Learning to Share Latent Tasks for Action Recognition

Qiang Zhou[1]* Gang Wang[2,3] Kui Jia[3] Qi Zhao[1]
[1] National University of Singapore, Singapore
[2]Nanyang Technological University, Singapore
[3]Advanced Digital Sciences Center, Singapore
zhouqiang@nus.edu.sg wanggang@ntu.edu.sg chris.jia@adsc.com.sg eleqiz@nus.edu.sg

Abstract

Sharing knowledge for multiple related machine learning tasks is an effective strategy to improve the generalization performance. In this paper, we investigate knowledge sharing across categories for action recognition in videos. The motivation is that many action categories are related, where common motion pattern are shared among them (e.g. diving and high jump share the jump motion). We propose a new multi-task learning method to learn latent tasks shared across categories, and reconstruct a classifier for each category from these latent tasks. Compared to previous methods, our approach has two advantages: (1) The learned latent tasks correspond to basic motion patterns instead of full actions, thus enhancing discrimination power of the classifiers. (2) Categories are selected to share information with a sparsity regularizer, avoiding falsely forcing all categories to share knowledge. Experimental results on multiple public data sets show that the proposed approach can effectively transfer knowledge between different action categories to improve the performance of conventional single task learning methods.

1. Introduction

Human action recognition is an important problem in computer vision and numerous methods have been proposed to tackle it [13, 3, 17, 31, 28, 14, 25, 30]. This work builds on a key observation that many action categories are highly correlated, as can be seen from published action data sets [21, 17]. For example, people playing different kinds of musical instruments in UCF50 [21] share similar motion patterns. Not much work in the literature, however, has been devoted to the understanding of shared knowledge in human actions. In this paper, we explore this particular problem of learning knowledge sharing in action recognition in a multi-task learning framework. Multi-task learning has been shown to improve the generalization capability of each single task in the machine learning community [4, 6, 33]. To be specific, we attempt to learn a large number of latent tasks shared by all the categories, and represent each action classifier as a linear combination of latent tasks. The proposed method automatically infers common visual knowledge (corresponding to latent tasks) that is sharable and finds the optimal linear combination of latent tasks to reconstruct each category model. Different from the existing works that use multi-task learning algorithms for other tasks, say text categorization, the use of respective methods in action recognition need to address the following distinct features in action recognition:

(1) In action recognition, the latent tasks should correspond to some basic motion patterns that can be most effectively shared among different categories. If too much "holistic" information is shared, then discrimination capability is compromised. To effectively address this issue, we formulate our model by enforcing ℓ_1 norm regularization on the parameter vectors of latent tasks. With the ℓ_1 regularization, most entries of the feature vectors would be zeros, and the remaining non-zero elements are expected to represent important motion patterns. This can be interpreted as a process of feature selection. In contrast, previous multi-task learning methods only enforce ℓ_2 norm regularization on the latent task model parameters to avoid overfitting. ℓ_2 norm regularization , however, does not have the feature selection capability.

(2) Most previous works [6, 1] assume that all the tasks are related, which is invalid for action recognition. For example, in the UCF 50 data set, playing musical instruments actions are different from sports actions. Forcing all the tasks to be relevant would simply introduce noise to the learned latent tasks. To approach this problem, we introduce the ℓ_1 norm sparsity regularizer on the combination weight parameter of each category and each action model is reconstructed using a few latent tasks. Consequently, in most cases, a latent task is shared by a small number of cat-

*Most of this work was performed while the first author was a research engineer at Advanced Digital Sciences Center egories. This way allows only related categories to share information, rather than forcing all the categories to share latent tasks. Relationship between any two categories can be determined according to the overlapping of their combination weights which are automatically learned from training data.

To summarize, this work proposes a new multi-task learning method to share latent tasks across categories. The new method can effectively learn discriminative latent tasks and automatically select combination weights for each category model. To learn the model parameters, we adopt an efficient alternating optimization algorithm based on the accelerated proximal gradient (APG) method[27], and extensive experiments on multiple public data sets are carried out which demonstrates the effectiveness of the approach.

2. Related Work

Action Recognition. In the last decade, there is an abundant literature on action recognition in videos [13, 15, 16, 17, 14, 31, 28, 25]. Among them, discriminative part-based action models [31, 17, 25, 12] attract a lot of attention recently. In particular, [31] attempts to model dependence between local patches in the spatial domain and [17, 25] learn structure among motion segments in the temporal domain. All these works learn a model for each category independently while our approach focuses on sharing visual knowledge for multiple categories via a multi-task learning method. Recently, there are some works which attempt to share information for action recognition. In [3], Cao et al. propose to train action models on unlabled target data set by modeling the correlation between labeled source data set and unlabeled target data set. Liu et al. [14] exploit attribute representation which is shared across categories. However, these attributes are manually specified. Furthermore, all these methods focus on learning an action model for each category independently. Yao et al. [32] learn the latent basis by ℓ_1 regularization for action recognition in still images. Their goal is to model high-order interactions of image attributes and parts. Different from their work, our approach attempts to share visual knowledge among multiple categories and improve the performance of action recognition.

