A Discriminative Model with Multiple Temporal Scales for Action Prediction

Yu Kong[1], Dmitry Kit[1], and Yun Fu[1,2]

[1] Department of Electrical and Computer Engineering,
Northeastern University, Boston, MA, USA
[2] College of Computer and Information Science,
Northeastern University, Boston, MA, USA
{yukong,dkit,yunfu}@ece.neu.edu

Abstract. The speed with which intelligent systems can react to an action depends on how soon it can be recognized. The ability to recognize ongoing actions is critical in many applications, for example, spotting criminal activity. It is challenging, since decisions have to be made based on partial videos of temporally incomplete action executions. In this paper, we propose a novel discriminative multi-scale model for predicting the action class from a partially observed video. The proposed model captures temporal dynamics of human actions by explicitly considering all the history of observed features as well as features in smaller temporal segments. We develop a new learning formulation, which elegantly captures the temporal evolution over time, and enforces the label consistency between segments and corresponding partial videos. Experimental results on two public datasets show that the proposed approach outperforms state-of-the-art action prediction methods.

Keywords: Action Prediction, Structured SVM, Sequential Data.

1 Introduction

Human action recognition [17,10,8,18] has been of great interest for the computer vision community for many decades due to its practical importance, such as video analysis and visual surveillance. A majority of action recognition approaches focus on classifying the action after fully observing the entire video. However, in many real-world scenarios (e.g. vehicle accident and criminal activity), intelligent systems do not have the luxury of waiting for the entire video before having to react to the action contained in it. For example, being able to predict a dangerous driving situation before it occurs; opposed to recognizing it thereafter. Unfortunately, most of the existing action recognition approaches are unsuitable for such early classification tasks as they expect to see the entire set of action dynamics extracted from a full video.

Different from action recognition, visual data arrives sequentially in action prediction. Therefore, to achieve accurate prediction as early as possible, it is essential to maximize the discriminative power of the beginning temporal segments in an action video. In addition, accurate action prediction relies on effectively

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 596–611, 2014.

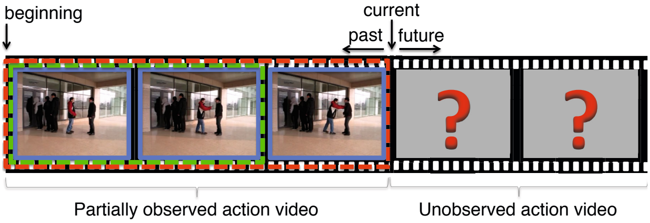


Fig. 1. Our method predicts action label given a partially observed video. Action dynamics are captured by both local templates (solid rectangles) and global templates (dashed rectangles).

utilizing useful history action information. As the action data are progressively observed, the confidence of the partial history observations should also increase.

In this paper, we propose a novel multiple temporal scale support vector machine (MTSSVM) for the early recognition of unfinished actions. Our model characterizes human actions at two different temporal granularities (Fig. 1) to learn the evolution and dynamics of actions, and predicts action labels from partially observed videos containing temporally incomplete action executions. Local templates in the MTSSVM consider the sequential nature of human actions at the fine granularity. The discriminative power of the beginning temporal segments are maximized by enforcing their label consistency. The temporal arrangements of these local templates also implicitly capture temporal orderings of inhomogeneous action segments.

We also build coarse global templates to capture the history action information. The global templates summarize action evolutions at different temporal lengths, from the start of the video to the current point in time. Our model uses this information to learn how to differentiate between classes using all available information. For example, for the action class "Push" the important feature is that the "arm is up", which can be used to distinguish it from the class "Kick". By learning a model for increasing amount of information, our model captures the evolution of actions in each class.

We develop a new convex learning formulation based on the structured SVM to consider the nature of the sequentially arriving action videos. This is achieved by introducing new constraints into the learning formulation. We enforce the label consistency between segments and their corresponding full video to maximize the discriminative power of the beginning temporal segments. In addition, we introduce a principled monotonic score function for the global template. This allows us to use the prior knowledge that informative action information is increasing as the data arrive sequentially. We show in Section 3.3 that the objective of the new learning formulation minimizes an upper bound of the empirical risk of the training data.

2 Related Work

Action Recognition: Human actions [17,26,15,3] have been popularly represented by a set of quantized local spatiotemporal features, known as bag-of-words. Bag-of-words models have shown to be robust to background noise but may not be expressive enough to describe actions in the presence of large appearance and pose variations. This problem has been addressed by introducing human knowledge into models and using semantic descriptions or attributes to characterize complex human actions [7,8,10]. In addition, recognizing human actions from a set of keyframes [12,22] and static images [25,24] have also been investigated in previous studies. However, most of existing action recognition methods were designed for recognizing complete actions, assuming the action in each testing video has been fully executed. This makes these approaches unsuitable for predicting action labels in partial videos.

Another line of work captures temporal evolutions of appearance or pose using sequential state models [11,23,20,19]. These approaches treat a video as a composition of temporal segments. However, they do not model temporal action evolution with respect to observation ratios. Therefore, they cannot characterize partially observed actions and are unsuitable for prediction. In contrast, we simulate the sequential data arrival in prediction and use large temporal scale templates to capture action evolutions from the beginning of the video to the current observed frame. Therefore, our model can recognize incomplete actions at different observation ratios.

