Pattern Anal Applic
DOI 10.1007/s10044-013-0349-3

INDUSTRIAL AND COMMERCIAL APPLICATION

Robust relative attributes for human action recognition

Zhong Zhang · Chunheng Wang · Baihua Xiao · Wen Zhou · Shuang Liu

Received: 16 August 2012 / Accepted: 19 August 2013

Abstract High-level semantic feature is important to recognize human action. Recently, relative attributes, which are used to describe relative relationship, have been proposed as one of high-level semantic features and have shown promising performance. However, the training process is very sensitive to noises and moreover it is not robust to zero-shot learning. In this paper, to overcome these drawbacks, we propose a robust learning framework using relative attributes for human action recognition. We simultaneously add Sigmoid and Gaussian envelops into the loss objective. In this way, the influence of outliers will be greatly reduced in the process of optimization, thus improving the accuracy. In addition, we adopt Gaussian Mixture models for better fitting the distribution of actions in rank score space. Correspondingly, a novel transfer strategy is proposed to evaluate the parameters of Gaussian Mixture models for unseen classes. Our method is verified on three challenging datasets (KTH, UIUC and HOLLYWOOD2), and the experimental results demonstrate that our method achieves better results than previous methods in both zero-shot classification and traditional recognition task for human action recognition.

Keywords Relative attributes · Envelop loss · Zero-shot learning · Human action recognition

1 Introduction

Recognizing human actions from videos is an essential issue in computer vision and pattern recognition due to its significant applications in areas such as video surveillance and retrieval. In this paper, we address the task of human action recognition in complex scenes such as diverse and realistic settings [1]. In the previous decades, many methods were proposed to focus on this problem. Early action recognition methods emphasized on tracking motion capture and the analysis of tracks [2]. In terms of action representation, a lot of strategies have been proposed by researchers to make action representation more discriminative such as space-time pattern templates [3, 4], 2D shape matching [5, 6], optical flow patterns [7], and trajectory-based representation [8], as well as spatio-temporal interest points [9–11]. Furthermore, methods based on spatio-temporal interest points together with bag-of-words model have shown expected performance. Since these approaches do not rely on some preprocessing techniques such as background modeling or body-part tracking, they are relatively robust to viewpoint, noises and background changing. Meanwhile, they are invariant to size and illumination variation. In addition, some approaches have attempted to integrate contextual information for capturing the spatial and temporal relationship among interest points [12–15]. Savarese et al. [13] used a local histogram called *ST-correlograms* to measure feature co-occurrence patterns

Z. Zhang · C. Wang (✉) · B. Xiao · W. Zhou · S. Liu
The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, ZhongGuanCun East Rd. 95, Beijing, China
e-mail: chunheng.wang@ia.ac.cn

Z. Zhang
e-mail: zhong.zhang@ia.ac.cn

B. Xiao
e-mail: baihua.xiao@ia.ac.cn

W. Zhou
e-mail: wen.zhou@ia.ac.cn

S. Liu
e-mail: shuang.liu@ia.ac.cn

Published online: 11 September 2013

in a local 3-D region. Ryoo and Aggarwal [14] proposed a so-called "feature × feature × relationship" histogram to capture both appearance and relationship information between pairwise interest points. Kovashka and Grauman [15] designed a hierarchy of codebooks using neighborhoods of spatio-temporal interest points. Zhang et al. [12] proposed a novel coding strategy called context-constrained linear coding (CLC), which not only considered spatio-temporal contextual information but also alleviated quantization error.

The above action models based on the bag-of-words model directly associate low-level features with class labels. However, rich visual spatio-temporal information can be hardly characterized by one single class label. In order to describe action related properties, it would be better to define high-level semantic concepts. Recent works show that attributes can act as high-level semantic concept which bridge the gap between low-level features and class labels. It proves that the attributes are useful in many ways. For instances, they are helpful for recognizing familiar actions [16, 17] and also a powerful tool for the zero-shot learning problem [18, 19] (i.e., no training samples are available). Some works [16, 17, 19–21] treat attributes as middle-level features for object or video classification. Lampert et al. [17] proposed the direct attribute prediction (DAP) model which intended to predict the presence of each attribute to train object models. Liu et al. [19] treated the action attributes as latent variables, whose classifiers are pre-trained by linear SVM with outputs as the inputs of latent SVM. The above methods train attribute and object classifiers independently. Hwang et al. [22] believed that respective supervision on attributes from low-level features and objects from attributes may restrict the final performance. He claimed that the learned action and attribute classifiers have to share the same low-level features. They assume prediction attributes and objects are related, and hence, multi-task learning model [23, 24] can be used to share the low-level features. In addition, some methods have been proposed to select attributes automatically. However, there are some disadvantages for automatically mining attributes. The method in [25], which mines the relationships among nouns (objects) from text and image data, is unsuitable for discovering the semantic relationships of verbs (actions). Liu et al. [19] mine attributes from action videos using clustering algorithm. Nevertheless, these data-driven attributes have no explicit semantics.

However, the above attribute-based methods totally regard attributes as binary values which indicate the presence or absence of the corresponding attribute. In this way, the binary attribute fails to capture the degree of existence of attribute, so it could hardly represent objects veritably. So Parikh and Grauman [18] proposed the concept of relative attributes, which explore a semantically rich representation by describing relative relationships in the world. For example, 'running' has stronger presence of 'leg motion' than that of 'jogging', and 'walking' has weaker presence of 'jumping motion' than that of 'jumping from situp'. In the implementation process, a rank function is trained for each relative attribute using RankSVM with quadratic loss under the supervision of pairs of samples. The learned rank function can estimate a rank score which indicates the relative strength of the attribute presence for each sample. Then, they utilize single Gaussian model to estimate the distribution of actions in rank scores space for zero-shot learning.

Yet, the above relative attributes [18] have two disadvantages which may impact the subsequent classification and zero-shot learning. One disadvantage is that it is not robust to outliers. Since quadratic loss function increases quadratically with loss penalty, these outliers will contribute a high penalty to the global loss and deviate the optimal process. Thereby, the method [18] is sensitive to outliers. In action videos, many aspects lead to outliers, for example, mis-labeled pairwise videos, or mistakenly

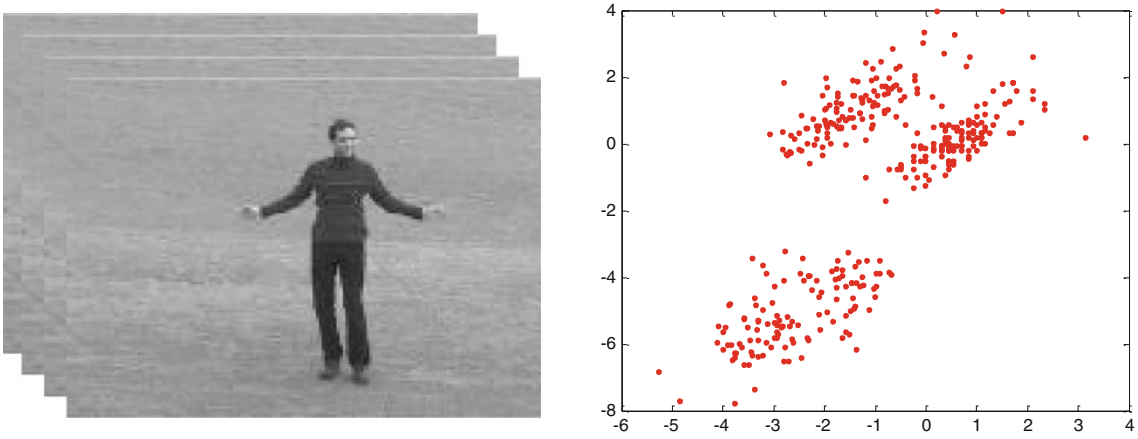


Fig. 1 The distribution of action 'handclapping' in rank score space

detected spatio-temporal interest points. In addition, if a video is a noise sample or a mis-labeled one, the number of pairwise outliers increases quadratically because the rank function is trained by pairwise training videos. This could degrade the performance of the final classifier. To overcome the above drawback, some researchers suggest that performance of pairwise learning algorithms can be improved by removing or down-weighting these outliers [26]. The other disadvantage is that it is not robust to zero-shot learning. Parikh and Grauman [18] trained generative model using signal Gaussian function in rank score space while it is unsuitable for the intrinsic distribution of actions in rank score space. Figure 1 shows the distribution of action 'handclapping' in rank score space, where the range of x and y are [6, 4] and [−8, 4], respectively. As can be deduced, signal Gaussian function does not well fit these distributions.

In this paper, we propose a robust framework to overcome the above two drawbacks. First, we present to weaken the penalty for samples that are misclassified (potential outliers), which is realized by simultaneously introducing Sigmoid and Gaussian envelopes into RankSVM loss objective. In such a way, the influence of outliers on the final trained model will be greatly reduced, thus improving the accuracy. Furthermore, we propose a novel zero-shot learning strategy. Concretely, we use Gaussian Mixture models (GMM) to train generative model, and a novel transfer strategy among classes is proposed. Experimental results on three challenging datasets show the significant improvements over the state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 briefly introduces RankSVM for relative attributes. Section 3 shows the Sigmoid and Gaussian envelop loss to overcome the negative effects of outliers. Section 4 presents robust zero-shot learning from rank functions. Section 5 further shows how our method describes the novel human actions in relative terms. In Sect. 6, we design a series of experiments to demonstrate that our experimental results are more accurate than those of the state-of-the-art methods both zero-shot classification and traditional recognition task on KTH, UIUC and HOLLYWOOD2 action database. Finally, in Sect. 7, we conclude this paper.