Knowledge Sharing for Object Recognition. In object recognition, a number of papers have been published on transferring visual knowledge between different object categories [23, 18, 5, 19, 24, 26]. Motivated by the fact that some object parts may have similar appearance from different views, Ott et al. [18] propose to extend the deformable part model [7] to share object part models among multiple mixture components and object classes. However, they assume that all part models are shared and this may introduce additional noise. On contrast, our model includes a ℓ_1 regularization term which enables our model to selectively share latent tasks across categories. Endres et al. [5] consider a more complex sharing scheme with a two level information sharing structure. On the top level, body plans are shared across object categories, and on the bottom level, these body plans share object part appearance models. However, the learning procedure heavily relies on additional supervision such as object part annotation. In [8], Harchaoui et al. incorporate low-rank regularization for large-scale multi-class object recognition. Trace-norm penalty in their formulation enforces all categories are related which may degrade the performance in real-world problem. In contrast, sparse combination weights in our model will make task sharing among all categories more flexible. Given the great success of visual knowledge sharing in object recognition, we believe it is also a promising research direction in action recognition.

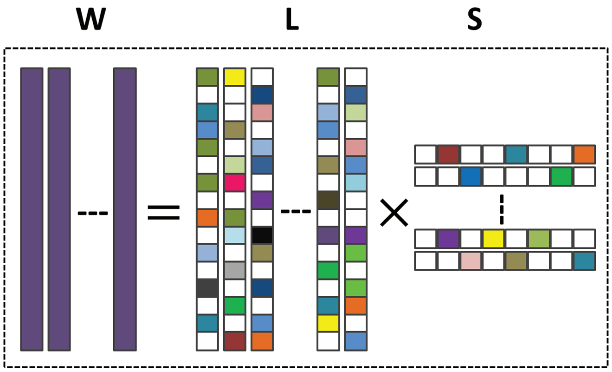

Figure 1. Illustration of our approach, where $\mathbf{W}$ denote model parameters of all the categories. $\mathbf{L}$ and $\mathbf{S}$ denotes latent tasks matrix and sparse combination weight matrix, respectively. White blocks represent zero-value entries. In this work, we learn the latent task matrix $\mathbf{L}$ and the combination weight matrix $\mathbf{S}$ instead of learning $\mathbf{W}$ directly.

Multi-Task Learning. Multi-task learning (MTL) [4] has been an active topic in machine learning for a long time. Most previous multi-task works [4, 6, 1] assume that all the tasks are related to each other or the tasks are related under certain prior assumptions, such as the tree-guided MTL [9], the clustered MTL [33], etc. We argue that these imposed prior assumptions are too strong for many practical problems. In this paper, we introduce a more flexible latent tasks sharing scheme for action recognition in videos. Our work is related to [11], but different from it on latent task modeling and optimization methods. Compared with their method, our model is equipped with the feature selection capability due to the use of the ℓ_1 normalization method to regularize the latent task model parameters. Therefore, our approach enforces the learned latent tasks to correspond to basic motion patterns, which can be more effectively shared across different activity categories.

3. Action Recognition with Sharing Latent Tasks

In this section, we describe our approach for action recognition by sharing latent tasks across categories. The generalization capability of each single model is improved by leveraging visual knowledge from other categories.

3.1. Learning to Share Latent Tasks

Suppose we have C action categories and our goal is to learn a binary linear classifier for each category. For the c-th action class, we denote its model parameter as w_c and the corresponding training data are $\{(X_{ci}, Y_{ci})\}_{i=1}^{N_c} \subset \mathbb{R}^d \times \{-1, +1\}(c = 1, \cdots, C)$, where ci and N_c are the index and the number of training data of the c-th class, respectively.

We attempt to learn shared tasks together for improved action recognition in the multi-task learning framework. Therefore, instead of training each classifier separately, we propose to learn classifiers for all the categories simultaneously. To be specific, we assume that all classifiers can be reconstructed from a number of shared latent tasks, and use a linear combination of latent tasks to reconstruct each classifiers. Let $\mathbf{L} = [L_1, L_2, \cdots, L_K] \in \mathbb{R}^{d\times K}$ denotes the shared latent task matrix with each column representing a latent task in $\mathbb{R}^d$ and K is the number of latent tasks. We write $s_c \in \mathbb{R}^K$ as the combination weight parameter for the c-th category. The model parameter of the c-th category can be expressed as

$$w_c = \mathbf{L}s_c \tag{1}$$

Model parameters of all the categories can be put together to form a large matrix $\mathbf{W} = [w_1, w_2, \cdots, w_C] \in \mathbb{R}^{d\times C}$. Similar manipulations can also be done for combination weights to form $\mathbf{S} = [s_1, s_2, \cdots, s_C] \in \mathbb{R}^{K\times C}$. Then we can obtain the following formulation:

$$\mathbf{W} = \mathbf{L}\mathbf{S} \tag{2}$$

Consequently, we will learn the latent task matrix $\mathbf{L}$ and the combination weight matrix $\mathbf{S}$ instead of learning $\mathbf{W}$ directly. This method enables different action categories to share similar visual pattern which are represented by latent tasks.