Action Prediction: Most of the existing work in action prediction aims at recognizing unfinished action videos. Ryoo [14] proposed the integral bag-of-words (IBoW) and dynamic bag-of-words (DBoW) approaches for action prediction. The action model of each progress level is computed by averaging features of a particular progress level in the same category. However, the learned model may not be representative if the action videos of the same class have large appearance variations, and it is sensitive to outliers. To overcome these two problems, Cao *et al.*[1] built action models by learning feature bases using sparse coding and used the reconstruction error in the likelihood computation. Li *et al.*[9] explored long-duration action prediction problem. However, their work detects segments by motion velocity peaks, which may not be applicable on complex outdoor datasets. Compared with [1,9,14], our model incorporates an important prior knowledge that informative action information is increasing when new observations are available. However, their methods have not taken advantage of this prior. In addition, our method models label consistency of segments, which is not presented in their methods. The label consistency provides discriminative local information and implicitly captures context information, which is beneficial for the prediction task. Moreover, we capture action dynamics in both global and local temporal scales while [1,14] capture dynamics in one single scale.

Additionally, an early event detector [4] was proposed to localize the starting and ending frames of an incomplete event, which is different from our goal. Activity forecasting, which aims at reasoning about the preferred path for people given a destination in a scene, has been investigated in [6].

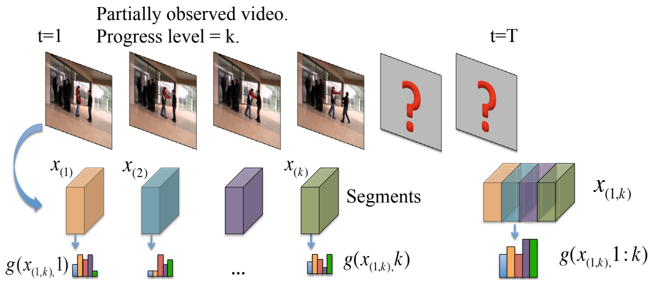


Fig. 2. Example of video segments $x_{(k)}$, partial video $x_{(1,k)}$, feature representations $g(x_{(1,k)}, l)$ of segments ($l = 1, \cdots, k$), and the representation of the partial video $g(x_{(1,k)}, 1 : k)$

3 Our Method

The aim of this work is to predict the action class y of a partially observed action video $x[1, t]$ before the action ends. Here 1 and t in $x[1, t]$ indicate the indices of the starting frame and the last observed frame of the partial video $x[1, t]$, respectively. Index t ranges from 1 to length T of a full video $x[1, T]$: $t \in \{1, \cdots, T\}$, to generate different partial videos. An action video is usually composed of a set of inhomogeneous temporal units, which are called segments. In this work, we uniformly divide a full video $x[1, T]$ into K segments $x[\frac{T}{K} \cdot (l-1) + 1, \frac{T}{K} \cdot l]$, where $l = 1, \cdots, K$ is the index of segment. The length of each segment is $\frac{T}{K}$. Note that for different videos, their lengths T may be different. Therefore, the length of segments of various videos may be different. For simplicity, let $x_{(k)}$ be the k-th segment $x[\frac{T}{K} \cdot (k-1) + 1, \frac{T}{K} \cdot k]$ and $x_{(1,k)}$ be the partially observed sequence $x[1, \frac{T}{K} \cdot k]$ (see Fig. 2). The progress level k of a partially observed video is defined as the number of observed segments that the video has. The observation ratio is the ratio of the number of frames in a partially observed video $x[1, t]$ to the number of frames in the full video $x[1, T]$, which is $\frac{t}{T}$. For example, if $T = 100$, $t = 30$ and $K = 10$, then the progress level of the partially observed video $x[1, t]$ is 3 and its observation ratio is 0.3.

3.1 Action Representations

We use the bag-of-words models to represent segments and partial videos. The procedure of learning the visual word dictionary for action videos is as follows. Spatiotemporal interest points detector [3] and tracklet [13] are employed to extract interest points and trajectories from a video, respectively. The dictionaries of visual words are learned by clustering algorithms.

We denote the feature of the partial video $x_{(1,k)}$ at progress level k by $g(x_{(1,k)}, 1 : k)$, which is the histogram of visual words contained in the entire partial video, starting from the first segment to the k-th segment (Fig. 2). The representation of the l-th ($l \in \{1, \cdots, k\}$) segment $x_{(l)}$ in the partial video is denoted by $g(x_{(1,k)}, l)$, which is a histogram of visual words whose temporal locations are within the l-th segment.

3.2 Model Formulation

Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ be the training data, where x_i is the i-th fully observed action video and y_i is the corresponding action label. The problem of action prediction is to learn a function $f : \mathcal{X} \to \mathcal{Y}$, which maps a partially observed video $x_{(1,k)} \in \mathcal{X}$ to an action label $y \in \mathcal{Y}$ ($k \in \{1, \cdots, K\}$).

We formulate the action prediction problem using the structured learning as presented in [21]. Instead of searching for f, we aim at learning a discriminant function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}$ to score each training sample (x, y). The score measures the compatibility between a video x and an action label y. Note that, in action prediction, videos of different observation ratios from the same class should be classified as the same action category. Therefore, we use the function F to score the compatibility between the videos of different observation ratios $x_{(1,k)}$ and the action label y, where $k \in \{1, \cdots, K\}$ is the progress level.

