.14	40.28	62.59	46.06	32.45	44.55	32.37
LDA	24.50	44.67	34.00	55.96	38.03	38.08	39.85	40.03
DAKM	18.21	37.93	33.78	57.9	31.21	35.86	33.76	37.95
MTIB	**29.91**	**51.29**	**41.45**	**63.73**	**50.21**	**48.27**	**49.60**	**47.79**

tering algorithms. The bold values in the last row of Table 1 show that MTIB algorithm obtains best ACC and NMI compared with other clustering methods. This is mainly because that MTIB can discover the shared information between multiple tasks effectively.

To further verify the effectiveness of MTIB on human action clustering, we adopt two unsupervised human action categorization method as baselines, which are Latent Dirichlet Allocation (LDA) and Dual Assignment K-Means (DAKM). Niebles *et al*. [12] utilize LDA to learn the probability distributions of the spatial-temporal words and the intermediate topics corresponding to human action categories. Jones *et al*. [3] estimate the mutual information between two clusterings and use it to improve the results of each clustering simultaneously, which conducts unsupervised dual assignment clustering of human actions in context. Table 1 shows the ACC and NMI of MTIB compared with these two action clustering methods. As shown in this table, the performances of MTIB are much better than LDA and DAKM on all the tasks of realistic videos. So it verifies the effectiveness of MTIB on realistic videos.

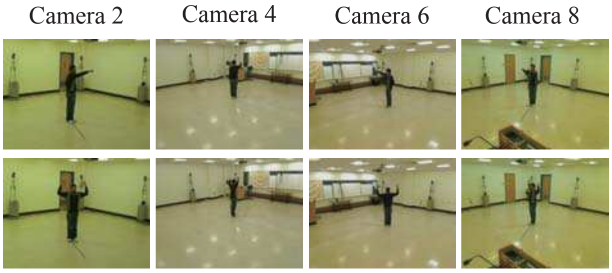


Figure 3. Exemplar frames from the WVU action data set. Each row shows one action viewed across four angles.

4.3. Results on Cross-view Data Sets

In the cross-view scenario, 2 cross-view action data sets are adopted for the evaluation of MTIB. 1) IXMAS [21], a well-known multi-view human action data set, consists of 11 different actions with the total of 1,148 video samples captured by 5 fixed cameras around the actors. Due to the unavoidable partial occlusion, we selected 4 views except the top-down view. 2) WVU [14] data set consists of 11 action patterns, each of which has 65 video sequences. This data set is obtained from a network of 8 embedded cameras, organized in a rectangular region, such that the cameras altogether can provide an overlapping coverage from various view directions. Figure 3 shows example frames in WVU data. In our experiment, we select the view 2, 4, 6, 8 as four tasks to evaluate our method. Action clustering on each viewpoint is treated as a learning task.

Table 2. ACC (%) comparison on cross-view data

	IXMAS				WVU			
	Task 1	Task 2	Task 3	Task 4	Task 1	Task 2	Task 3	Task 4
KM	31.85	36.48	30.12	39.70	30.28	31.51	32.02	31.28
IB	55.73	58.85	56.09	60.64	55.29	47.45	53.69	50.94
ALL-KM	23.52	20.52	13.12	19.61	31.85	22.63	26.98	24.62
ALL-IB	49.12	47.88	41.06	50.03	45.42	46.36	38.61	46.58
LSSMTC	29.39	26.49	26.07	24.73	33.35	30.72	31.03	35.75
MBC-P	29.75	27.21	27.91	25.18	33.46	31.34	33.20	38.64
MTMVC	51.00	53.49	53.27	52.49	47.11	43.57	44.35	48.05
DMTRC	52.73	57.58	56.06	58.48	**61.38**	53.38	60.92	56.77
LDA	37.76	41.00	34.45	47.67	50.37	45.48	50.65	47.97
DAKM	30.00	38.18	39.09	40.09	33.66	32.32	31.82	30.86
MTIB	**66.97**	**66.69**	**66.51**	**67.30**	61.32	**61.26**	**61.55**	**61.23**