Regularization is critical for learning a robust model, we apply the ℓ_2 norm regularization on all the latent task model parameters to avoid overfitting. In the context of action recognition, we expect latent tasks to represent basic motion patterns that can be shared among categories. Discriminative information is lost if categories share too much holistic information. One possible method is to model each category as a set of "parts", and let different categories share the common parts. This method, however, may require a rather complicated model. Alternatively, we apply feature selection methods that force each latent task to respond only to particular feature patterns and obtain shareable latent tasks. In this paper, we apply the sparsity regularizer ℓ_1 norm to force most of the dimensions to be 0. Hence, the remaining parameters are expected to represent basic visual patterns.

While previous methods usually assume that all categories are related to each other, this work enforces latent tasks to be selectively shared by different categories. To achieve this, formally, we apply the ℓ_1 norm regularization on the matrix of combination weight $\mathbf{S}$. As a result, each category model is reconstructed by small number of latent tasks, which forces latent tasks to be shared only among related categories.

We propose a new multi-task learning approach to learn multiple classifiers simultaneously by sharing latent task across categories. Based on the above motivation, our learning problem is formulated as follows:

$$\min_{\mathbf{L},\mathbf{S}} \sum_{c=1}^{C} \sum_{i=1}^{N_c} \frac{1}{2}\mathcal{L}(\mathbf{Y}_{ci}, (\mathbf{L}s_c)^T \mathbf{X}_{ci}) + \mu\|\mathbf{S}\|_1 + \lambda\|\mathbf{L}\|_F^2 + \gamma\|\mathbf{L}\|_1 \tag{3}$$

The first term $\mathcal{L}(\cdot,\cdot)$ represents a pre-defined loss function. In this paper, we adopt the squared hinge loss which is defined as:

$$\mathcal{L}(\mathbf{Y}_{ci}, (\mathbf{L}s_c)^T\mathbf{X}_{ci}) = [\max(0, 1 - \mathbf{Y}_{ci}(\mathbf{L}s_c)^T\mathbf{X}_{ci})]^2$$

The second term $\|\mathbf{S}\|_1 = \sum_{c=1}^{C} \|s_c\|_1$ denotes the ℓ_1 norm of the linear combination weight for each category. This regularization term enables us to learn a sparse linear combination for each category.

The last two terms are Frobenius norm and ℓ_1 norm of $\mathbf{L}$ which are defined as $\|\mathbf{L}\|_F^2 = trace\left(\mathbf{L}\mathbf{L}^T\right)$ and $\|\mathbf{L}\|_1 = \sum_{k=1}^{K} \|L_k\|_1$, respectively. Frobenius norm of $\mathbf{L}$ helps to avoid overfitting while ℓ_1 norm of each latent task forces to focus on specific motion patterns rather than the full actions. μ, λ and γ are regularization parameters. Fig. 1 shows the illustration of our approach.

Pirsiavash et al. [20] proposed a bilinear classifier for visual recognition. Our work, however, is significant different from their work in motivation and formulation. Their work mainly focus on reducing the number of parameters of a weight vector and improving run-time efficiency, while our goal is a more effective method to share knowledge across categories. Therefore, we enforce the latent tasks to correspond to basic patterns (instead of full actions) so that they be shared by more related categories. Furthermore, in our work, each category only selects a few latent tasks, avoiding sharing knowledge with unrelated categories.

After learning latent tasks matrix $\mathbf{L}$ and the combination weight matrix $\mathbf{S}$, we can obtain a linear classifier for each category by Eq. 1. For a new testing sample, we calculate decision values to all categories by running all the category classifier on it, and then choose the category that results in the largest decision value to be the predicted label.

3.2. Model Learning

Eq. (3) is not jointly convex in $\mathbf{S}$ and $\mathbf{L}$. However, it is convex in $\mathbf{S}$ given fixed $\mathbf{L}$, and is convex in $\mathbf{L}$ given fixed $\mathbf{S}$. Hence we adopt the block coordinate descent method to solve this problem by alternately optimizing $\mathbf{S}$ and $\mathbf{L}$. Our optimization procedure can be outlined as two steps:
(1) with the fixed $\mathbf{L}$, learn the combination weight matrix $\mathbf{S}$ by solving the following optimization problem:

$$\min_{\mathbf{S}} \sum_{c=1}^{C} \sum_{i=1}^{N_c} \frac{1}{2} \mathcal{L}(\mathbf{Y}_{ci}, (\mathbf{L}s_c)^T \mathbf{X}_{ci}) + \mu \|\mathbf{S}\|_1 \tag{4}$$