We are interested in a linear function $F(x_{(1,k)}, y; \mathbf{w}) = \langle \mathbf{w}, \Phi(x_{(1,k)}, y) \rangle$, which is a family of functions parameterized by $\mathbf{w}$, and $\Phi(x_{(1,k)}, y)$ is a joint feature map that represents the spatio-temporal features of action label y given a partial video $x_{(1,k)}$. Once the optimal model parameter $\mathbf{w}^*$ is learned, the prediction of the action label y^* is computed by

$$y^* = \arg\max_{y \in \mathcal{Y}} F(x_{(1,k)}, y; \mathbf{w}^*) = \arg\max_{y \in \mathcal{Y}} \langle \mathbf{w}^*, \Phi(x_{(1,k)}, y) \rangle. \tag{1}$$

We define $\mathbf{w}^{\mathrm{T}} \Phi(x_{(1,k)}, y)$ as a summation of the following two components:

$$\mathbf{w}^{\mathrm{T}} \Phi(x_{(1,k)}, y) = \alpha_k^{\mathrm{T}} \psi_1(x_{(1,k)}, y) + \sum_{l=1}^{K} \Big[\mathbf{1}(l \leqslant k) \cdot \beta_l^{\mathrm{T}} \psi_2(x_{(1,k)}, y) \Big], \tag{2}$$

where $\mathbf{w} = \{\alpha_1, \cdots \alpha_K, \beta_1, \cdots, \beta_K\}$ is model parameter, k is the progress level of the partial video $x_{(1,k)}$, l is the index of progress levels, and $\mathbf{1}(\cdot)$ is the indicator function. The two components in Eq.(2) are summarized as follows.

Global Progress Model (GPM). $\alpha_k^{\mathrm{T}} \psi_1(x_{(1,k)}, y)$ indicates how likely the action class of an unfinished action video $x_{(1,k)}$ (at progress level k) is y. We define GPM as

$$\alpha_k^{\mathrm{T}} \psi_1(x_{(1,k)}, y) = \sum_{a \in \mathcal{Y}} \alpha_k^{\mathrm{T}} \mathbf{1}(y = a) g(x_{(1,k)}, 1:k). \tag{3}$$

Here, feature vector $g(x_{(1,k)}, 1{:}k)$ of dimensionality D is an action representation for the partial video $x_{(1,k)}$, where features are extracted from the entire partial video, from its beginning (i.e., progress level 1) to its current progress level k. Parameter α_k of size $D \times |\mathcal{Y}|$ can be regarded as a progress level-specific template. Since the partial video is at progress level k, we select the template α_k at the same progress level, from K parameter matrices $\{\alpha_1, \cdots, \alpha_K\}$. The selected template α_k is used to score the unfinished video $x_{(1,k)}$. Define $A = [\alpha_1, \cdots, \alpha_K]$ as a vector of all the parameter matrices in the GPM. Then A is a vector of size $D \times K \times |\mathcal{Y}|$ encoding the weights for the configurations between progress levels and action labels, with their corresponding video evidence.

The GPM simulates the sequential segment-by-segment data arrival for training action videos. Essentially, the GPM captures the action appearance changes as the progress level increases, and characterizes the entire action evolution over time. In contrast to the IBoW model [14], our GPM does not assume any distributions on the data likelihood; while the IBoW model uses the Gaussian distribution. In addition, the compatibility between observation and action label in our model is given by the linear model of parameter and feature function, rather than using a Gaussian kernel function [14].

Local Progress Model (LPM). $\mathbf{1}(l \leqslant k) \cdot \beta_l^{\mathrm{T}} \psi_2(x_{(1,k)}, y)$ indicates how likely the action classes of all the temporal segments $x_{(l)}$ $(l = 1, \cdots, k)$ in an unfinished video $x_{(1,k)}$ are all y. Here, the progress level of the partial video is k and we consider all the segments of the video whose temporal locations l are smaller than k. We define LPM as

$$\beta_l^{\mathrm{T}} \psi_2(x_{(1,k)}, y) = \sum_{a \in \mathcal{Y}} \beta_l^{\mathrm{T}} \mathbf{1}(y = a) g(x_{(1,k)}, l), \tag{4}$$

where feature vector $g(x_{(1,k)}, l)$ of dimensionality D extracts features from the l-th segment of the unfinished video $x_{(1,k)}$. β_l of size $D \times |\mathcal{Y}|$ is the weight matrix for the l-th segment. We use the indicator function $\mathbf{1}(l \leqslant k)$ to select all the segment weight matrices, $\beta_1, \cdots, \beta_k$, whose temporal locations are smaller than or equal to the progress level k of the video. Then the selected weight matrices are used to score the corresponding segments. Let $B = [\beta_1, \cdots, \beta_K]$ be a vector of all the parameters in the LPM. Then B is a vector of size $D \times K \times |\mathcal{Y}|$ encoding the weights for the configurations between segments and action labels, with their corresponding segment evidence.