2 RankSVM for relative attributes

RankSVM is used to train rank functions for relative attributes, in which each rank function corresponds to one attribute [18]. Let $\mathbf{x}_i \in \mathbb{R}^D$ denote the feature vector of the ith sample in the training data set, and $A = \{a_k\}$ denote a set of K attributes. Moreover, $B_k = \{(i, j)\}$ and $S_k = \{(m, n)\}$ denote a set of ordered and un-ordered pairs of samples for attribute a_k. Specifically, $(i,j) \in B_k \Rightarrow i \succ j$ means that sample i has a stronger presence of attribute a_k than j, and $(m, n) \in S_k \Rightarrow m \sim n$ means that sample m and n have similar attribute a_k strengths. The goal is to learn K ranking functions which are subjected to the attribute strengths relationship between pairs of samples. To achieve the ends, the RankSVM for each attribute is formulated as:

$$\min \frac{1}{2}\|\mathbf{w}_k\|^2 + C\left(\sum_{i,j}\varepsilon_{ij}^p + \sum_{m,n}\delta_{mn}^p\right) \tag{1}$$

$$s.t. \quad \mathbf{w}_k^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j) \geq 1 - \varepsilon_{ij}; \quad \forall (i,j) \in B_k \tag{2}$$

$$|\mathbf{w}_k^{\mathrm{T}}(\mathbf{x}_m - \mathbf{x}_n)| \leq \delta_{mn}; \quad \forall (m,n) \in S_k \tag{3}$$

$$\varepsilon_{ij} \geq 0, \quad \delta_{mn} \geq 0. \tag{4}$$

where $\mathbf{w}_k$ is the parameter of the kth Rank SVM, ε_{ij} and δ_{mn} are non-negative slack variables, C is the balancing constant and p is the exponential of ε_{ij} and δ_{mn}. In Eq. (1), the first term maximizes the classification margin, and the second term makes sure that constrains are satisfied. C is the tradeoff parameter between them. The above formulation is equivalent to an unconstrained optimization problem:

$$\begin{aligned}&\min \lambda\|\mathbf{w}_k\|^2 + \textstyle\sum_{i,j} L\left(1 - \mathbf{w}_k^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)\right) + \\ &\textstyle\sum_{m,n} L\left(\mathbf{w}_k^{\mathrm{T}}(\mathbf{x}_m - \mathbf{x}_n)\right) + \sum_{m,n} L\left(-\mathbf{w}_k^{\mathrm{T}}(\mathbf{x}_m - \mathbf{x}_n)\right)\end{aligned} \tag{5}$$

where $\lambda = \frac{1}{2C}$ and L is the loss function, i.e., $L(t) = \max(0,t)^p$. When $p = 1$, it is hinge (linear) loss function; when $p = 2$, it is quadratic loss function. The second term penalizes ordered pairwise samples, and the loss functions can be seen in Fig. 2, as the dashed lines. The last two terms penalize similar pairwise samples, and the loss functions are shown in Fig. 3, as the dashed lines.

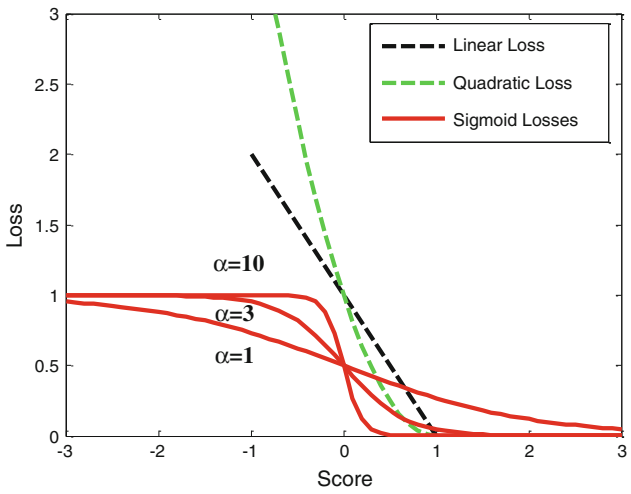


Fig. 2 Loss functions for ordered pairwise samples. The *black dashed line* indicates the linear loss function, the *green dashed line* indicates the quadratic loss function and the *red solid lines* indicate the Sigmoid envelops with $\alpha = 10, 3, 1$

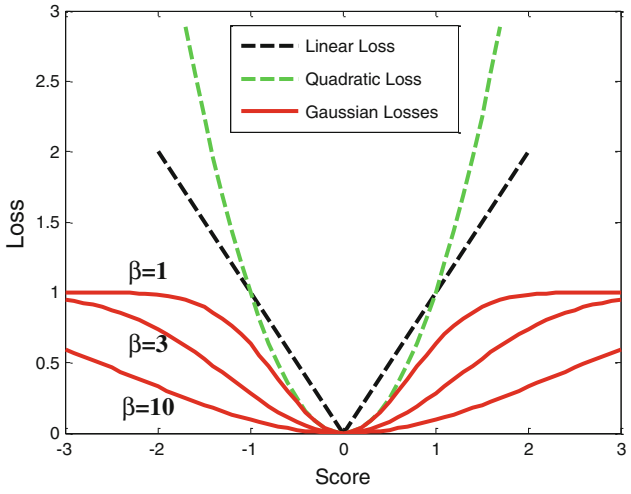


Fig. 3 Loss functions for un-ordered pairwise samples. The *black dashed line* indicates the linear loss function, the *green dashed line* indicates the quadratic loss function and the *red solid lines* indicate the Gaussian envelops with $\beta = 1, 3, 10$

3 Envelop loss for robust relative attributes

3.1 Sigmoid and Gaussian envelops

One of the disadvantages of the hinge (linear) and quadratic loss functions is that they are sensitive to outliers which frequently appear in action videos. Outlier points produce large penalties, when they are predicted by a value much smaller than zero (see the dashed lines in Fig. 2) or far away from zero (see the dashed lines in Fig. 3). Since the hinge loss increases linearly with loss penalty and quadratic loss increases quadratically, these outliers will contribute a high penalty to the global loss. Thereby, these outliers will deviate the optimal process, and the final classifier will be biased to the outliers.

To overcome the above drawback, we propose to simultaneously utilize Sigmoid and Gaussian envelops instead of the above loss functions. This could reduce the loss penalty for the outliers.

For the ordered pairs of action videos, the loss function is replaced by the Sigmoid envelop:

$$L_1(t,\alpha) = 1 - \text{sigmoid}(t,\alpha) = 1 - \frac{1}{1+e^{-\alpha t}} \tag{6}$$

where t is the penalty value, and α is a parameter which determines the steepness of the Sigmoid envelop. The Sigmoid envelops with several values of α are shown as the red solid lines in Fig. 2 where the range of x and y are $[-3, 3]$ and $[0, 3]$, respectively. We can see that when the parameter α gets smaller, the Sigmoid envelop gives smaller penalties for the same score. In contrast, when the α gets greater, the Sigmoid envelop approximates the empirical 0/1 loss.

For the similar pairwise action videos, the Gaussian envelop is utilized as loss function:

$$L_2(t,\beta) = 1 - \mathcal{N}(t|0,\beta^2) = 1 - \frac{1}{\beta\sqrt{2\pi}} e^{-t^2/2\beta^2} \tag{7}$$

where β^2 is the variance that controls the steepness of the Gaussian envelop. The Gaussian envelops with different values of β are illustrated as the red solid lines in Fig. 3. When the parameter β gets greater, the Gaussian envelop gives smaller penalties for the same score.

It is noticed that as long as loss functions give smaller loss penalties than hinge and quadratic loss functions, these loss functions can be used as envelops. However, the advantage of Sigmoid and Gaussian envelops lies in two folds. The first is that Sigmoid and Gaussian functions are derivable, which can help to solve the RankSVM. The other is that when the parameter α increases, the Sigmoid envelop approximates the empirical 0/1 loss, which is in principle a good criterion but a NP-hard problem. Thus, we utilize Sigmoid and Gaussian as envelops.

So with the Sigmoid and Gaussian envelops, the unconstrained optimization problem of RankSVM in Eq. (1) is expressed as:

$$\min \lambda\|\mathbf{w}\|^2 \ + \sum_{i,j} L_1\left(\mathbf{w}^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j), \alpha\right) + \sum_{m,n} L_2(\mathbf{w}^{\mathrm{T}}(\mathbf{x}_m - \mathbf{x}_n), \beta) \tag{8}$$

where $\mathbf{w}$ is the parameter of classifier.

3.2 Learning

To solve the objective function, we adopt gradient descent technique. The gradient of Eq. (8) with respect to $\mathbf{w}$ is:

$$\nabla = 2\mathbf{w}\lambda - \sum_{i,j} \alpha F_1(1-F_1)(\mathbf{x}_i - \mathbf{x}_j) - \sum_{i,j} \frac{1}{\beta^3} F_2(\mathbf{x}_i - \mathbf{x}_j) \tag{9}$$

where

$$\begin{cases} F_1 = 1/\{1 + e^{-\alpha \mathbf{w}^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)}\} \\ F_2 = \sqrt{\frac{2}{\pi}} e^{[\mathbf{w}^{\mathrm{T}}(\mathbf{x}_m - \mathbf{x}_n)]^2/\beta^2} \mathbf{w}^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j) \end{cases} \tag{10}$$

The gradient descent algorithm can then be written as:

$$\mathbf{w}^{\text{iter}+1} = \mathbf{w}^{iter} - \eta \nabla^{\text{iter}} \tag{11}$$

where the index iter indicates the iteration number and η is a learning rate parameter. If η is too big, the objective function may not reach the local minimum. While if η is too small, the convergence rate is too slow. In practice, η is set to 0.01 and learning stops when the relative decrease in loss is less than 0.001.