(2) with the fixed $\mathbf{S}$, the optimal $\mathbf{L}$ to Eq. (3) can be obtained by solving the following optimization problem:

$$\min_{\mathbf{L}} \sum_{c=1}^{C} \sum_{i=1}^{N_c} \frac{1}{2} \mathcal{L}(\mathbf{Y}_{ci}, (\mathbf{L}s_c)^T \mathbf{X}_{ci}) + \lambda \|\mathbf{L}\|_F^2 + \gamma \|\mathbf{L}\|_1 \tag{5}$$

Both optimization problems in Eq. 4 and Eq. 5 are non-smooth due to the ℓ_1 norm regularization of $\mathbf{S}$ and $\mathbf{L}$. We employ the accelerated proximal gradient method (APG) [27] in both two steps. Different from traditional gradient descend methods, at each iteration, APG uses a linear combination of previous two points as the search point, instead of only using the latest point. APG has the convergence rate of $O(\frac{1}{k^2})$ [27], which is most optimal among all the first order methods. Furthermore, it can also deal with non-smooth convex optimization problem with proximal operator. Following are the details of the optimization procedure.

Optimizing $\mathbf{S}$ with fixed $\mathbf{L}$. After fixing the latent task matrix $\mathbf{L}$, the objective function in Eq. (4) is a non-smooth convex function. For simplicity, we represent Eq. (4) as

$$\min_{\mathbf{S}} f(\mathbf{S}) + g(\mathbf{S}) \tag{6}$$

where the functions $f(\mathbf{S})$ and $g(\mathbf{S})$ are defined respectively as:

$$f(\mathbf{S}) = \sum_{c=1}^{C} \sum_{i=1}^{N_c} \frac{1}{2} \mathcal{L}(\mathbf{Y}_{ci}, (\mathbf{L}s_c)^T \mathbf{X}_{ci})$$

$$g(\mathbf{S}) = \mu \|\mathbf{S}\|_1$$

We note that $f(\mathbf{S})$ is a smooth convex function and $g(\mathbf{S})$ is a convex but non-smooth function. In APG, given the search point $\hat{\mathbf{S}}^t$ and the gradient of smooth part $\nabla_{\mathbf{S}} f(\hat{\mathbf{S}}^t)$ for the t-th iteration, we consider the following update scheme [2] for problem Eq. (6):

$$\mathbf{S}^t = \mathcal{T}_{\frac{\mu}{V}} \left(\hat{\mathbf{S}}^t - \frac{1}{V} \nabla_{\mathbf{S}} f(\hat{\mathbf{S}}^t) \right) \tag{7}$$

where $\mathcal{T}_\alpha$ is the shrinkage operator defined by

$$\mathcal{T}_\alpha(x_i) = (|x_i| - \alpha)_+ \operatorname{sgn}(x_i)$$

V is the Lipschitz constant and we calculate it by the backtracking line search method.

As mentioned above, APG uses the linear combination of two previous points as a search point for the next iteration. Specifically, given two previous points $\mathbf{S}^{t-1}$ and $\mathbf{S}^{t-2}$, the search point $\hat{\mathbf{S}}^t$ at the t-th iteration is $\left(\mathbf{S}^{t-1} + \left(\frac{p^{t-2}-1}{p^{t-1}}\right)(\mathbf{S}^{t-1} - \mathbf{S}^{t-2})\right)$, where p is initialized as 1 and updated as $p^t = \frac{1+\sqrt{1+4(p^{t-1})^2}}{2}$.

Algorithm 1: Solving Optimization Problem Eq. 3 by Accelerated Proximal Gradient (APG)

```
Input: Training data: $\mathbf{D}_c = \{(\mathbf{X}_{ci}, \mathbf{Y}_{ci})\}$
Output: Latent task matrix $\mathbf{L}$
        Combination weight matrix $\mathbf{S} = [s_1, \ldots, s_C]$
        Model parameters $\mathbf{W} = \mathbf{LS}$
Step 1. Optimize $\mathbf{S}$ with fixed $\mathbf{L}$
repeat
    $\hat{\mathbf{S}}^m = \mathcal{T}_{\frac{\mu}{V}}\left(\mathbf{S}^{m-1} + \frac{p^{m-2}-1}{p^{m-1}}(\mathbf{S}^{m-1} - \mathbf{S}^{m-2})\right)$
    $p^m = \frac{1+\sqrt{1+4(p^{m-1})^2}}{2}$
until Converged;
Step 2. Optimize $\mathbf{L}$ with fixed $\mathbf{S}$
repeat
    $\hat{\mathbf{L}}^n = \mathcal{T}_{\frac{\gamma}{V}}\left(\mathbf{L}^{n-1} + \frac{p^{n-2}-1}{p^{n-1}}(\mathbf{L}^{n-1} - \mathbf{L}^{n-2})\right)$
    $p^n = \frac{1+\sqrt{1+4(p^{n-1})^2}}{2}$
until Converged;
Step 3. Repeat Step 2 and Step 3 until Eq. (3)
Converged
```