The LPM considers the sequential nature of a video. The model decomposes a video of progress level k into segments and describes the temporal dynamics of segments. Note that the action data preserve the temporal relationship between the segments. Therefore, the discriminative power of segment $x_{(k)}$ is critical to the prediction of $x_{(1,k)}$ given the prediction results of $x_{(1,k-1)}$. In this work, the segment score $\beta_k^{\mathrm{T}} g(x_{(1,k)}, k)$ measures the compatibility between the segment $x_{(k)}$ and all the classes. To maximize the discriminability of the segment, the score difference between the ground-truth class and all the other classes is maximized in our learning formulation. Thus, accurate prediction can be achieved using the newly-introduced discriminative information in the segment $x_{(k)}$.

3.3 Structured Learning Formulation

The MTSSVM is formulated based on the structured SVM [21,5]. The optimal model parameter $\mathbf{w}^*$ of MTSSVM in Eq.(1) is learned by solving the following convex problem given training data $\{x_i, y_i\}_{i=1}^N$:

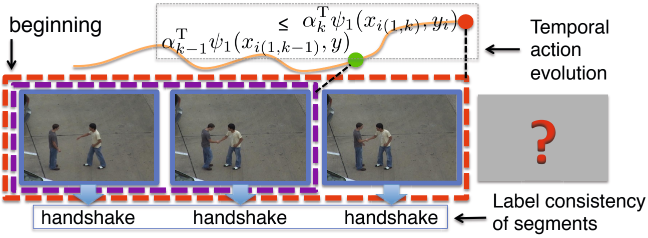

beginning
$\leq \alpha_k^{\mathrm{T}}\psi_1(x_{i(1,k)}, y_i)$
$\alpha_{k-1}^{\mathrm{T}}\psi_1(x_{i(1,k-1)}, y)$
Temporal action evolution
?
handshake
handshake
handshake
Label consistency of segments

Fig. 3. Graphical illustration of the temporal action evolution over time and the label consistency of segments. Blue solid rectangles are LPMs, and purple and red dashed rectangles are GPMs.

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N}(\xi_{1i} + \xi_{2i} + \xi_{3i}) \tag{5}$$

$$\text{s.t. } \mathbf{w}^{\mathrm{T}}\Phi(x_{i(1,k)}, y_i) \geqslant \mathbf{w}^{\mathrm{T}}\Phi(x_{i(1,k)}, y) + K\delta(y, y_i) - \frac{\xi_{1i}}{u(k/K)}, \quad \forall i, \forall k, \forall y, \tag{6}$$

$$\alpha_k^{\mathrm{T}}\psi_1(x_{i(1,k)}, y_i) \geqslant \alpha_{k-1}^{\mathrm{T}}\psi_1(x_{i(1,k-1)}, y) + K\delta(y, y_i) - \frac{\xi_{2i}}{u(k/K)}, \quad \forall i, k = 2, \cdots, K, \forall y, \tag{7}$$

$$\beta_k^{\mathrm{T}}\psi_2(x_{i(k)}, y_i) \geqslant \beta_k^{\mathrm{T}}\psi_2(x_{i(k)}, y) + kK\delta(y, y_i) - \frac{\xi_{3i}}{u(1/K)}, \quad \forall i, \forall k, \forall y, \tag{8}$$

where C is the slack trade-off parameter similar to that in SVM. ξ_{1i}, ξ_{2i} and ξ_{3i} are slack variables. $u(\cdot)$ is a scaling factor function: $u(p) = p$. $\delta(y, y_i)$ is the 0-1 loss function.

The slack variables ξ_{1i} and the Constraint (6) are usually used in SVM constraints on the class labels. We enforce this constraint for all the progress levels k since we are interested in learning a classifier that can correctly recognize partially observed videos with different progress levels k. Therefore, we simulate the segment-by-segment data arrival for training and augment the training data with partial videos of different progress levels. The loss function $\delta(y, y_i)$ measures the recognition error of a partial video and the scaling factor $u(\frac{k}{K})$ scales the loss based on the length of the partial video.

Constraint (7) considers **temporal action evolution** over time (Fig. 3). We assume that the score $\alpha^{\mathrm{T}}\psi_1(x_{i(1,k)}, y_i)$ of the partial observation $x_{i(1,k)}$ at progress level k and ground truth label y_i must be greater than the score $\alpha^{\mathrm{T}}\psi_1(x_{i(1,k-1)}, y)$ of a previous observation $x_{i(1,k-1)}$ at progress level $k-1$ and all incorrect labels y. This provides a monotonically increasing score function for partial observations and elaborately characterizes the nature of sequentially arriving action data in action prediction. The slack variable ξ_{2i} allows us to model outliers.

The slack variables ξ_{3i} and the Constraint (8) are used to maximize the discriminability of segments $x_{(k)}$. We encourage the **label consistency** between segments and the corresponding full video due to the nature of sequential data in action prediction (Fig. 3). Assume a partial video $x_{(1,k-1)}$ has been correctly recognized, then the segment $x_{(k)}$ is the only newly-introduced information and its discriminative power is the key to recognizing the video $x_{(1,k)}$. Moreover, context information of segments is implicitly captured by enforcing the label consistency. It is possible that some segments from different classes are visually similar and may not be linearly separable. We use the slack variable ξ_{3i} for each video to allow some segments of a video to be treated as outliers.

