Chapter 6
Activity Prediction

Yu Kong and Yun Fu

1 Introduction

Human action recognition [8, 11, 18, 19] is one of the active topics in the computer vision community, and has a broad range of applications, for example, video retrieval, visual surveillance, and video understanding.

After fully observing the entire video, action recognition approaches will classify the video observation into one of the action categories. It should be noted that certain real-world applications (e.g., vehicle accident and criminal activity) do not allow the luxury of waiting for the entire action to be executed. Reactions must be performed in a prompt to the action. For instance, it is extremely important to predict a dangerous driving situation before any vehicle crash occurs. Unfortunately, a majority of the existing action recognition approaches are limited to such particular scenarios since they must fully observe the action sequence extracted from the video.

One of the major differences between action prediction and action recognition is that action video data arrive sequentially in action prediction. However, action recognition takes the full observation as input. The key to perform early classification accurately is to extract the most discriminative information from the beginning segments in a temporal sequence. Furthermore, it is also important to

Y. Kong (✉)
Department of Electrical and Computer Engineering, Northeastern University,
360 Huntington Avenue, Boston, MA 02115, USA
e-mail: yukong@ece.neu.edu

Y. Fu
Department of Electrical and Computer Engineering and College of Computer and Information Science (Affiliated), Northeastern University, 360 Huntington Avenue, Boston, MA 02115, USA
e-mail: yunfu@ece.neu.edu

Y. Fu (ed.), *Human Activity Recognition and Prediction*,
DOI 10.1007/978-3-319-27004-3_6

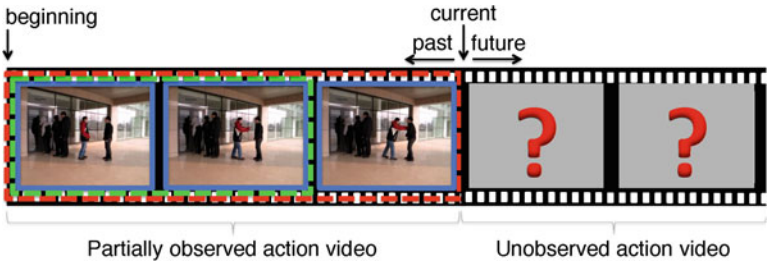

Fig. 6.1 Our method predicts action label given a partially observed video. Action dynamics are captured by both local templates (*solid rectangles*) and global templates (*dashed rectangles*)

effectively utilize history action information. The confidence of history observations is expected to increase since action data are progressively arriving in action prediction.

A novel multiple temporal scale support vector machine (MTSSVM) [9] is proposed in this chapter for early classification of unfinished actions. In MTSSVM, a human action is described at two different temporal granularities (Fig. 6.1). This allows us to learn the evolution and dynamics of actions, and make prediction from partially observed videos with temporally incomplete action executions. The sequential nature of human actions is considered at the fine granularity by local templates in the MTSSVM. The label consistency of temporal segments is enforced in order to maximize the discriminative information extracted from the segments. Note that the temporal orderings of inhomogeneous action segments is also captured by the temporal arrangement of these local templates in an implicit manner.

MTSSVM also capture history action information using coarse global templates. Different from local templates, the global templates characterize action evolutions at various temporal length, ranging from the beginning of the action video to the current frame. This global action information is effectively exploited in MTSSVM to differentiate between action categories. For instance, the key feature for differentiating action "push" from action "kick" is the motion "arm is up". Our model is learned for describing such increasing amount of information in order to capture featured motion evolution of each action class.

A new convex learning formulation is proposed in this chapter to consider the essence of the progressively arriving action data. The formulation is based on the structured SVM (SSVM), with new constraints being added. The label consistency is enforced between the full video and its containing temporal segments. This allows us to extract the discriminative information as much as possible for action prediction. Furthermore, a principled monotonic scoring function is modelled for the global templates. This scoring function enables us to utilize the fact that useful information is accumulating with the action data progressively arriving. We show that our new learning formulation can be efficiently solved using a standard SSVM solver. In addition, we demonstrate that the formulation essentially minimizes the upper bound of the empirical risk of the training data.

2 Related Work

Action Recognition A popular representation for human actions is called bag-of-words approach, which characterizes the actions by a set of quantized local spatiotemporal features [3, 16, 18, 27]. Bag-of-words approach can capture local motion characteristics and insensitive to background noise. Nevertheless, it does not build expressive representation when large appearance and pose variations occur in videos. Researchers address this problem by integrating classification models with human knowledge and representing complex human actions by semantic descriptions or attributes [7, 8, 11]. Other solutions such as learning actions from a set of key frames [13, 23] or from status images [25, 26] have also been studied as well. Nevertheless, a majority of current action recognition algorithms are expected to fully observe actions before making predictions. This assumption hinders these algorithms from the task that human actions must be predicted when only partial of the action videos is observed.