Optimizing $\mathbf{L}$ with fixed $\mathbf{S}$. For fixed $\mathbf{S}$, the optimization of Eq. (5) is similar with $f(\mathbf{L})$ and $g(\mathbf{L})$ becomes

$$f(\mathbf{L}) = \sum_{c=1}^{C} \sum_{i=1}^{N_c} \frac{1}{2} \mathcal{L}(\mathbf{Y}_{ci}, (\mathbf{L}s_c)^T \mathbf{X}_{ci}) + \lambda \|\mathbf{L}\|_F^2$$

$$g(\mathbf{L}) = \gamma \|\mathbf{L}\|_1$$

Given search point $\hat{\mathbf{L}}^t$ and the gradient of smooth part $\nabla_{\mathbf{L}} f(\hat{\mathbf{L}}^t)$ at the t-th iteration , we obtain $\mathbf{L}^t$ as

$$\mathbf{L}^t = \mathcal{T}_{\frac{\gamma}{V}} \left(\hat{\mathbf{L}}^t - \frac{1}{V} \nabla_{\mathbf{L}} f(\hat{\mathbf{L}}^t) \right) \tag{8}$$

Model Initialization. The first step of our optimization algorithm is to initialize the latent task matrix $\mathbf{L}$. In this paper, we first train a linear SVM classifier for each category independently. Suppose w_c is the SVM classifier the c-th category and all the classifiers are denoted by

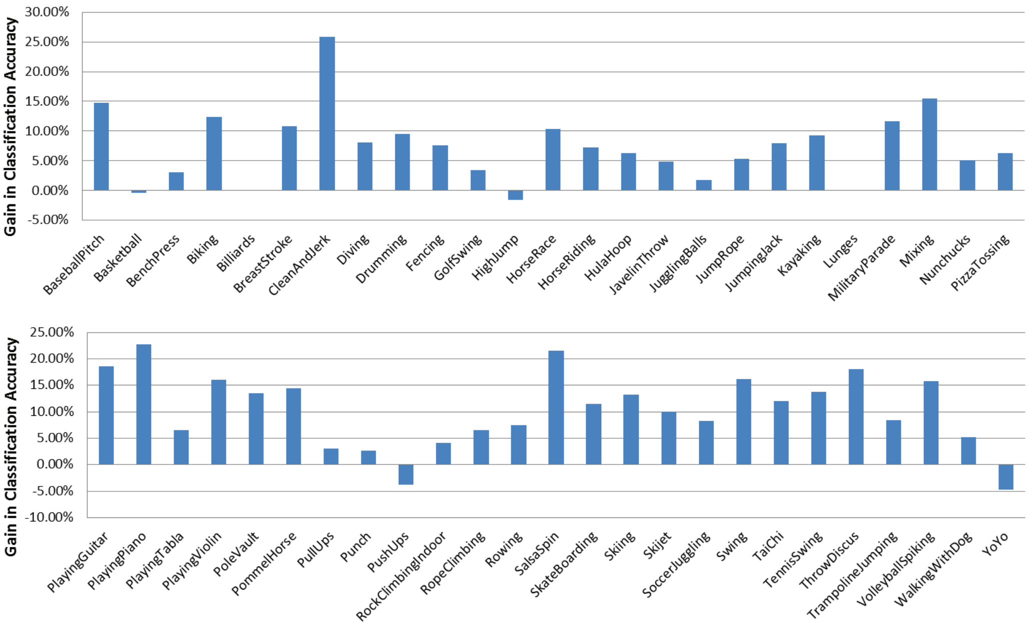

Figure 2. Classification accuracy gain of each category by sharing latent tasks across categories on the UCF50 data set when using only 25% of training data.

$\mathbf{W} = [w_1, w_2, \cdots, w_C]$, we compute the singular value decomposition (SVD) for $\mathbf{W}$ to obtain $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. We employ the first K columns of $\mathbf{U}$ to initialize the latent tasks matrix $\mathbf{L}$. For the combination weight matrix $\mathbf{S}$, we randomly generate a matrix as its initialization. This simple initialization method has been shown to work well in our experiments.

Our overall optimization procedure is summarized in Alg. 1.

4. Experiments

4.1. Implementation Details

Features. Motivated by recent success in dense trajectory [28] in action recognition, we adopt this feature in our experiments. Specifically, it includes four types of descriptors: Histograms of Oriented Gradients (HOG), Histograms of Optical Flow (HOF), Motion Boundary Histogram (MBH) and Trajectory. We then use Locality-constrained Linear Coding (LLC) [29] to encode these extracted local features. Following [28], we randomly select 100,000 features and build a codebook with 4000 words for each descriptor. LLC coefficients of the four descriptors are concatenated to form the final feature descriptor to represent each video.

Bias Terms in Latent Tasks. In order to make the scores of multiple latent tasks comparable when they are combined to form a category classifier, we introduce a bias term for each latent task. We implement this by augmenting each feature vector with one constant (1 is used in this paper).

Percentage	25%	50%	75%	100%
STL	54.2 ± 2.9	64.5 ± 1.0	71.7 ± 1.2	75.7
This Work	63.2 ± 2.2	73.2 ± 1.0	76.8 ± 1.7	80.2
Gain	9.0	8.7	5.1	4.5

Table 1. Average accuracy and standard deviation (%) of our approach and single task learning (STL) on the UCF50 data set with varying number of training samples.