Empirical Risk Minimization: We define $\varDelta(y_i, y)$ as the function that quantifies the loss for a prediction y, if the ground-truth is y_i. Therefore, the loss of a classifier $f(\cdot)$ for action prediction on a video-label pair (x_i, y_i) can be quantified as $\varDelta(y_i, f(x_i))$. Usually, the performance of $f(\cdot)$ is given by the empirical risk $R_{\text{emp}}(f) = \frac{1}{N}\sum_{i=1}^{N} \varDelta(y_i, f(x_i))$ on the training data (x_i, y_i), assuming data samples are generated i.i.d.

The nature of continual evaluation in action prediction requires aggregating the values of loss quantities computed during the action sequence process. Define the loss associated with a prediction $y = f(x_{i(1,k)})$ for an action x_i at progress level k as $\varDelta(y_i, y)u(\frac{k}{K})$. Here $\varDelta(y_i, y)$ denotes the misclassification error, and $u(\frac{k}{K})$ is the scaling factor that depends on how many segments have been observed. In this work, we use summation to aggregate the loss quantities. This leads to an empirical risk for N training samples: $R_{\text{emp}}(f) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\left\{\varDelta(y_i, y)u(\frac{k}{K})\right\}$.

Denote by ξ_1^*, ξ_2^* and ξ_3^* the optimal solutions of the slack variables in Eq. (5-8) for a given classifier f, we can prove that $\frac{1}{N}\sum_{i=1}^{N}(\xi_{1i}^*+\xi_{2i}^*+\xi_{3i}^*)$ is an upper bound on the empirical risk $R_{\text{emp}}(f)$ and the learning formulation given in Eq. (5-8) minimizes the upper bound of the empirical risk $R_{\text{emp}}(f)$[1].

3.4 Discussion

We highlight here some important properties of our model, and show some differences from existing methods.

Multiple Temporal Scales. Our method captures action dynamics in both local and global temporal scales, while [1,4,14] only use a single temporal scale.

Temporal Evolution over Time. Our work uses the prior knowledge of temporal action evolution over time. Inspired by [4], we introduce a principled monotonic score function for the GPM to capture this prior knowledge. However, [4] aims at finding the starting frame of an event while our goal is to predict action class of an unfinished video. The methods in [1,14,9] do not use this prior.

Segment Label Consistency. We effectively utilize the discriminative power of local temporal segments by enforcing label consistency of segments. However, [1,14,9,4] do not consider the label consistency. The consistency also implicitly

[1] Please refer to the supplemental material for details.

models temporal segment context by enforcing the same label for segments while [1,14,4] explicitly treat successive temporal segments independently.

Principled Empirical Risk Minimization. We propose a principled empirical risk minimization formulation for action prediction, which is not discussed in [1,14,9].

3.5 Model Learning and Testing

Learning. We solve the optimization problem (5-8) using the regularized bundle algorithm [2]. The basic idea of the algorithm is to iteratively approximate the objective function by adding a new cutting plane to the piecewise quadratic approximation.

The equivalent unconstrained problem of the optimization problem (5-8) is $\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N} \cdot L(\mathbf{w})$, where $L(\mathbf{w}) = \sum_{i=1}^{N}(U_i + Z_i + V_i)$ is the empirical loss. Here, U_i, Z_i and V_i are given by

$$U_i = \sum_{k=1}^{K} u(\frac{k}{K}) \max_{y} \Big[K\delta(y, y_i) + \mathbf{w}^{\mathrm{T}}\Phi(x_{i(1,k)}, y) - \mathbf{w}^{\mathrm{T}}\Phi(x_{i(1,k)}, y_i)\Big], \tag{9}$$

$$Z_i = \sum_{k=2}^{K} u(\frac{k}{K}) \max_{y} \Big[K\delta(y, y_i) + \alpha_{k-1}^{\mathrm{T}}\psi_1(x_{i(1,k-1)}, y) - \alpha_k^{\mathrm{T}}\psi_1(x_{i(1,k)}, y_i)\Big], \tag{10}$$

$$V_i = \sum_{k=1}^{K} u(\frac{1}{K}) \max_{y} \Big[kK\delta(y, y_i) + \beta_k^{\mathrm{T}}\psi_2(x_{i(k)}, y) - \beta_k^{\mathrm{T}}\psi_2(x_{i(k)}, y_i)\Big]. \tag{11}$$

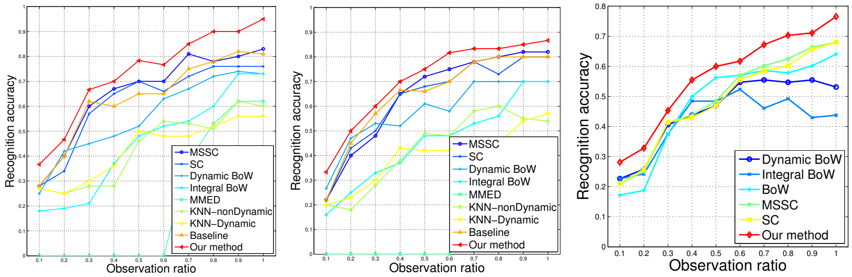
The regularized bundle algorithm requires the subgradient of the training loss with respect to the parameter, $\frac{\partial L}{\partial \mathbf{w}} = \sum_{i=1}^{N}(\frac{\partial U_i}{\partial \mathbf{w}} + \frac{\partial Z_i}{\partial \mathbf{w}} + \frac{\partial V_i}{\partial \mathbf{w}})$, in order to find a new cutting plane to be added to the approximation[2].