Human actions can also be modeled as temporal evolutions of appearance or pose. This line of approaches generally utilize sequential state models [12, 20, 21, 24] to capture such evolutions, where a video is treated as an ordered temporal segments. However, the relationship of temporal action evolution in reference to observation ratios is not considered in these approaches, making them improper for action prediction. In comparison, the progressive data arrival is simulated in our approach. Large scale temporal templates are used to model action evolutions from the first frame to the current observed one. Hence, unfinished actions at various observation ratios can be accurately recognized using our approach.

Action Prediction The goal of action prediction is to recognize unfinished action execution from partial videos. The integral bag-of-words (IBoW) and dynamic bag-of-words (DBoW) approaches were proposed in [15] for action prediction. These two approaches compute the mean of features in the same action category at the same progress level, and use the mean as the model for each progress level. Nevertheless, the constructed models are sensitive to outliers due to large intra-class appearance variations. This problem was overcome by [1], in which representative action models are built using the sparse coding technique. Results demonstrate that the proposed method achieves superior performance over the IBoW and DBoW approaches. All these method deal with short-duration action prediction problem, while long-duration problem was explored in [10]. One limitation of [10] is that the temporal segments are detected using motion velocity peaks that are very difficult to obtain in real-world outdoor datasets. Different from existing work [1, 10, 15], our prediction model integrates a crucial prior knowledge that the amount of useful information is accumulating with the arriving of new observations. This important prior information is not utilized in their methods. Furthermore, the proposed approach takes label consistency of segments into account, but it is not considered in their methods. Thanks to the label consistency, our approach is able to extract discriminative information in local segments and captures temporal ordering

information implicitly. In addition, our model captures action dynamics at multiple scales while [1, 15] only capture the dynamics at one single scale.

Besides action prediction, [4] investigated early event detection problem. Their method can localize the beginning and ending frames given an unfinished event video. Kitani et al. [6] studied the problem of activity forecasting. The approach is able to reason the optimal path for a person to go from location A to location B.

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 $x[1,T]$: $t \in \{1,\ldots,T\}$, 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 $l = 1,\ldots,K$ 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 kth 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. 6.2). 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.

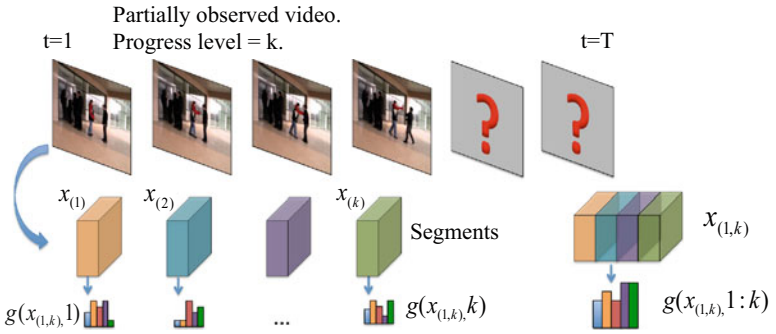

Fig. 6.2 Example of video segments $x_{(k)}$, partial video $x_{(1,k)}$, feature representations $g(x_{(1,k)}, l)$ of segments ($l = 1,\ldots,k$), and the representation of the partial video $g(x_{(1,k)}, 1:k)$

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 [14] 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 kth segment (Fig. 6.2). The representation of the lth ($l \in \{1, \ldots, k\}$) 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 lth segment.

3.2 Model Formulation

Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ be the training data, where x_i is the ith 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 $y \in \mathcal{Y}$ ($k \in \{1, \ldots, K\}$).

We formulate the action prediction problem using the structured learning as presented in [22]. 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 $k \in \{1, \ldots, K\}$ 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

$$y^* = \arg\max_{y \in \mathcal{Y}} F(x_{(1,k)}, y; \mathbf{w}^*) = \arg\max_{y \in \mathcal{Y}} \langle \mathbf{w}^*, \Phi(x_{(1,k)}, y) \rangle. \tag{6.1}$$

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

$$\mathbf{w}^{\mathrm{T}} \Phi(x_{(1,k)}, y) = \boldsymbol{\alpha}_k^{\mathrm{T}} \psi_1(x_{(1,k)}, y) + \sum_{l=1}^{K} \Big[\mathbf{1}(l \leqslant k) \cdot \boldsymbol{\beta}_l^{\mathrm{T}} \psi_2(x_{(1,k)}, y) \Big], \tag{6.2}$$

where $\mathbf{w} = \{\alpha_1, \ldots \alpha_K, \beta_1, \ldots, \beta_K\}$ 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. (6.2) are summarized as follows.