Parameters. The regularization λ in Eq. 3 is set as 0.4 in all experiments. Other two parameters μ and γ are chosen by a cross validation procedure. **Baseline.** We compare our approach with the single task learning (STL) methods, in which no task sharing is enforced and all classifiers are learned separately. Specifically, we employ the linear SVM classifier as the single-task learning method, which has been shown very good performance on visual recognition with LLC representation [29].

4.2. Experiments on UCF50 Action Data Set

UCF50 [21] is one of the largest public action data sets. It contains 50 action categories with a total of 6617 action videos. This data set is created by collecting realistic action video from Youtube. We adopt the same experiment setup in [22] to repeat the experiment for 5 times. Moreover, we also test the performance of the proposed method with different numbers of training samples. In particular, we con-

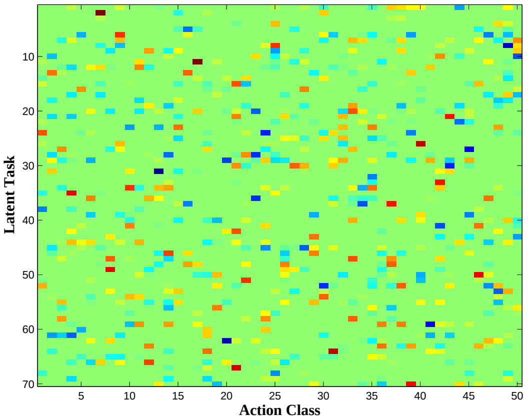

Figure 3. Sparsity pattern (the sparse weight matrix S) learned by our approach on the UCF50 action data set.

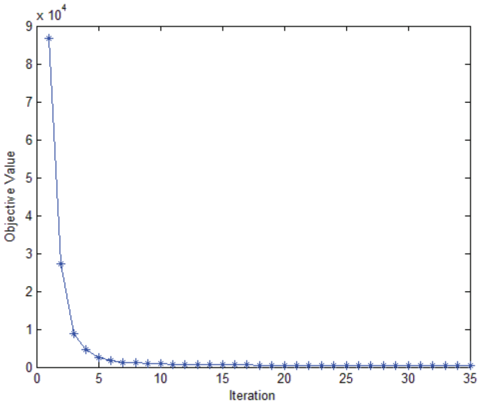

Figure 4. A convergence curve of the model learning algorithm on UCF50. It takes less than 20 iterations to reach convergence.

duct the experiments with 25%, 50%, 75% and 100% of training samples and the results are presented in Tabel 1.

As shown in Table 1, the proposed multi-task learning method outperforms the single task learning in all the settings. Interestingly, the proposed method achieves a notably larger gain with less training samples, (i.e., a gain of about 9% when the proportion of training sample is less than 50%). Intuitively, a big part of performance improvement comes from the fact that the knowledge sharing mechanism amounts to increasing the number of training data for each category. The positive samples for learning a shared task is the sum of those from all categories that share the task, thus the advantage is particular notable with a small number of training samples. Fig. 2 shows the classification accuracy gain of each category when we only use 25% of the training data. As demonstrated in Fig. 2, the proposed method achieves remarkable improvement on almost all the categories. For example, all actions in the group of playing instruments receive more than 5% gain due to sharing tasks, largely due to the fact that these categories are more related to each other, therefore gaining benefits by sharing information. The performance of PlayingGuitar, PlayingPiano and PlayingViolin is increased for more than 15%. We also show one example of the learned linear combination weights **S** in Fig. 3. From Fig. 3, we can see that each action model is sparsely reconstructed as expected. Fig. 4 shows one example of the convergence curve of our model learning algorithm. It takes less than 20 iterations to reach the convergence.

Method	Accuracy
Laptev et al. [13]	47.9%
Sadanand and J. Corso[22]	57.9%
Kliper-Gross et al.[10]	68.5%
Wang et al. [28]	75.7%
Our Method	**80.2%**

Table 2. Performance comparison with some several state-of-the-art approaches on the UCF50 data set in terms of accuracy. All these results are obtained using the same data split scheme with all training data.

In Tabel 2, we compare the proposed approach with some state-of-the-art methods on the UCF50 data set. We note that all these results obtained by using the same experiment setup in [22].

4.3. Experiments on Olympic Sports Data Set

The Olympic Sports data set [17] is collected from Youtube video and consists of 16 Olympic sport actions. We follow the original experiment setup suggested by [17] that uses 649 video clips for training and the other 134 video clips as test set. We also compare the proposed method with single task learning by changing the size of training data. In our experiments, we randomly select 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% of the positive and negative videos respectively in the training data for each category. For each setting, we repeat the experiments for 10 times by randomly selecting training examples. We report average classification accuracy over categories as a performance measure, and show all results in Table 3. Fig. 5 shows the detailed comparison between the proposed method and single task learning methods using 40% of the training data.