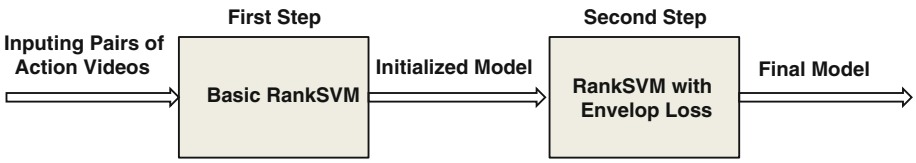


Fig. 4 The two-step learning strategy: nonconvex optimization problem is initialized with the output of a basic RankSVM

The Sigmoid and Gaussian envelops are nonconvex functions, so the solution may reach a local minimum. To avoid the locally optimal solutions, we treat the RankSVM with envelop loss as a second optimization step, refining the results from another ranker, similar to [27]. Concretely, this RankSVM is initialized with the model learned from a basic RankSVM, and then it converges to a local optimum. The flow chart of the two-step optimization problem is shown in Fig. 4.

4 Robust zero-shot learning from rank functions

After learning rank functions using the above loss envelops for each attribute, in this section we propose a robust strategy for zero-shot learning, i.e., no training samples are available [28]. Consider R action classes of interest. All the action classes should predefine attributes and their relationships, which can help deal with zero-shot learning problem. Q classes with available training data are called 'seen' classes, while the remaining $U = R - Q$ classes with no training data available are called 'unseen' classes. For the Q seen classes, they are described by relative attributes with respect to each other. For example, "'jogging' has stronger presence of 'leg motion' than 'walking', yet weaker than 'running'", "'jump forward' and 'jump from situp' have similar 'jumping motion'", etc. On the other hand, the U unseen classes should be described relative to one or two seen classes for a subset of the attributes in order to transfer the knowledge from the seen classes. Concretely, the unseen class $c_j^{(u)}$ can be described as $c_i^{(q)} \succ c_j^{(u)} \succ c_k^{(q)}$ for attribute a_k or $c_i^{(q)} \succ c_j^{(u)}$ or $c_j^{(u)} \succ c_k^{(q)}$ or $c_i^{(q)} \sim c_j^{(u)}$, where $c_i^{(q)}$ and $c_k^{(q)}$ are seen classes.

The class relationship is propagated via relative attributes during training to the corresponding action videos, e.g., for seen classes $c_i^{(q)}$ and $c_j^{(q)}, c_i^{(q)} \succ c_j^{(q)} \Rightarrow i \succ j; \quad \forall i \in c_i^{(q)}, \quad \forall j \in c_j^{(q)}$ for attribute a_k. Then we learn all K relative attributes using RankSVM with Sigmoid and Gaussian envelop loss as described in Sect. 3. Afterwards, all the action videos in the training set predicted a real-valued rank score for each rank function. This allows us to transform the ith action video $\mathbf{x}_i \in \mathbb{R}^D$ to a K-dimensional vector $\tilde{\mathbf{x}}_i \in \mathbb{R}^K$, $\tilde{\mathbf{x}}$ which indicates its real-valued rank scores for all K attributes. We build a generative model using GMM for each of the Q seen classes in rank score space $\mathbb{R}^K$, and it is formulated as:

$$P_i^{(q)}(\tilde{\mathbf{x}}) = \sum_{z=1}^{Z} \gamma_{iz}^{(q)} \varphi\left(\tilde{\mathbf{x}} \mid \boldsymbol{\mu}_{iz}^{(q)}, \boldsymbol{\Sigma}_{iz}^{(q)}\right), \quad i = 1, \ldots, Q \tag{12}$$

where Z is the number of distributions, $\gamma_{iz}^{(q)}$ is the weight of the zth Gaussian which is learned from Q seen classes, $\boldsymbol{\mu}_{iz}^{(q)} \in \mathbb{R}^K$ is the mean value, $\boldsymbol{\Sigma}_{iz}^{(q)} \in \mathbb{R}^{K \times K}$ is the covariance matrix. In addition, φ is a Gaussian probability density function where the covariance matrix is assumed to be diagonal matrix for computational efficiency:

$$\varphi\left(\tilde{\mathbf{x}} \mid \boldsymbol{\mu}_{iz}^{(q)}, \boldsymbol{\Sigma}_{iz}^{(q)}\right) = \frac{1}{(2\pi)^{K/2} |\boldsymbol{\Sigma}_{iz}^{(q)}|^{1/2}} e^{-\frac{1}{2}(\tilde{\mathbf{x}} - \boldsymbol{\mu}_{iz}^{(q)})^{\mathrm{T}} \boldsymbol{\Sigma}_{iz}^{(q)^{-1}} (\tilde{\mathbf{x}} - \boldsymbol{\mu}_{iz}^{(q)})} \tag{13}$$

Now, we transfer knowledge from seen classes to unseen classes. The parameters of the GMM for U unseen classes are estimated using the relative descriptions. More specifically, we first sort the Z Gaussian distributions with the corresponding mean values in descending order for each GMM of seen class. Then, an unseen class $c_j^{(u)}$ is given and we follow the rules below:

1. In the case of $c_i^{(q)} \succ c_j^{(u)} \succ c_k^{(q)}$ for attribute a_d, where a_d is one of pre-defined attributes for unseen class $c_j^{(u)}$, and $c_i^{(q)}$ and $c_k^{(q)}$ are seen classes, the dth dimension of mean value of the zth Gaussian component $\boldsymbol{\mu}_{jz\_d}^{(u)}$ is set to $\frac{1}{2}(\gamma_{iz}^{(q)} \boldsymbol{\mu}_{iz\_d}^{(q)} + \gamma_{kz}^{(q)} \boldsymbol{\mu}_{jz\_d}^{(q)})$;
2. In the case of $c_i^{(q)} \succ c_j^{(u)}$, $\boldsymbol{\mu}_{jz\_d}^{(u)}$ is set to $\boldsymbol{\mu}_{iz\_d}^{(q)} - g_z$, where g_z is the weighted average value of rank scores of seen classes for attribute a_d, i.e., $g_z = \frac{1}{Q} \sum_{b=1}^{Q} \gamma_{bz}^{(q)} \boldsymbol{\mu}_{bz\_d}^{(q)}$;
3. Similarly, in the case of $c_j^{(u)} \succ c_k^{(q)}$, $\boldsymbol{\mu}_{jz\_d}^{(u)}$ is set to $\boldsymbol{\mu}_{kz\_d}^{(q)} + g_z$;
4. In the case of $c_j^{(u)} \sim c_i^{(q)}$, $\boldsymbol{\mu}_{jz\_d}^{(u)}$ is equal to $\boldsymbol{\mu}_{iz\_d}^{(q)}$, and d, the entry of the covariance matrix of the zth Gaussian component $\boldsymbol{\Sigma}_{jz\_d}^{(u)}$ is equal to $\boldsymbol{\Sigma}_{iz\_d}^{(q)}$;
5. If a_d is not used to describe $c_j^{(u)}$, $\boldsymbol{\mu}_{jz\_d}^{(u)}$ is simply set to g_z.

In all the above cases except case 4, we simply set $\boldsymbol{\Sigma}_{jz\_d}^{(u)} = \frac{1}{Q} \sum_{b=1}^{Q} \gamma_{bz}^{(q)} \boldsymbol{\Sigma}_{bz\_d}^{(q)}$.

For a test action video, we first compute $\tilde{\mathbf{x}}_i \in \mathbb{R}^K$ using the proposed RankSVM that indicates the rank scores of relative attribute for the action video. Then the video is assigned to the seen or unseen class which has the highest probability:

$$c^* = \arg \max_{k \in \{1,\ldots,R\}} P_k(\tilde{\mathbf{x}}_i) \tag{14}$$

From the above process of setting the parameters of generative model for the unseen classes, we can see that it is essential to transfer knowledge from seen classes according to prior, i.e., pre-defined relative attributes. The algorithm of our method is shown in Algorithm 1.

Algorithm 1: Robust relative attributes

Input: Training data $\{x_i\}$, attributes $A = \{a_k\}$ and their relationships $B_k = \{(i,j)\}, S_k = \{(m,n)\}$

Output: Action class label

1. train RankSVM with the Sigmoid and Gaussian loss according to Eq. (8);
2. get the rank score for each training data $\{x_i\}$;
3. build a generative model for each Q seen classes according to Eq. (12);
4. transfer the generative model from seen classes to unseen classes according to the rules mentioned in Sec. 4;
5. classify the test sample according to Eq. (14).

5 Describing human actions in relative terms

We can make use of relative attributes to describe a novel action video: whether its class happens to be familiar or not. Our goal is to describe the specific attribute strength of a novel action video on the basis of two reference action videos.