Testing. Given an unfinished action video with progress level k (k is known in testing), our goal is to infer the class label y^* using the learned model parameter $\mathbf{w}^*$: $y^* = \arg\max_{y\in\mathcal{Y}}\langle \mathbf{w}^*, \Phi(x_{(1,k)}, y)\rangle$. Note that testing phase does not require sophisticated inference algorithms such as belief propagation or graph cut since we do not explicitly capture segment interactions. However, the context information between segments is implicitly captured in our model by the label consistency in Constraint (8).

4 Experiments

We test the proposed MTSSVM approach on three datasets: the UT-Interaction dataset (UTI) Set 1 (UTI #1) and Set 2 (UTI #2) [16], and the BIT-Interaction dataset (BIT) [7]. UTI #1 were taken on a parking lot with mostly static background and little camera jitters. UTI #2 were captured on a lawn with slight

[2] Please refer to the supplemental material for details.



(a) UTI #1 dataset. (b) UTI #2 dataset. (c) BIT dataset.

Fig. 4. Prediction results on the UTI #1, UTI #2 and BIT dataset

background movements (e.g. tree moves) and camera jitters. Both of the two sets consist of six types of human actions, with ten videos per class. We adopt the leave-one-out training scheme on the two datasets. The BIT dataset consists of eight types of human actions between two people, with fifty videos per class. For this dataset, a random sample of 272 videos is chosen as training samples, and the remaining 128 videos are used for testing. The dictionary size for interest point descriptors is set to 500, and the size for tracklet descriptors is automatically determined by the clustering method in all the experiments.

MTSSVM is evaluated for classifying videos of incomplete action executions using 10 observation ratios, from 0.1 to 1, representing the increasing amount of sequential data with time. For example, if a full video containing T frames is used for testing at the observation ratio of 0.3, the accuracy of MTSSVM is evaluated by presenting it with the first $0.3 \times T$ frames. At observation ratio of 1, the entire video is used, at which point MTSSVM acts as a conventional action recognition model. The progress level k of testing videos is known to all the methods in our experiments.

4.1 Results

UTI #1 and UTI #2 Datasets. The MTSSVM is compared with DBoW and IBoW in [14], the MMED [4], the MSSC and the SC in [1], and the method in [12]. The KNN-nonDynamic, the KNN-Dynamic, and the baseline method implemented in [1] are also used in comparison. The same experiment settings in [1] are followed in our experiments.

Fig. 4(a) shows the prediction results on the UTI #1 dataset. Our MTSSVM achieves better performance over all the other comparison approaches. Our method outperforms the MSSC method because we not only model segment dynamics but also characterize temporal evolutions of actions. Our method can achieve an impressive 78.33% recognition accuracy when only the first 50% frames of testing videos are observed. This result is even higher than the SC method with full observations. Results of our method are significantly higher than the DBoW and IBoW for all observation ratios. This is mainly due to the

Table 1. Prediction results compared with [12] on half and full videos

Observation ratio	Accuracy with half videos	Accuracy with full videos
Raptis and Sigal [12]	73.3%	93.3%
Our model	**78.33**%	**95**%

fact that the action models in our work are discriminatively learned while the action models in the DBoW and IBoW are computed by averaging feature vectors in a particular class. Therefore, the action models in the DBoW and IBoW may not be the representative models and are sensitive to outliers. MMED does not perform well as other prediction approaches since it is optimized for early detection of the starting and ending frame of an action. This is a different goal from this paper, which is to classify unfinished actions. We also compare with [12] on half and full video observations. Results in Table 1 show that our method achieves better performance over [12].

Comparison results on the UTI #2 datasets are shown in Fig. 4(b). The MTSSVM achieves better performance over all the other comparison approaches in all the cases. At 0.3, 0.5 and 1 observation ratios, MSSC achieves 48.33%, 71.67%, and 81.67% prediction accuracy, respectively, and SC achieves 50%, 66.67%, and 80% accuracy, respectively. By contrast, our MTSSVM achieves 60%, 75% and 83.33% prediction results, respectively, which is consistently higher than MSSC and SC. Our MTSSVM achieves 75% accuracy when only the first 50% frames of testing videos are observed. This accuracy is even higher than the DBoW and IBoW with full observations.

To demonstrate that both the global progress model (GPM) and the local progress model (LPM) are important for action prediction, we compare the performance of MTSSVM with the model that only uses one of the two sources of information on the UTI #1 dataset. Fig. 5 shows the scores of the GPM and LPM ($\alpha_k^{\mathrm{T}}\psi_1(x_{(1,k)}, y)$ of the GPM and $\sum_{l=1}^{K}\mathbf{1}(l \leqslant k)\cdot\beta_l^{\mathrm{T}}\psi_2(x_{(1,k)}, y)$ of the LPM), and compare them to the scores of the full MTSSVM model with respect to the observation ratio. Results show that the LPM captures discriminative temporal segments for prediction. LPM characterizes temporal dynamics of segments and discriminatively learns to differentiate segments from different classes. In most cases, the score of LPM is monotonically increasing, which indicates a discriminative temporal segment is used for prediction. However, in some cases, segments from different classes are visually similar and thus are difficult to discriminate. Therefore, in the middle of the “handshake” class and the “hug” class in Fig. 5 (observation ratio from 0.3 to 0.7), adding more segment observations does not increase LPM’s contribution to MTSSVM. Fig. 6 shows examples of visually similar segments of the two classes at $k = 6$. However, when such situations arise, GPM can provide necessary appearance history information and therefore increases the prediction performance of MTSSVM.