For comparison with state-of-the-art approaches, we also evaluate the mean average precision (which differs from the average accuracy) for all categories. Table 4 shows a comparison over different approaches in mean average precision.

4.4. Effect of Different Number of Latent Tasks

We analyze the effect of different number of latent tasks on the UCF50 data set using 25% of the training data. Fig. 6 illustrates the results of our approach with different num-

Percentage	10%	20%	30%	40%	50%
STL	33.1 ± 3.7	45.8 ± 4.6	50.0 ± 4.9	53.0 ± 2.3	57.3 ± 2.5
This Work	40.1 ± 5.2	52.4 ± 4.0	57.3 ± 2.8	62.6 ± 2.5	66.1 ± 1.8
Gain	7.0	6.6	7.3	9.6	8.8
Percentage	**60%**	**70%**	**80%**	**90%**	**100%**
STL	61.0 ± 3.7	65.0 ± 2.59	65.0 ± 2.7	66.9 ± 1.3	68.9
This Work	67.0 ± 3.7	69.4 ± 2.64	70.9 ± 2.0	72.3 ± 1.5	73.6
Gain	6.0	4.4	5.9	5.4	4.7

Table 3. Average accuracy and standard deviation (%) of our approach and single task learning (STL) on the Olympic Sports data set with a varying number of training samples.

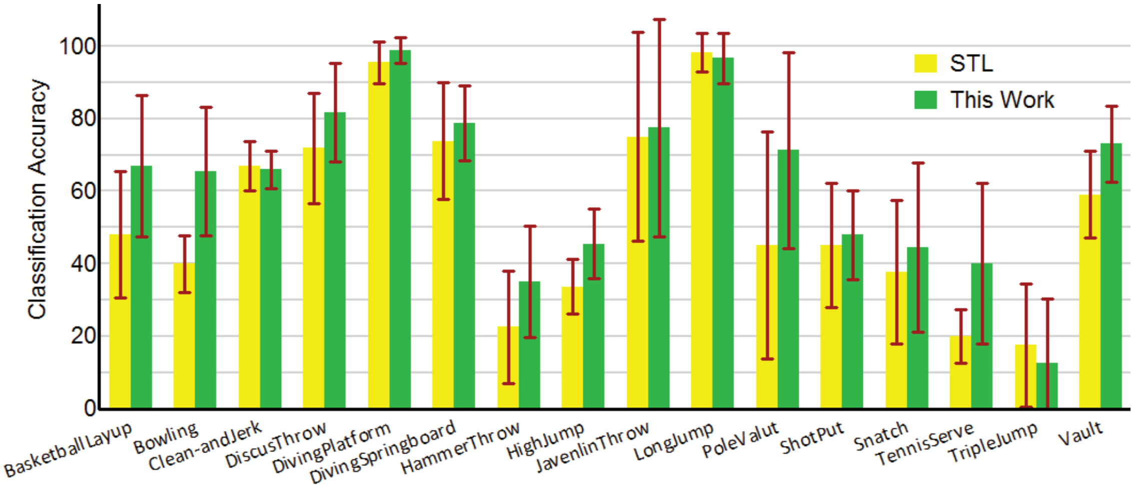

Figure 5. Detailed comparison between our method and single task learning methods on the Olympic Sports data set with 40% of the training data. We show the classification accuracy of each category.

Method	Mean Average Precision
Laptev et al. [13]	62.0%
Tang et al [25]	66.8%
Niebles et al. [17]	72.1%
Liu et al. [14]	74.4%
Wang et al. [28]	74.1%
Our Method	**78.3%**

Table 4. Performance comparison with several state-of-the-art approaches on the Olympic Sports data set in terms of mean average precision. All these results are obtained using the same data split scheme with all training data.

ber of latent tasks. According to Fig. 6, the classification accuracy increases with the number of latent tasks, potentially due to the finer visual patterns captured by more latent tasks. The larger the number, the higher the computational cost though. In our experiments, the number of latent tasks is determined empirically.

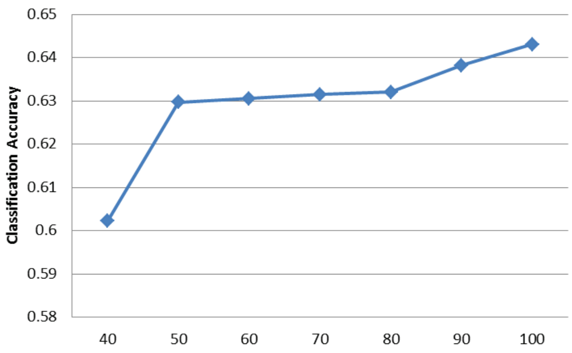

Figure 6. Classification performance of different number of latent tasks on the UCF50 data set using 25% of the training data. Seven different sizes have been tested: 40, 50, 60, 70, 80, 90 and 100.