In the training stage, we are given a set of training videos, each represented by a feature vector $\mathbf{x}_i \in \mathbb{R}^D$, a list $A = \{a_k\}$ of K attributes, and the relationship in relative strength of a_k, i.e., $B_k = \{(i,j)\} s.t. i \succ j$ and $S_k = \{(m, n)\}$ $s.t.\ m \sim n$. Note that a_k is the pre-defined attribute and all the training data are from seen classes. We learn K rank functions by our proposed RankSVM as described in Sect. 3 and evaluate them on all training action videos.

To describe a novel action video j, we calculate rank scores using all K rank functions. For each attribute a_k, we choose two reference action videos i and l from training set to describe j via relative attributes. In theory, any training video with a good rank function could be a reference video. However, in practice we are in a dilemma. On one hand, we wish to select reference videos not similar to the novel video according to attribute strength to avoid an overly precise description. On the other hand, to avoid unrelated descriptions, the value of attribute strength should not be too far from that of the novel action video. Therefore, we pick the reference action videos i and l as in [18]. Concretely, when $i \succ j \succ l$, we prefer to leave 1/8 action videos between i and j, as well as j and l. When i or l does not exist in extreme case, l is chosen to be the action video with the least strength of a_k, and i corresponds to the action video with the highest strength of a_k. After getting the reference videos, we can describe the novel action video depending on the rank scores, for instance "j has a stronger presence of attribute a_k than j, yet weaker than l".

6 Experiments and discussion

In this section, we first present the low-level features for action and attribute representation, as well as the compared baseline algorithms. Second, we introduce the experimental datasets and define the action relative attributes. Third, we verify the effect of our method for zero-shot learning on three datasets. Fourth, we present the results of traditional classification task. Fifth, we show our method could describe the actions in videos using relative attributes. Finally, we discuss the influence of parameters in our algorithm.

6.1 Low-level feature extraction and baseline

6.1.1 Low-level feature extraction

To detect interest points from videos of action, we adopt the Harris 3D corner detector proposed in [29], which is an extension of the Harris 2D corner. The Harris 3D method detects the location where the video intensities have significant local variations in both space and time. In order to achieve this goal, matrix F is defined as:

$$F = \begin{bmatrix} L_x^2 & L_xL_y & L_xL_t \\ L_xL_y & L_y^2 & L_yL_t \\ L_xL_y & L_yL_t & L_t^2 \end{bmatrix} \tag{15}$$

where L_x, L_y and L_t are the gradients of Gaussian smoothed video in horizontal, vertical and temporal directions. The Harris 3D corner detector finds points whose F has large eigenvalues. For each interest point, the histogram of oriented gradients (HOG) [30] and histogram of optical flow

(HOF) [31] are used as local appearance descriptors. The two features can reflect the action characteristic. All the descriptors are quantized to D visual words using the k-means clustering. Then, each action video is represented by a histogram vector $\mathbf{x}_i \in \mathbb{R}^D$ in the framework of standard bag-of-words model [32].

6.1.2 Baseline

We compare our zero-shot learning algorithm with relevant baselines and other excellent algorithms on human action recognition. There are two algorithms as our relevant baselines. The first one is the Direct Attribute Prediction (DAP) model proposed by Lampert et al. [17], which uses binary attribute descriptions for all classes. Action videos from Q seen classes are trained by linear SVM for each binary attribute. A test action video is assigned to a class using:

$$c^* = \arg \max_{c \in \{1,\ldots,R\}} \prod_{k=1}^{K} P\left(a_k = h_k^c | \mathbf{x}\right) \tag{16}$$

where $P(a_k = h_k^c|\mathbf{x})$ is computed by transforming the binary classifier score via a Sigmoid function, and h_k^c is the ground-truth binary bit for attribute a_k of class c. If a_k is not used to describe an unseen class, we set $P(a_k = h_k^c|\mathbf{x})$ to 0.5. We implement this baseline using LIBSVM [33].

The other baseline is proposed by Parikh and Grauman [18]. They use RankSVM with quadratic loss (**QRS**) to train rank functions for per attribute. After getting the predicted rank scores from the rank function, they then build a generative model via a single Gaussian function (**SG**) for the sequential zero-shot learning.

6.2 Datasets and action relative attributes

We validate our algorithm on three publicly available datasets: KTH dataset [34], UIUC action dataset [35] and HOLLYWOOD2 [36]. The KTH dataset is a standard benchmark for human action recognition. It contains six action classes (box, handclap, hand wave, jog, run, and walk), each of which is performed in four different scenarios by 25 subjects, resulting in a total of 599 video clips. We quantize the descriptors by k-means clustering and the number of codebook is set to 2,000. We manually define 5 relative attributes such as "leg motion" and "arm motion", as shown in Table 1. The UIUC action dataset [35] contains about 532 videos of 13 actions ('sit to stand' and 'stand to sit' are combined to 'sit and stand') including jump forward, push up, raise one hand, etc. These videos are performed by 8 actors. We quantize the descriptors by k-means clustering and the number of codebook is set to 2,000. We manually define 10 relative attributes as illustrated in Table 2. The Hollywood 2 dataset is composed of video clips extracted from 69 Hollywood movies, and contains 12 classes of human actions (AnswerPhone, DriveCar, Eat, FightPerson, GetOutCar, HandShake, HugPerson, Kiss, Run, SitDown, SitUp and StandUp). There are totally 1,707 action videos divided into a training set (823 videos) and a test set (884 videos), where training and test samples are obtained from different movies. The dataset intends to provide a comprehensive benchmark for human action recognition in realistic and challenging settings. We quantize the descriptors by k-means clustering and the number of codebook is set to 3,000. We manually define 7 relative attributes as shown in Table 3.

Table 1 The definition of action relative attributes for the KTH dataset

Attribute	Relative
Leg motion	Run≻jog≻walk≻box ∼ handclap ∼ handwave
Arm motion	Handwave≻handclap≻box≻run≻jog ∼ walk
Arm–hand open	Handclap≻handwave≻box ∼ jog ∼ run ∼ walk
Arm-shape straight	Handclap ∼ handwave≻walk≻box≻jog ∼ run
Over chest-level arm motion	Handwave≻box ∼ handclap≻jog ∼ run≻walk

Table 2 The definition of action relative attributes for the UIUC action dataset

Attribute	Relative
Jumping motion	3 ∼ 4≻5≻8≻12≻1 ∼ 2 ∼ 6 ∼ 7 ∼ 9 ∼ 10 ∼ 11 ∼ 13
Arm motion over shoulder	5 ∼ 10≻13 ∼ 7≻1≻8≻12 ∼ 11 ∼ 3 ∼ 4 ∼ 6 ∼ 9 ∼ 2
Arm: intense motion	5≻3≻10≻13≻7≻4 ∼ 8≻1≻12≻11≻2 ∼ 6≻9
Leg: intense motion	3 ∼ 5≻8≻12≻4≻9≻2≻11≻1 ∼ 6 ∼ 7 ∼ 10 ∼ 13
Cyclic motion	1 ∼ 5 ∼ 6≻2 ∼ 3 ∼ 8 ∼ 12≻13≻4 ∼ 7 ∼ 9 ∼ 10 ∼ 11
Arm straight	4 ∼ 5 ∼ 10 ∼ 11≻2 ∼ 3 ∼ 9 ∼ 12≻6≻7 ∼ 13≻1≻8
Raise arms	5 ∼ 10≻7 ∼ 13≻1 ∼ 3 ∼ 9≻8 ∼ 12≻2 ∼ 4 ∼ 6 ∼ 11
Leg: fold motion	4 ∼ 9≻2≻3 ∼ 8≻12 ∼ 5≻11≻1 ∼ 6 ∼ 7 ∼ 10 ∼ 13
Arm–hand: move-back–forward	3 ∼ 8≻12 ∼ 2≻1 ∼ 4 ∼ 5 ∼ 6 ∼ 7 ∼ 9 ∼ 10 ∼ 11 ∼ 13
Torso vertical-shape up/down motion	4≻9≻6≻3 ∼ 5≻8 ∼ 12≻11≻1 ∼ 2 ∼ 7 ∼ 10 ∼ 13

1 Hand-clap, *2* crawl, *3* jump forward, *4* jump from situp, *5* jump jacks, *6* push up, *7* raise one hand, *8* run, *9* sit and stand, *10* streth out, *11* turn, *12* walk, *13* wave

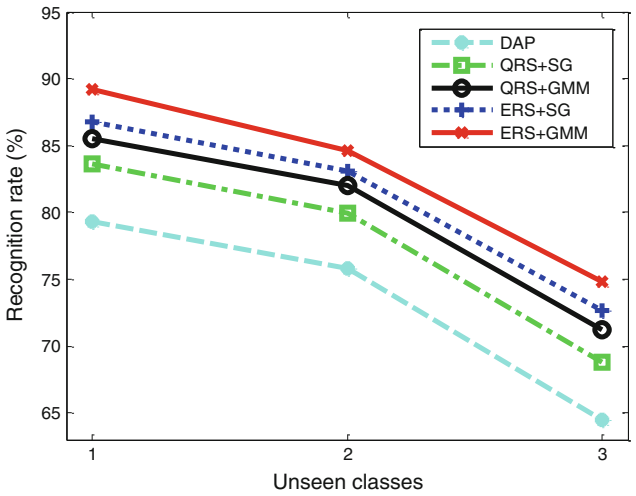
6.3 Zero-shot learning results

In this subsection, we verify the effect of zero-shot learning on the three datasets. As indicated above, relative attributes can help to recognize unseen action classes. We use the