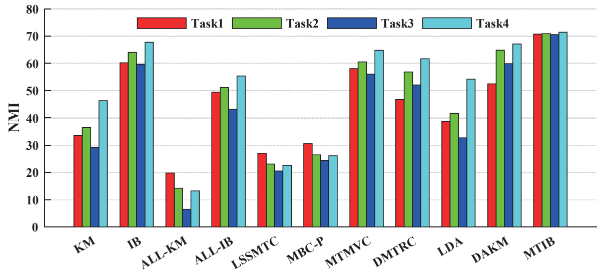


Figure 4. NMI (%) comparison of different methods on IXMAS.

Table 2 shows the results (ACC) obtained with different clustering methods. From this table, it is evident that the MTIB outperforms all three types of clustering algorithms, i.e., single-task, multi-task and action clustering. For instance, the convex method DMTRC obtains best ACC compared with all the competitive algorithms on the IXMAS and WVU as shown in Table 2. Compared with DMTRC, MTIB gets significant improvements (14.24%, 9.11%, 10.45% and 8.82%, respectively) on IXMAS data. For WVU data set, the MTIB also obtains improvements (7.88%, 0.63% and 4.46%, respectively) compared with DMTRC except for the task 1 (MTIB produces comparable performance). The same observations can be obtained from the NMI value in Figure 4 and Figure 5.

To further demonstrate the effectiveness of MTIB, we give the confusion matrices of DMTRC and MTIB on the four tasks of the IXMAS data set in Figure 6. From this figure, it is obvious that the learned categories of MTIB on all the four tasks are much purer than DMTRC algorithm. So we can conclude that MTIB algorithm can effectively discover meaningful action categories by exploiting shared information between multiple tasks.

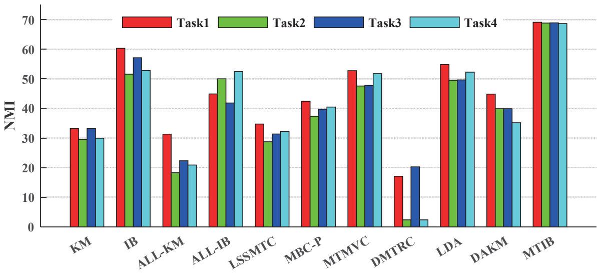
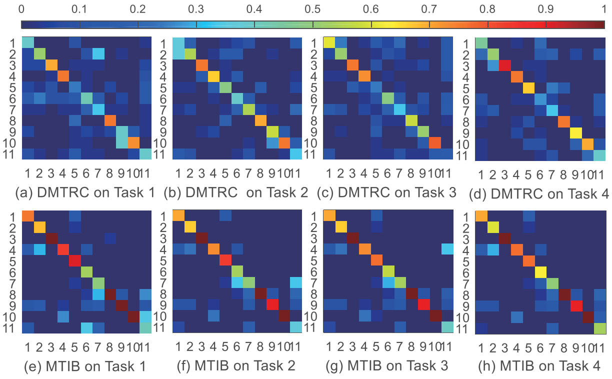


Figure 5. NMI (%) comparison of different methods on WVU.



(a) DMTRC on Task 1 (b) DMTRC on Task 2 (c) DMTRC on Task 3 (d) DMTRC on Task 4

(e) MTIB on Task 1 (f) MTIB on Task 2 (g) MTIB on Task 3 (h) MTIB on Task 4

Figure 6. Confusion matrices of DMTRC and MTIB on the four tasks of IXMAS data.

Table 3. ACC (%) comparison of MTIB with different vocabulary generation methods, i.e., AIM and KM.