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

$$\boldsymbol{\alpha}_k^{\mathrm{T}}\psi_1(x_{(1,k)}, y) = \sum_{a\in\mathcal{Y}} \alpha_k^{\mathrm{T}}\mathbf{1}(y = a)g(x_{(1,k)}, 1:k). \tag{6.3}$$

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 $\{\alpha_1, \ldots, \alpha_K\}$. The selected template α_k is used to score the unfinished video $x_{(1,k)}$. Define $A = [\alpha_1, \ldots, \alpha_K]$ 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 [15], 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 [15].

Local progress model (LPM) $\mathbf{1}(l \leqslant k) \cdot \boldsymbol{\beta}_l^{\mathrm{T}}\psi_2(x_{(1,k)}, y)$ indicates how likely the action classes of all the temporal segments $x_{(l)}$ $(l = 1, \ldots, k)$ 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

$$\boldsymbol{\beta}_l^{\mathrm{T}}\psi_2(x_{(1,k)}, y) = \sum_{a\in\mathcal{Y}} \beta_l^{\mathrm{T}}\mathbf{1}(y = a)g(x_{(1,k)}, l), \tag{6.4}$$

where feature vector $g(x_{(1,k)}, l)$ of dimensionality D extracts features from the lth segment of the unfinished video $x_{(1,k)}$. β_l of size $D \times |\mathcal{Y}|$ is the weight matrix for the lth segment. We use the indicator function $\mathbf{1}(l \leqslant k)$ to select all the segment weight matrices, $\beta_1, \ldots, \beta_k$, 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 $B = [\beta_1, \ldots, \beta_K]$ 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 SSVM [5, 22]. The optimal model parameter $\mathbf{w}^*$ of MTSSVM in Eq. (6.1) is learned by solving the following convex problem given training data $\{x_i, y_i\}_{i=1}^N$:

$$\min \ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N}(\xi_{1i} + \xi_{2i} + \xi_{3i}) \tag{6.5}$$

$$\text{s.t. } \mathbf{w}^{\mathrm{T}}\Phi(x_{i(1,k)}, y_i) \geq \mathbf{w}^{\mathrm{T}}\Phi(x_{i(1,k)}, y) + K\delta(y, y_i) - \frac{\xi_{1i}}{u(k/K)}, \quad \forall i, \forall k, \forall y, \tag{6.6}$$

$$\boldsymbol{\alpha}_k^{\mathrm{T}}\psi_1(x_{i(1,k)}, y_i) \geq \boldsymbol{\alpha}_{k-1}^{\mathrm{T}}\psi_1(x_{i(1,k-1)}, y) + K\delta(y, y_i) - \frac{\xi_{2i}}{u(k/K)}, \quad \forall i, k = 2, \ldots, K, \forall y, \tag{6.7}$$

$$\boldsymbol{\beta}_k^{\mathrm{T}}\psi_2(x_{i(k)}, y_i) \geq \boldsymbol{\beta}_k^{\mathrm{T}}\psi_2(x_{i(k)}, y) + kK\delta(y, y_i) - \frac{\xi_{3i}}{u(1/K)}, \quad \forall i, \forall k, \forall y, \tag{6.8}$$

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 (6.6) 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.

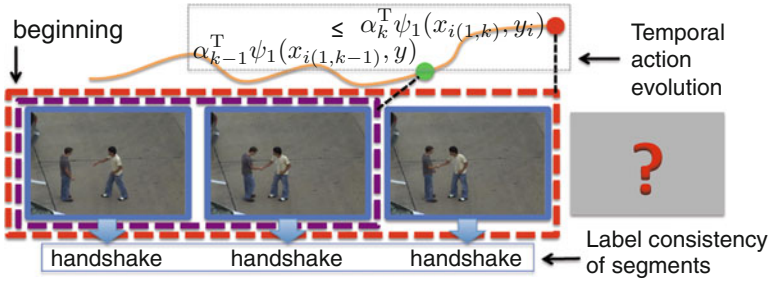

Fig. 6.3 Graphical illustration of the temporal action evolution over time and the label consistency of segments. *Blue solid rectangles* are LPMs, and *purple* and *red dashed rectangles* are GPMs

Constraint (6.7) considers **temporal action evolution** over time (Fig. 6.3). We assume that the score $\boldsymbol{\alpha}^{\mathrm{T}}\psi_1(x_{i(1,k)}, y_i)$ of the partial observation $x_{i(1,k)}$ at progress level k and ground truth label y_i must be greater than the score $\boldsymbol{\alpha}^{\mathrm{T}}\psi_1(x_{i(1,k-1)}, y)$ 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 (6.8) 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. 6.3). 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 $\Delta(y_i, y)$ 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 $\Delta(y_i, f(x_i))$. Usually, the performance of $f(\cdot)$ is given by the empirical risk $R_{\mathrm{emp}}(f) = \frac{1}{N}\sum_{i=1}^{N}\Delta(y_i, f(x_i))$ 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 $\Delta(y_i, y)u(\frac{k}{K})$. Here $\Delta(y_i, y)$ 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: $R_{\mathrm{emp}}(f) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\{\Delta(y_i, y)u(\frac{k}{K})\}$.