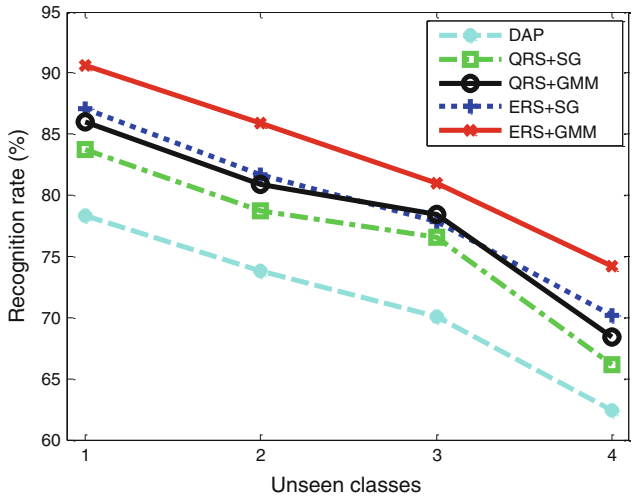
Table 3 The definition of action relative attributes for the HOLLYWOOD2 action dataset

Attribute	Relative
Arm motion	4≻9≻6≻7∼3∼2∼1≻5∼8∼10∼11∼12
Arm Cyclic motion	9≻6≻2≻1∼3∼4∼5∼7∼8∼10∼11∼12
Raise arms	4≻1∼3∼7≻9≻11∼12∼6≻2∼5∼8∼10
Torso up	12≻5∼11≻1∼2∼3∼4∼6∼7∼8∼9≻10
Leg: intense motion	9≻12∼5≻1∼2∼3∼4∼6∼7∼8∼10∼11
Arm–hand open	4≻7∼8≻9≻1∼2∼3∼5∼6∼10∼11∼12
Arm–hand: move-back-forward	9≻10∼11∼12≻5≻1∼2∼3∼4∼6≻7∼8

1 answerphone, *2* drivecar, *3* eat, *4* fightperson, *5* getoutcar, *6* handshake, *7* hugperson, *8* kiss, *9* run, *10* sitdown, *11* SitUp, *12* standup



(a) KTH dataset



(b) UIUC dataset

Fig. 5 Zero-shot learning performance as the proportion of unseen classes increase for **a** KTH dataset and **b** UIUC dataset

leave-$\mathbb{Z}$-classes-out-cross-validation strategy [18, 19]. Concretely, for each run we leave $\mathbb{Z}$ classes out as unseen classes, and the remaining classes are used for training.

We first examine the zero-shot learning accuracy as the proportion of unseen classes increases. The results are illustrated in Figs. 5 and 6, in which the unseen classes are from 1 to 3 for KTH and HOLLYWOOD2 dataset, and from 1 to 4 for UIUC dataset. In this figure, DAP is the method proposed in [17]; QRS + SG [18] uses RankSVM with quadratic loss combining with a single Gaussian function; QRS + GMM uses RankSVM with quadratic loss combining with Gaussian mixture model; ERS + SG uses RankSVM with Sigmoid and Gaussian envelop loss combining with a single Gaussian function; ERS + GMM uses RankSVM with Sigmoid and Gaussian envelop loss combining with Gaussian mixture model. Some interesting observations can be drawn from the figure. The first one is recognition rate for all five algorithms decreases with more unseen classes. However, our ERS + GMM performs better than the other algorithms in most situations. The second one is ERS + GMM about 2% better than ERS + SG, and QRS + GMM is also better than QRS + SG, due to training generative model using GMM which can accurately capture the intrinsic distribution of actions in rank score space. The results show that GMM is beneficial for the subsequent recognizing unseen classes. The third one is ERS + GMM which gains about 3 % recognition rate over QRS + GMM and ERS + SG also gains about 3 % recognition rate over QRS + SG, owing to adopting Sigmoid and Gaussian loss to restrain the influence of outliers. In a word, the accuracy of our method (ERS + GMM) is about 5 % better than that of QRS + SG [18]. The last one is ERS + GMM, QRS + GMM, ERS + SG, and QRS + SG, all of which achieve better results than DAP due to the benefit of relative description.

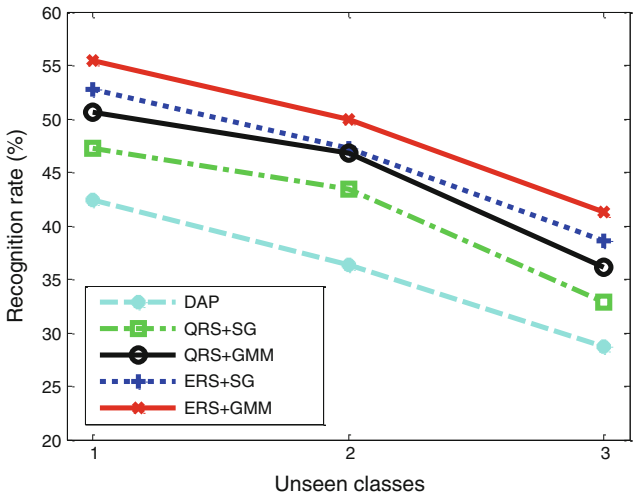


Fig. 6 Zero-shot learning performance as the proportion of unseen classes increase for HOLLYWOOD2 dataset

Further, we show the average accuracy of each class in the case of $\mathbb{Z} = 2$. In this situation, all 15 possible configurations of training and test classes for KTH dataset and 78 for UIUC dataset are used. See Figs. 7 and 8. We can see that our method (ERS + GMM) gets better results in most classes. For the KTH dataset, majority of classes are recognized with an accuracy of 80 % and obtain better performance than the compared methods. Yet, our approach achieves worse result than other approaches only on the 'run' action. It is because the appearances and attribute relationships are close to 'jog' action. In addition, our method successfully achieves 75 % accuracy rate for most of the classes in UIUC dataset, and 8 of the classes over 90 %.

We then examine the performance of our method under a noisy environment. For the two datasets, feature vector $\mathbf{x}$ of each sample is corrupted by the zero mean additive Gaussian noises, where its standard deviation is determined by the corresponding signal to noise ratios (SNR) value. Figures 9 and 10 show the results on KTH and UIUC datasets, respectively. In the two figures, the left column is ERS + GMM vs. QRS + GMM, while the right column is ERS + SG vs. QRS + SG. As we can see, all approaches maintain their performances in two datasets when SNR $\geq$ 40. The reason is that in such cases, the noise intensity is negligible compared to the action video signal intensity. The action videos are not significantly corrupted by noises. However, when the SNR value drops below 40,

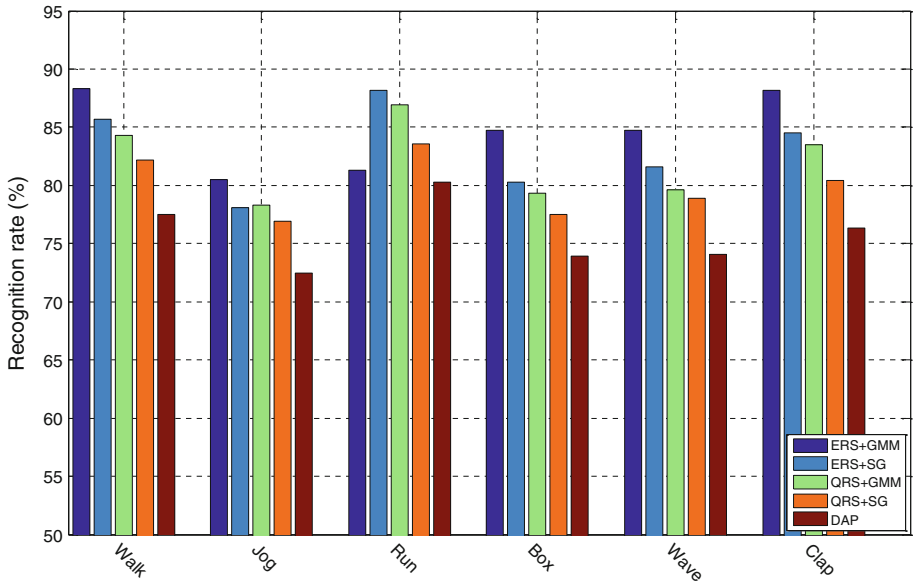


Fig. 7 The average accuracy of leave-two-classes-out-cross-validation on KTH dataset for recognizing unseen classes