	Realistic Data				Cross-view Data							
	Realistic-1		Realistic-2		IXMAS				WVU			
	T1	T2	T1	T2	T1	T2	T3	T4	T1	T2	T3	T4
KM	27.47	36.35	46.85	46.75	64.66	65.63	66.09	65.78	57.83	57.64	57.71	57.80
AIM	**29.91**	**41.45**	**50.21**	**49.60**	**66.97**	**66.69**	**66.51**	**67.30**	**61.32**	**61.26**	**61.55**	**61.23**

4.4. Exploration of Impact Factors

Common Vocabulary: In this study, we propose an agglomerative information maximization (AIM) to construct common vocabulary of multiple tasks. To test the impact of AIM, MTIB is performed on the common vocabulary constructed by AIM and KM respectively. Table 3 provides the ACC comparison results, in which the size of common vocabulary is set as 1000. From this table, we can observe that the MTIB based on AIM can get improvements on all the 12 tasks used in this study compared with these on KM. It is mainly because the high-level common vocabulary constructed by AIM is more discriminative than KM. So we can conclude that the common vocabulary constructed by AIM is more suitable to represent actions from multiple tasks.

Parameters: Since there are two tasks in realistic and four tasks in cross-view scenarios, we set all parameters equal to each other, i.e., $\lambda = \lambda_1 = \lambda_2$ for realistic data and $\lambda = \lambda_1 = \lambda_2 = \lambda_3 = \lambda_4$ for cross-view data. In this experiment, the values of λ vary from 0 to 1, with 0.1 as the change gap between adjacent values. From the Figure 7, we observe that the fluctuation of ACC value of MTIB on all tasks are typically slight, which demonstrates that MTIB is not sensitive to the trade-off parameters and the impact of parameters can be negligible.

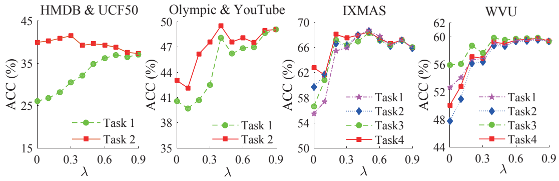


Figure 7. The performance of MTIB with parameter λ.

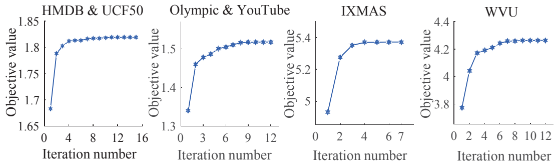


Figure 8. The convergence of MTIB on all used data sets.

Convergence: We investigate the convergence of the objective function of MTIB algorithm. Figure 8 shows the value of objective function of MTIB on all the four multi-task settings in this paper. We observe that the value of objective function of MTIB increases monotonically with each iteration, which shows that MTIB can converges to a optimal solution in a finite number of iterations.

5. Conclusion

This paper presents a novel multi-task information bottleneck (MTIB) method for discovering action patterns. Unlike previous methods, we exploited the shared information between multiple tasks in totally unsupervised setting. Specifically, to bridge the gap between multiple tasks, an agglomerative information maximization is proposed, which is general and can be beneficial to many multivariate problems. Then, the multi-task human action clustering is generally formulated as an information loss minimization function, in which the task relatedness can be quantified by the mutual information of clusters between different tasks. Extensive experiments on two kinds of challenging data sets, including realistic action data sets (HMDB & UCF50, Olympic & YouTube), and cross-view data sets (IXMAS, WVU), show that the proposed approach compares favorably to the state-of-the-art methods.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under grant No. 61170223, 61502434, 61672469.

References

[1] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *TKDE*, 17(12):1624–1637, 2005.

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

[3] S. Jones and L. Shao. Unsupervised spectral dual assignment clustering of human actions in context. In *CVPR*, pages 604–611, 2014.

[4] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011.

[5] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008.

[6] A. A. Liu, Y. T. Su, W. Z. Nie, and M. Kankanhalli. Hierarchical clustering multi-task learning for joint human action grouping and recognition. *TPAMI*, 39(1):102–114, 2016.