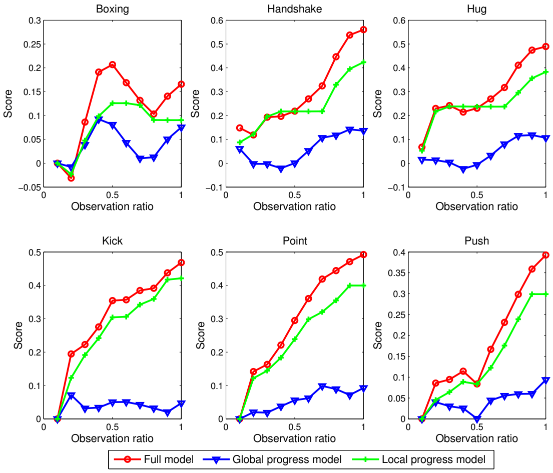


Fig. 5. Contributions of the global progress model and the local progress model to the prediction task

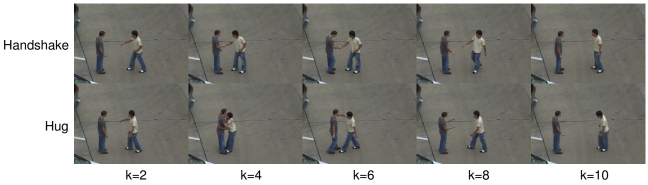


Fig. 6. Examples of segments in “handshake” and “hug”. Segments $k = 6, 8, 10$ in the two classes are visually similar.

BIT-Interaction Dataset. We also compare MTSSVM with the MSSC, SC, DBoW and IBoW on the BIT-Interaction dataset. A BoW+SVM method is used as a baseline. The parameter σ in DBoW and IBoW is set to 36 and 2, respectively, which are the optimal parameters on the BIT-Interaction dataset. Results shown in Fig. 4(c) demonstrate that MTSSVM outperforms MSSC and SC in all cases due to the effect of the global progress model, which effectively captures temporal action evolution information. MTSSVM also outperforms the DBoW and IBoW. Our method achieves 60.16% recognition accuracy with only the first 50% frames of testing videos are observed, which is better than the DBoW and IBoW at all observation ratios. Note that the performance of DBoW and IBoW do not increase much when the observation ratios are increased from 0.6 to 0.9. The IBoW performs even worse. This is due to the fact that some video segments from different classes are visually similar; especially the segments in the second half of the videos, where people return to their starting positions (see Fig. 7). However, because MTSSVM models both the segments and the

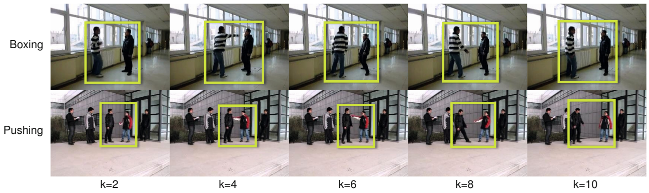


Fig. 7. Examples of visually similar segments in the "boxing" action (Top) and the "pushing" action (Bottom) with segment index $k \in \{2, 4, 6, 8, 10\}$. Bounding boxes indicate the interest regions of actions

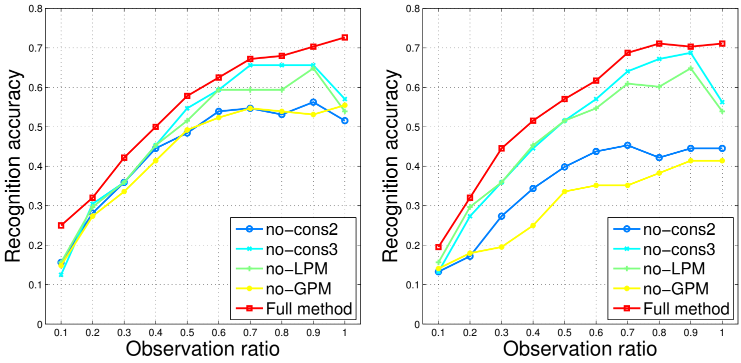
entire observation, its performance increases with the increasing of observation ratio even if the newly introduced segments contain only a small amount of discriminative information.

We further investigate the sensitivity of MTSSVM to the parameters C in Eq. (5). We set C to 0.5, 5, and 10, and test MTSSVM on all parameter combinations with observation ratios 0.3, 0.5, and 0.8. Results in Table 2 indicate that MTSSVM is not sensitive to the parameters when the observation ratio is low but the sensitivity increases when the observation ratio becomes large. In the beginning of a video, the small number of features available does not capture the variability of their class. Therefore, it does not help to use different parameters, because MTSSVM cannot learn the appropriate class boundaries to separate all the testing data. As observation ratio increases, the features become more expressive. However, since structural features in MTSSVM are very complex, appropriate parameters are required to capture the complexity of data.