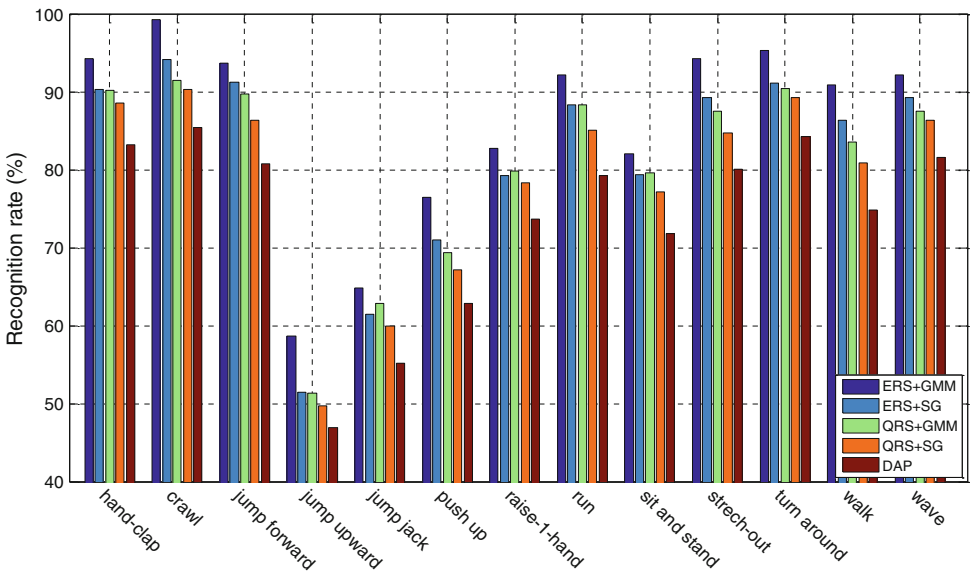


Fig. 8 The average accuracy of leave-two-classes-out-cross-validation on UIUC action dataset for recognizing unseen classes

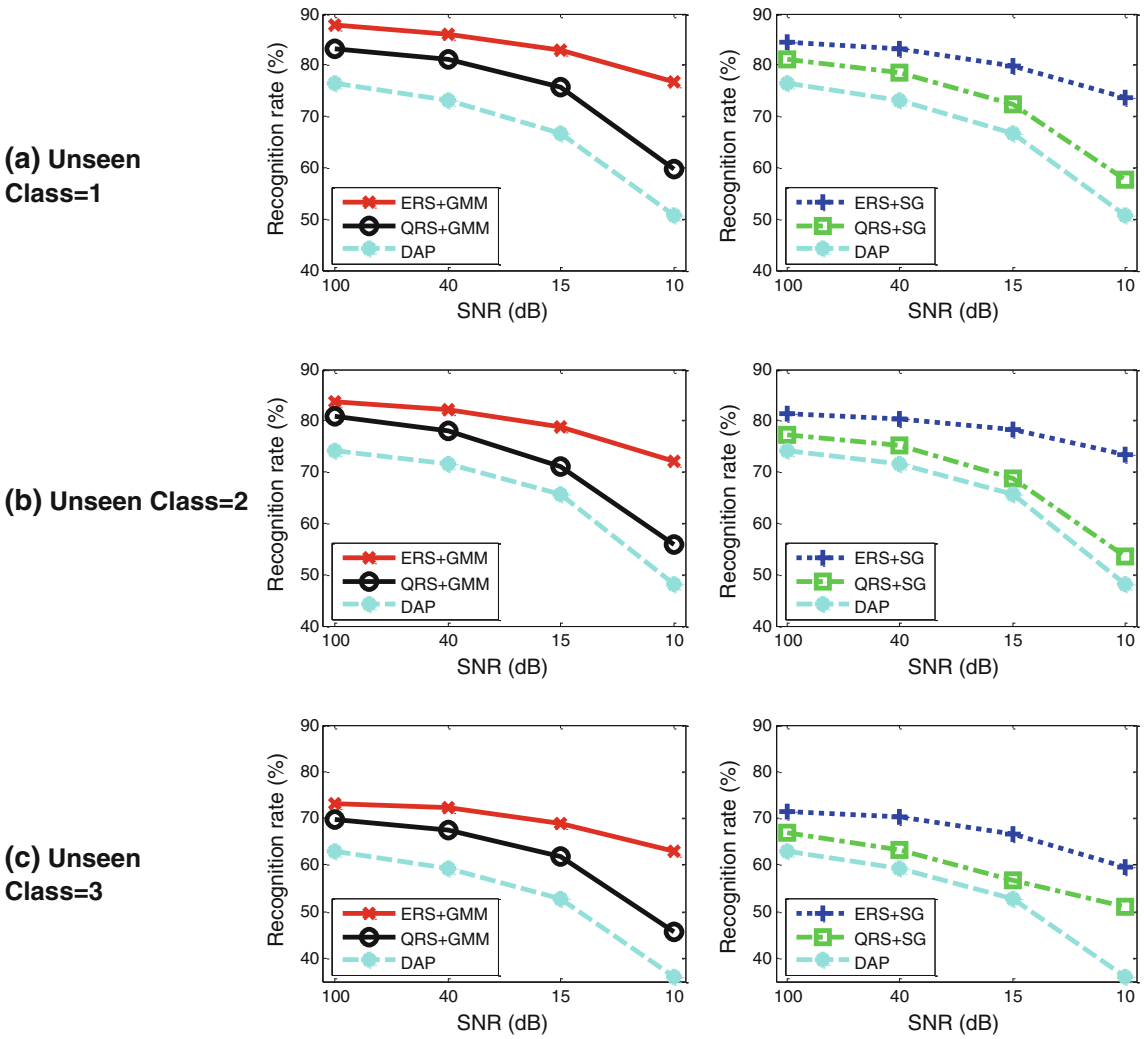


Fig. 9 The performance of different methods under the additive Gaussian noise environment of different SNR on KTH dataset for recognizing unseen classes

the noise power rises so that different methods have considerable declines of accurate rate.

In KTH dataset, by adopting RankSVM with Sigmoid and Gaussian envelop loss, i.e., ERS + GMM and ERS + SG, they are more robust against noises than QRS + GMM and QRS + SG. Even though in the toughest situation (SNR = 10), the ERS effectively restrains noises to deliver promising accurate rate. In the UIUC dataset, the classification abilities of various methods reduce significantly when SNR falls below 15. In contrast, the proposed ERS outperforms other approaches with a tolerable accuracy due to adopting the Sigmoid and Gaussian envelop loss.

6.4 Traditional classification results

In this subsection, we take a series of experiments on KTH, UIUC and HOLLYWOOD2 action dataset to

Table 4 Recognition results of different methods on the KTH dataset

Method	Accuracy (%)
Laptev et al.[37]	91.8
Savarese et al. [13]	86.83
Ryoo and Aggarwal [14]	93.8
Kovashka and Grauman [15]	94.53
Wang et al. [38]	93.8
Liu et al. [19]	91.59
Ji et al. [39]	90.2
DAP [17]	82.2
QRS + SG [18]	88.1
QRS + GMM	92.7
ERS + SG	93.8
ERS + GMM (Ours)	**95.98**

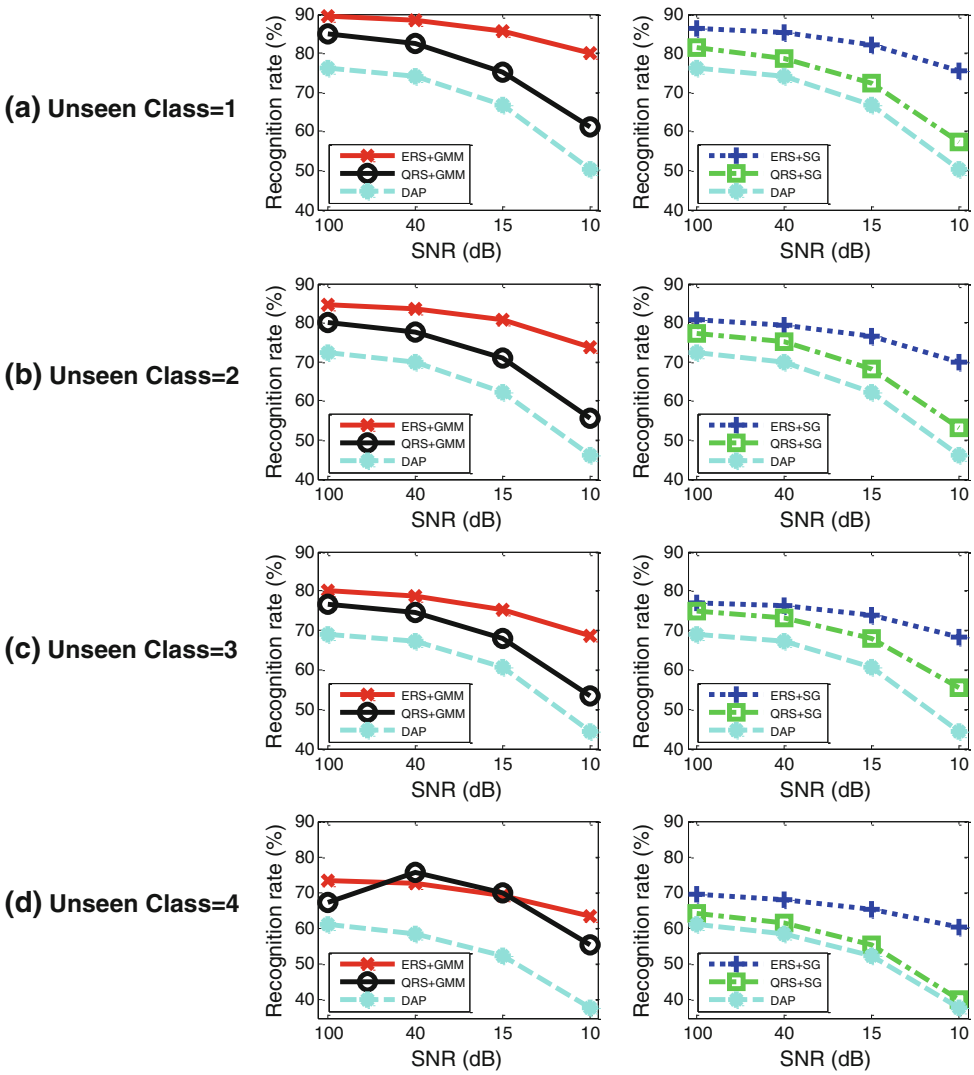


Fig. 10 The performance of different methods under the additive Gaussian noise environment of different SNR on UIUC action dataset for recognizing unseen classes