[7] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, pages 1996–2003, 2009.

[8] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, pages 1–8, 2008.

[9] Z. Ma, Y. Yang, F. Nie, N. Sebe, S. Yan, and A. G. Hauptmann. Harnessing lab knowledge for real-world action recognition. *IJCV*, 109(1):60–73, 2014.

[10] B. Mahasseni and S. Todorovic. Latent multitask learning for view-invariant action recognition. In *ICCV*, pages 3128–3135, 2013.

[11] J. C. Niebles, C. W. Chen, and F. F. Li. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, pages 392–405, 2010.

[12] J. C. Niebles, H. Wang, and F. F. Li. Unsupervised learning of human action categories using spatial-temporal words. *I-JCV*, 79(3):299–318, 2008.

[13] A. Pentina, V. Sharmanska, and C. H. Lampert. Curriculum learning of multiple tasks. In *CVPR*, pages 5492–5500, 2015.

[14] S. Ramagiri, R. Kavi, and V. Kulathumani. Real-time multi-view human action recognition using a wireless camera network. In *ICDSC*, pages 1–6, 2011.

[15] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vison and Applications (MVA)*, 24(5):971–981, 2013.

[16] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *ICCV*, pages 1–7, 2005.

[17] N. Slonim, N. Friedman, and Tishby. Multivariate information bottleneck. *Neural Computation*, 18(8):1739–1789, 2006.

[18] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *NIPS*, pages 617–623, 2000.

[19] Thomas and A. Joy. *Elements of information theory*. Wiley Interscinece, 1991.

[20] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Annual Allerton Conference on Communnication, Control and Computing*, pages 368–377, 1999.

[21] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *ICCV*, pages 1–7, 2007.

[22] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 104(2):249–257, 2006.

[23] C. Xu, D. Tao, and C. Xu. Large-margin multi-view information bottleneck. *TPAMI*, 36(8):1559–1572, 2014.

[24] X. Yan, Y. Ye, and X. Qiu. Unsupervised human action categorization with consensus information bottleneck method. In *IJCAI*, pages 2245–2251, 2016.

[25] Y. Yan, E. Ricci, G. Liu, and N. Sebe. Egocentric daily activity recognition via multitask clustering. *TIP*, 24(10):2984–2995, 2015.

[26] Y. Yan, E. Ricci, R. Subramanian, G. Liu, and N. Sebe. Multitask linear discriminant analysis for view invariant action recognition. *TIP*, 23(12):5599–611, 2014.

[27] Y. Yang, I. Saleemi, and M. Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *TPAMI*, 35(7):1635–1648, 2013.

[28] C. Yuan, W. Hu, G. Tian, S. Yang, and H. Wang. Multi-task sparse learning with beta process prior for action recognition. In *CVPR*, pages 423–429, 2013.

[29] J. Zhang and C. Zhang. Multitask bregman clustering. In *AAAI*, pages 655–660, 2010.

[30] X. Zhang and X. Zhang. Smart multi-task bregman clustering and multi-task kernel clustering. In *AAAI*, pages 1034–1040, 2013.

[31] X. Zhang, X. Zhang, and H. Liu. Multi-task multi-view clustering for non-negative data. In *IJCAI*, pages 4055–4061, 2015.

[32] X. Zhang, X. Zhang, and H. Liu. Self-adapted multi-task clustering. In *IJCAI*, pages 2357–2363, 2016.

[33] X. L. Zhang. Convex discriminative multitask clustering. *T-PAMI*, 37(1):28–40, 2015.

[34] Q. Zhou, G. Wang, K. Jia, and Q. Zhao. Learning to share latent tasks for action recognition. In *ICCV*, pages 2264–2271, 2013.

[35] F. Zhu and L. Shao. Weakly-supervised cross-domain dictionary learning for visual recognition. *IJCV*, 109(1):42–59, 2014.