Table 2. Recognition accuracy of our model on videos of observation ratio 0.3, 0.5, and 0.8 with different C parameters

Observation ratio	C=0.5	C=5	C=10
0.3	42.97%	39.84%	38.28%
0.5	54.69%	57.03%	51.56%
0.8	66.41%	61.72%	55.47%

Finally, we also evaluate the importance of each component in the MTSSVM, including the Constraint (7), the Constraint (8), the local progress model (LPM in Eq. (4)) and the global progress model (GPM in Eq. (3)). We remove each of these components from the MTSSVM, and obtain four variant models, the no-cons2 model (remove the Constraint (7) from MTSSVM), the no-cons3 model (remove the Constraint (8)), the no-LPM model (remove the LPM and Constraint (8)), and the no-GPM model (remove the GPM and Constraint (7)). We compare MTSSVM with these variants with parameter C of 1 and 100. Results in Fig. 8 show that the GPM is the key component in the MTSSVM.



(a) Parameter $C = 1$ (b) Parameter $C = 100$

Fig. 8. Prediction results of each component in the full MTSSVM with C parameter 1 and 100

Without the GPM, the performance of the no-GPM model degrades significantly compared with the full MTSSVM model, especially with parameter C of 100. The performances of the no-cons3 model and the no-LPM model are worse compared with the full method in all cases. This is due to the lack of the segment label consistency in the two models. The label consistency can help use the discriminative information in segments and also implicitly model context information. In the ending part of videos in BIT dataset, since most of observations are visually similar (people return back to their normal position), label consistency is of great importance for discriminating classes. However, due to the lack of label consistency in the the no-cons3 model and the no-LPM model, they cannot capture useful information for differentiating action classes.

5 Conclusion

We have proposed the multiple temporal scale support vector machine (MTSSVM) for recognizing actions in incomplete videos. MTSSVM captures the entire action evolution over time and also considers the temporal nature of a video. We formulate the action prediction task as a structured SVM learning problem. The discriminability of segments is enforced in the learning formulation. Experiments on two datasets show that MTSSVM outperforms state-of-the-art approaches.

Acknowledgements. This research is supported in part by the NSF CNS award 1314484, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, U.S. Army Research Office Young Investigator Award W911NF-14-1-0218, and IC Postdoc Program Grant 2011-11071400006.

References

1. Cao, Y., Barrett, D., Barbu, A., Narayanaswamy, S., Yu, H., Michaux, A., Lin, Y., Dickinson, S., Siskind, J., Wang, S.: Recognizing human activities from partially observed videos. In: CVPR (2013)
2. Do, T.-M.-T., Artieres, T.: Large margin training for hidden markov models with partially observed states. In: ICML (2009)
3. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS (2005)
4. Hoai, M., De la Torre, F.: Max-margin early event detectors. In: CVPR (2012)
5. Joachims, T., Finley, T., Yu, C.-N.: Cutting-plane training of structural svms. Machine Learning 77(1), 27–59 (2009)
6. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 201–214. Springer, Heidelberg (2012)
7. Kong, Y., Jia, Y., Fu, Y.: Learning human interaction by interactive phrases. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 300–313. Springer, Heidelberg (2012)
8. Kong, Y., Jia, Y., Fu, Y.: Interactive phrases: Semantic descriptions for human interaction recognition. TPAMI (2014)
9. Li, K., Hu, J., Fu, Y.: Modeling complex temporal composition of actionlets for activity prediction. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 286–299. Springer, Heidelberg (2012)
10. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: CVPR (2011)
11. Niebles, J.C., Chen, C.-W., Fei-Fei, L.: Modeling temporal structure of decomposable motion segments for activity classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 392–405. Springer, Heidelberg (2010)
12. Raptis, M., Sigal, L.: Poselet key-framing: A model for human activity recognition. In: CVPR (2013)
13. Raptis, M., Soatto, S.: Tracklet descriptors for action modeling and video analysis. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 577–590. Springer, Heidelberg (2010)
14. Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: ICCV (2011)
15. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV, pp. 1593–1600 (2009)
16. Ryoo, M., Aggarwal, J.: UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities, SDHA (2010)
17. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: ICPR, vol. 3, pp. 32–36. IEEE (2004)
18. Shapovalova, N., Vahdat, A., Cannons, K., Lan, T., Mori, G.: Similarity constrained latent support vector machine: An application to weakly supervised action classification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 55–68. Springer, Heidelberg (2012)

19. Shi, Q., Cheng, L., Wang, L., Smola, A.: Human action segmentation and recognition using discriminative semi-markov models. IJCV 93, 22–32 (2011)
20. Tang, K., Fei-Fei, L., Koller, D.: Learning latent temporal structure for complex event detection. In: CVPR (2012)
21. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. JMLR 6, 1453–1484 (2005)
22. Vahdat, A., Gao, B., Ranjbar, M., Mori, G.: A discriminative key pose sequence model for recognizing human interactions. In: ICCV Workshops. pp. 1729–1736 (2011)
23. Wang, Z., Wang, J., Xiao, J., Lin, K.-H., Huang, T.S.: Substructural and boundary modeling for continuous action recognition. In: CVPR (2012)
24. Yao, B., Fei-Fei, L.: Action recognition with exemplar based 2.5D graph matching. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 173–186. Springer, Heidelberg (2012)
25. Yao, B., Fei-Fei, L.: Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. TPAMI 34(9), 1691–1703 (2012)
26. Yu, T.-H., Kim, T.-K., Cipolla, R.: Real-time action recognition by spatiotemporal semantic and structural forests. In: BMVC (2010)