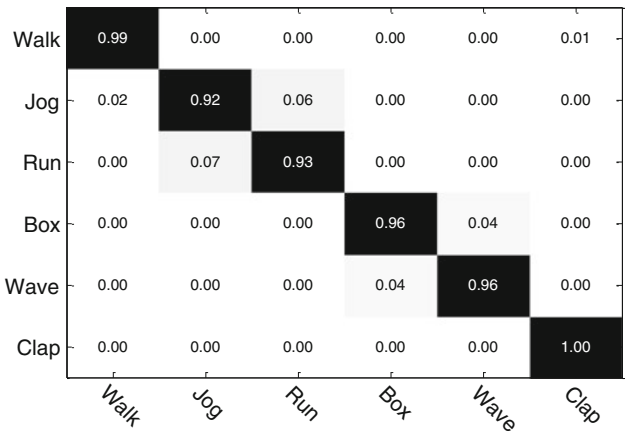


Fig. 11 Confusion table of our method on the KTH database

prove that our proposed algorithm can also improve performance of traditional action classification. Note that when zero unseen class $Z = 0$, the zero-shot learning method degenerates to the traditional classification task.

Table 5 Recognition results of different methods on the UIUC action dataset

Method	Accuracy (%)
Tran and Sorokin [35]	98.31
DAP [17]	86.2
QRS + SG [18]	93.4
QRS + GMM	95.7
ERS + SG	96.2
ERS + GMM (Ours)	**98.87**

In KTH dataset, we adopt the Leave One Out Cross Validation (LOOCV) strategy [14, 15], specifically 24 videos of actors as training and the rest one as test videos. Table 4 compares our algorithm with the other excellent algorithms. We can see that our method achieves the

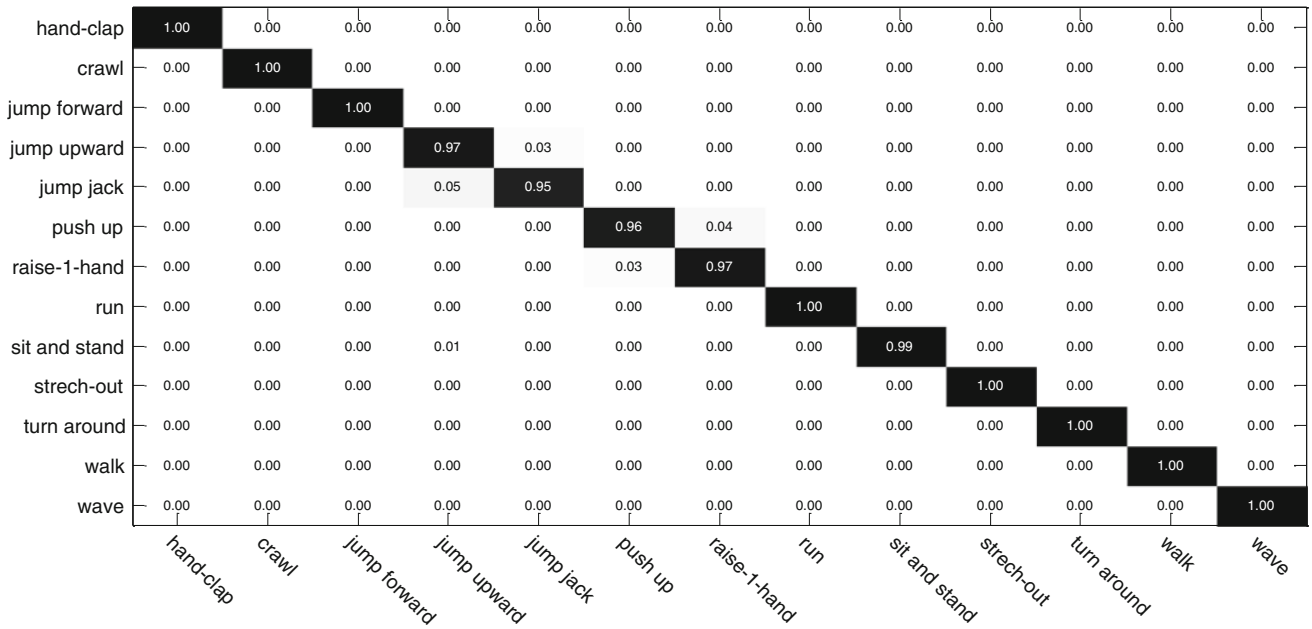


Fig. 12 Confusion table of our method on the UIUC action database

Table 6 Recognition results of different methods on the HOLLYWOOD2 action dataset

Method	Accuracy (%)
Marzalek et al. [36]	35.50
Han et al. [40]	42.12
Gilbert et al. [41]	50.90
Ullah et al. [42]	55.70
Chakraborty et al. [43]	58.46
DAP [17]	46.23
QRS + SG [18]	53.10
QRS + GMM	56.87
ERS + SG	58.65
ERS + GMM (Ours)	**61.73**

highest recognition accuracy of 95.98 %. Noticeably, the accuracy rate of ERS + GMM is over 2 % than that of ERS + SM baseline, due to accurately describing the intrinsic distribution of actions using GMM in rank score space. Also, ERS + GMM gains more than 3 % accuracy rate over QRS + GMM, owing to simultaneously adding Sigmoid and Gaussian envelop loss to restrain the impact of noises. In addition, relative descriptions (ERS + GMM, ERS + SM, QRS + GMM, QRS + SM) achieve significant improvement than binary attributes (DAP). Note that our method uses less training data but achieve better performance than [19] which treats attributes as latent variables. Concretely, we only use KTH as training data, yet [19] uses a mixed dataset including KTH [34], Weizmann [4], and UIUC [35]. Figures 10, 11 shows the confusion table of recognition results on the KTH dataset, from which we can see that leg-related actions (jog and run) are prone to be misclassified, due to their similar strength of attributes and appearance exhibitions.

We then verify our method on the UIUC action dataset, and Table 5 shows the results. Our algorithm achieves the highest recognition accuracy of 98.87 %. Figure 12 shows the confusion table of recognition results on this dataset, from which we can see that 8 classes achieves recognition accuracy of 100 %.

Finally, we test our algorithm on the HOLLYWOOD2 action dataset, and Table 6 shows the results. Our algorithm obtains the best results. Once again, we prove the effectiveness of our algorithm on this dataset which is realistic and challenging.

6.5 Describing action videos using relative attributes

As mentioned in Sect. 5, our algorithm can describe novel action video using relative attributes. For instance, 'jump jack' has a stronger presence of attribute than 'crawl', but weaker than 'jump from situp'. Figure 13 shows some examples of describing novel action videos.

6.6 Discussion

In this subsection, we further evaluate the performance of the proposed method with respect to α for Sigmoid envelop, β for Gaussian envelop, and the number of Gaussian components Z for zero-shot learning. The paper mainly reports the results on KTH dataset, and our

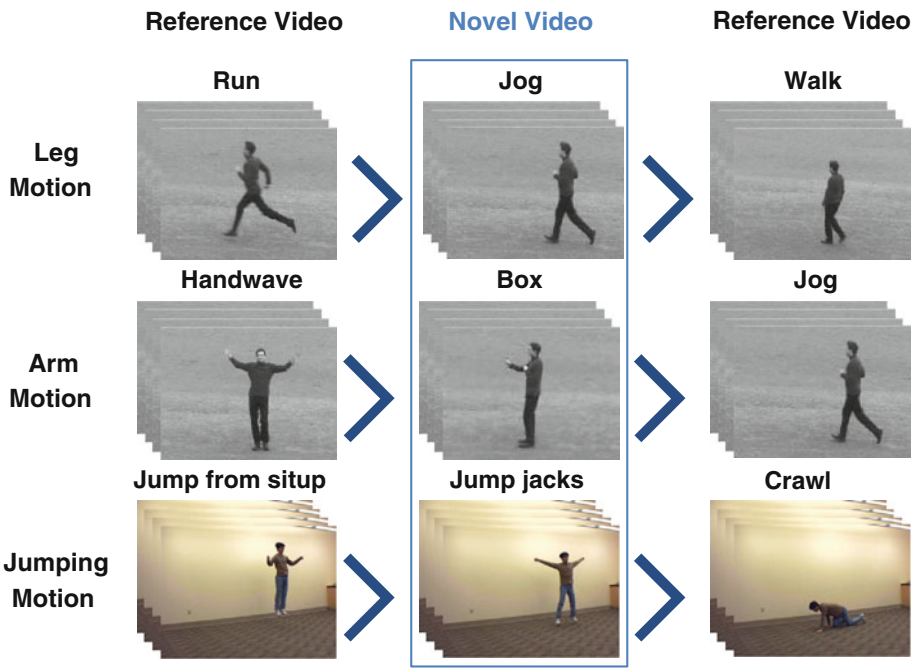


Fig. 13 Examples of describing novel action videos using relative attributes. $A > B$ denotes A has a stronger presence of attribute than B