4.5. Effect of Regularization Terms

We evaluate the effect of regularization terms $\|\mathbf{L}\|_1$ and $\|\mathbf{S}\|_1$ in our model on the UCF50 data set using 25% of the training data and on the Olympic Sports data set. Fig. 7

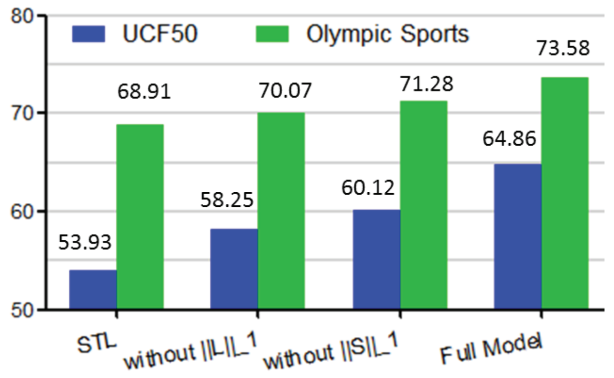

Figure 7. Classification accuracy of different methods (STL, our model without $\|\mathbf{L}\|_1$, our model without $\|\mathbf{S}\|_1$ and our full model) on the UCF50 data set using 25% of the training data and on the Olympic Sports data set.

compares the results of STL, our model without $\|\mathbf{L}\|_1$, our model without $\|\mathbf{S}\|_1$, and our full model. Without regularization terms $\|\mathbf{L}\|_1$ or $\|\mathbf{S}\|_1$, the performance degrades significantly.

5. Conclusions and Discussions

In this work, we have proposed an approach to share latent tasks for action recognition. Extensive experiments on multiple action data sets show that the proposed approach outperforms single task learning methods, especially when only a small number of training examples are available. For future work, we plan to investigate how to develop convex formulation for sharing latent tasks since the current formulation is not convex.

Acknowledgments

This study is supported by the research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR), and is also supported by the start-up grant at ECE of NUS (No.R-263-000-648-133).

References

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

[3] L. Cao, Z. Liu, and T. S. Huang. Cross-dataset action detection. In *Proc. CVPR*, 2010.

[4] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[5] I. Endres, V. Srikumar, M.-W. Chang, and D. Hoiem. Learning shared body plans. In *Proc. CVPR*, 2012.

[6] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proc. KDD*, 2004.

[7] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. PAMI*, 32(9):1627–1645, 2010.

[8] Z. Harchaoui, M. Douze, M. Paulin, M. Dudík, and J. Malick. Large-scale image classification with trace-norm regularization. In *Proc. CVPR*, 2012.

[9] S. Kim and E. P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proc. ICML*, 2010.

[10] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *Proc. ECCV*, 2012.

[11] A. Kumar and H. D. III. Learning task grouping and overlap in multi-task learning. In *Proc. ICML*, 2012.

[12] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori. Discriminative latent models for recognizing contextual group activities. *IEEE Trans. PAMI*, 34(8):1549–1562, 2012.

[13] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008.

[14] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Proc. CVPR*, 2011.

[15] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proc. CVPR*, 2009.

[16] B. Ni, S. Yan, and A. A. Kassim. Recognizing human group activities with localized causalities. In *Proc. CVPR*, 2009.

[17] J. C. Niebles, C.-W. Chen, and F.-F. Li. Modeling temporal structure of decomposable motion segments for activity classification. In *Proc. ECCV*, 2010.

[18] P. Ott and M. Everingham. Shared parts for deformable part-based models. In *Proc. CVPR*, 2011.

[19] H. Pirsiavash and D. Ramanan. Steerable part models. In *Proc. CVPR*, 2012.

[20] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Bilinear classifiers for visual recognition. In *Proc. NIPS*, 2009.

[21] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 2012.

[22] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *Proc. CVPR*, 2012.

[23] R. Salakhutdinov, A. Torralba, and J. B. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Proc. CVPR*, 2011.

[24] H. O. Song, S. Zickler, T. Althoff, R. Girshick, C. Geyer, M. Fritz, P. Felzenszwalb, and T. Darrell. Sparselet models for efficient multiclass object detection. In *Proc. ECCV*, 2012.

[25] K. Tang, F.-F. Li, and D. Koller. Learning latent temporal structure for complex event detection. In *Proc. CVPR*, 2012.

[26] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. PAMI*, 29(5):854–869, 2007.

[27] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *Submitted to SIAM Journal on Optimization*, 2008.

[28] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proc. CVPR*, pages 3169–3176, 2011.

[29] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. CVPR*, 2010.

[30] L. Wang, Y. Qiao, and X. Tang. Motionlets: Mid-level 3d parts for human motion recognition. In *Proc. CVPR*, pages 2674–2681, 2013.

[31] Y. Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Trans. PAMI*, 33(7):1310–1323, 2011.

[32] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and F.-F. Li. Human action recognition by learning bases of action attributes and parts. In *Proc. ICCV*, 2011.

[33] J. Zhou, J. Chen, and J. Ye. Clustered multi-task learning via alternating structure optimization. In *Proc. NIPS*, 2011.