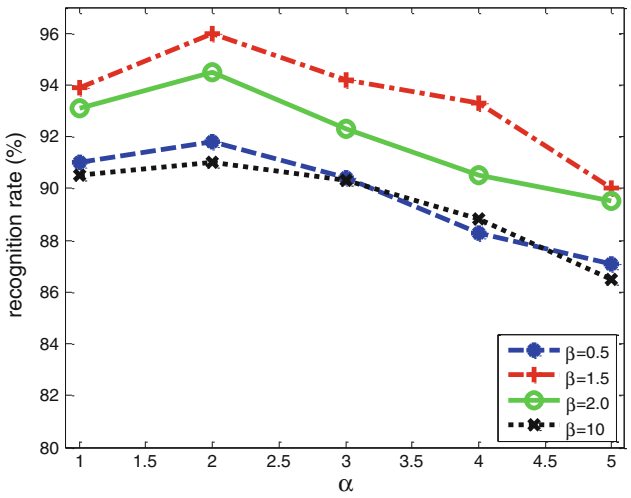


Fig. 14 Performance of our algorithm under different α and β on KTH dataset

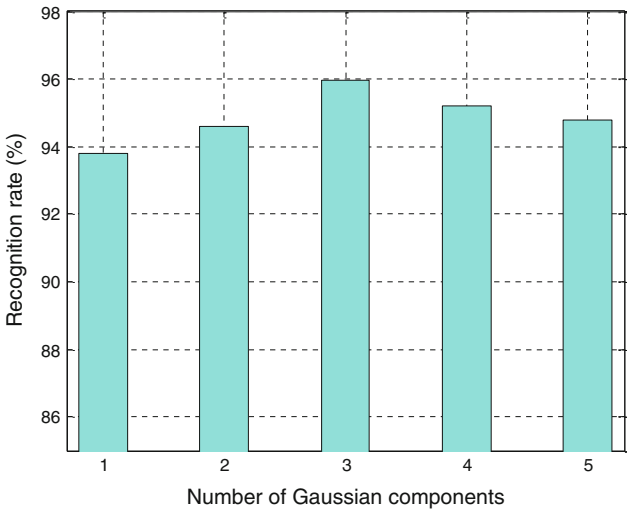


Fig. 15 Performance of our algorithm under different Z on KTH dataset

experiments have shown that the conclusions can be generalized to UIUC and HOLLYWOOD2 action dataset as well.

We first study the influence of parameters α in Eq. (6) and β in Eq. (7) in our algorithm. From Fig. 14, the experimental results indicate that when $\alpha = 2$ and $\beta = 1.5$, results are the best. We then test the performance under different number of Gaussian components Z. Figure 15 lists the performance using 1, 2, 3, 4, and 5 Gaussian components, respectively. As can be seen, the performance increases with the raise of Gaussian components, while it starts to drop when Z goes bigger than 3. Thus, we use $Z = 3$ in our experiments. Note that the previous set of experiments has been conducted with the optimal set of parameters.

7 Conclusion

In this paper, a robust learning framework using relative attributes is proposed for human action recognition. We first restrain the influence of outliers implemented by simultaneously adding Sigmoid and Gaussian envelops into the traditional RankSVM loss objective. In this way, the

effect of outliers on the optimization has been greatly reduced, thus improving the accuracy. Furthermore, we utilize GMM to estimate the distribution of actions in rank score space, and then a novel transfer strategy is proposed for evaluating the parameters of GMM for unseen classes. The experimental results show better results than previous methods in human action recognition.

Acknowledgments This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 60933010, No. 61172103 and No. 61271429 and National High-tech R&D Program of China (863 Program) under Grant No. 2012AA041312.

References

1. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) HMDB: a large video database for human motion recognition. In: IEEE conference on computer vision (ICCV), pp 2556–2563
2. Aggarwal JK, Cai Q (1997) Human motion analysis: a review. In: IEEE nonrigid and articulated motion workshop, pp 90–102
3. Yilmaz A, Shah M (2005) Actions sketch: a novel action representation. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 984–989
4. Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as space-time shapes. In: IEEE conference on computer vision (ICCV), pp 1395–1402
5. Lin Z, Jiang Z, Davis LS (2009) Recognizing actions by shape-motion prototype trees. In: IEEE conference on computer vision (ICCV), pp 444–451
6. Lv F, Nevatia R (2007) Single view human action recognition using key pose matching and viterbi path searching. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8
7. Efros A, Berg A, Mori G, Malik J (2003) Recognizing action at a distance. In: IEEE conference on computer vision (ICCV), pp 726–733
8. Raptis M, Soatto S (2010) Tracklet descriptors for action modeling and video analysis. In: European Conference on Computer Vision (ECCV) pp 577–590
9. Dollár P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. In: IEEE conference on computer vision workshop on visual surveillance and performance evaluation of tracking and surveillance (VS-PETS), pp 65–72
10. Liu J, Shah M (2008) Learning human actions via information maximization. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8
11. Liu J, Yang Y, Shah M (2009) Learning semantic visual vocabularies using diffusion distance. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 461–468
12. Zhang Z, Wang C, Xiao B, Zhou W, Liu S (2012) Action recognition using context-constrained linear coding. IEEE Signal Process Lett 19(7):439–442
13. Savarese S, DelPozo A, Niebles JC, Fei-Fei L (2008) Spatial-Temporal correlatons for unsupervised action classification. In: IEEE workshop on Motion and Video Computing (WMVC), pp 1–8
14. Ryoo MS, Aggarwal JK (2009) Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: IEEE conference on computer vision (ICCV), pp 1593–1600
15. Kovashka A, Grauman K (2010) Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2046–2053
16. Farhadi A, Endres I, Hoiem D, Forsyth D (2009) Describing objects by their attributes. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1778–1785
17. Lampert CH, Nickisch H, Harmeling S (2009) Learning to detect unseen object classes by between-class attribute transfer. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 951–958
18. Parikh D, Grauman K (2011) Relative attributes. In: IEEE conference on computer vision (ICCV), pp 503–510
19. Liu J, Kuipers B, Savarese S (2011) Recognizing human actions by attributes. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 3337–3344
20. Kumar N, Berg AC, Belhumeur PN, Nayar SK (2009) Attribute and simile classifiers for face verification, In: IEEE conference on computer vision (ICCV), pp 365–372
21. Wang Y, Mori G (2010) A discriminative latent model of object classes and attributes. In: European Conference on Computer Vision (ECCV), pp 155–168
22. Hwang SJ, Sha F, Grauman K (2011) Sharing features between objects and their attributes. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1761–1768
23. Argyriou A, Evgeniou T, Pontil M (2008) Convex multi-task feature learning. Mach Learn 73(3):243–272
24. Liu J, Ji S, Ye J (2009) Multi-task feature learning via efficient $l_{2,1}$-norm minimization. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, pp 339–348
25. Berg T, Berg A, Shih J (2010) Automatic attribute discovery and characterization from noisy web data. In: European Conference on Computer Vision (ECCV), pp 663–676
26. Elsas JL, Carvalho VR, Carbonell JG (2008) Fast learning of document ranking functions with the committee perceptron. In: ACM conference on web search and data mining (WSDM), pp 55–64
27. Perez-Cruz F, Navia-Vazquez A, Figueiras-Vidal AR, Artes-Rodriguez A (2008) Empirical risk minimization for support vector classifiers. IEEE Trans Neural Netw 14(2):296–303
28. Larochelle H, Erhan D, Bengio Y (2008) Zero-data learning of new tasks. In: AAAI Conference on Artificial Intelligence (AAAI), pp 646–651
29. Laptev I (2005) On space-time interest points. Int J Comput Vis (IJCV) 64(2):107–123
30. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 886–893
31. Dalal N, Triggs B, Schmid C (2006) Human detection using oriented histograms of flow and appearance. In: European Conference on Computer Vision (ECCV), pp 428–441
32. Wang H, Ullah MM, Klaser A, Laptev I, Schmid C (2009) Evaluation of local spatio-temporal features for action recognition. In: British Machine Vision Conference (BMVC)
33. Chang CC, Lin CJ (2001) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27
34. Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: A local SVM approach. In: International Conference on Pattern Recognition (ICPR), pp 32–36
35. Tran D, Sorokin A (2008) Human activity recognition with metric learning. In: European Conference on Computer Vision (ECCV), pp 548–561
36. Marszalek M, Laptev I, Schmid C (2009) Actions in context. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2929–2936

37. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8
38. Wang J, Chen Z, Wu Y (2011) Action recognition with multi-scale spatio-temporal contexts. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 3185–3192
39. Ji S, Xu W, Yang M, Yu K (2013) 3D convolutional neural networks for human action recognition In: IEEE Transactions on Pattern Analysis and Machine Intelligence
40. Han D, Bo L, Sminchisescu C (2009) Selection and context for action recognition. In: IEEE conference on computer vision (ICCV), pp 1933–1940
41. Gilbert A, Illingworth J, Bowden R (2009) Fast realistic multi-action recognition using mined dense spatio-temporal features. In: IEEE conference on computer vision (ICCV), pp 925–931
42. Ullah M, Parizi S, Laptev I (2010) Improving bag-of-features action recognition with non-local cues. In: British Machine Vision Conference (BMVC)
43. Chakraborty B, Holte M, Moeslund T, Gonzààlez J (2012) Selective spatio-temporal interest points. Comput Vis Image Underst 116(3):396–410