Denote by $\boldsymbol{\xi}_1^*$, $\boldsymbol{\xi}_2^*$ and $\boldsymbol{\xi}_3^*$ the optimal solutions of the slack variables in Eq. (6.5)–(6.8) 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. (6.5)–(6.8) minimizes the upper bound of the empirical risk $R_{\text{emp}}(f)$.

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, 15] 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, 10, 15] 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, 4, 10, 15] do not consider the label consistency. The consistency also implicitly models temporal segment context by enforcing the same label for segments while [1, 4, 15] 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, 10, 15].

3.5 Model Learning and Testing

Learning We solve the optimization problem (6.5)–(6.8) 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 (6.5)–(6.8) 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

$$U_i = \sum_{k=1}^{K} u\left(\frac{k}{K}\right) \max_{y} \left[K\delta(y, y_i) + \mathbf{w}^{\mathrm{T}}\Phi(x_{i(1,k)}, y) - \mathbf{w}^{\mathrm{T}}\Phi(x_{i(1,k)}, y_i)\right], \tag{6.9}$$

$$Z_i = \sum_{k=2}^{K} u\left(\frac{k}{K}\right) \max_y \left[K\delta(y, y_i) + \boldsymbol{\alpha}_{k-1}^{\mathrm{T}} \psi_1(x_{i(1,k-1)}, y) - \boldsymbol{\alpha}_k^{\mathrm{T}} \psi_1(x_{i(1,k)}, y_i)\right], \tag{6.10}$$

$$V_i = \sum_{k=1}^{K} u\left(\frac{1}{K}\right) \max_y \left[kK\delta(y, y_i) + \boldsymbol{\beta}_k^{\mathrm{T}} \psi_2(x_{i(k)}, y) - \boldsymbol{\beta}_k^{\mathrm{T}} \psi_2(x_{i(k)}, y_i)\right]. \tag{6.11}$$

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.

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 (6.8).

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) [17], 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 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 50 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 ten 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 [15], the MMED [4], the MSSC and the SC in [1], and the method in [13]. 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.

Figure 6.4a 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 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 chapter, which is to classify unfinished actions. We also compare with [13] on half and full video observations. Results in Table 6.1 show that our method achieves better performance over [13].

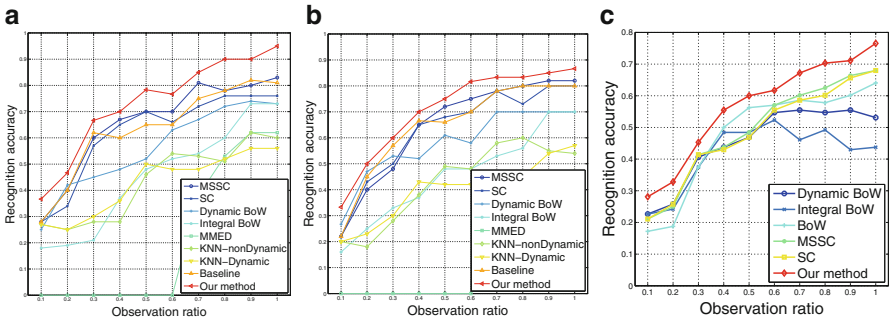

Fig. 6.4 Prediction results on the (**a**) UTI #1, (**b**) UTI #2, and (**c**) BIT datasets

Table 6.1 Prediction results compared with [13] on half and full videos

Observation ratio	Accuracy with half videos (%)	Accuracy with full videos (%)
Raptis and Sigal [13]	73.3	93.3
Our model	**78.33**	**95**

Comparison results on the UTI #2 datasets are shown in Fig. 6.4b. 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 GPM and the 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. Figure 6.5 shows the scores of the GPM and LPM ($\boldsymbol{\alpha}_k^{\mathrm{T}}\psi_1(x_{(1,k)}, y)$ of the GPM and $\sum_{l=1}^{K}\mathbf{1}(l \leqslant k)\cdot\boldsymbol{\beta}_l^{\mathrm{T}}\psi_2(x_{(1,k)}, y)$ 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. 6.5 (observation ratio from 0.3 to 0.7), adding more segment observations does not increase LPM's contribution to MTSSVM. Figure 6.6 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.

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. 6.4c demonstrate that MTSSVM outperforms MSSC and SC in all cases due to the effect of the GPM, 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 does 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;

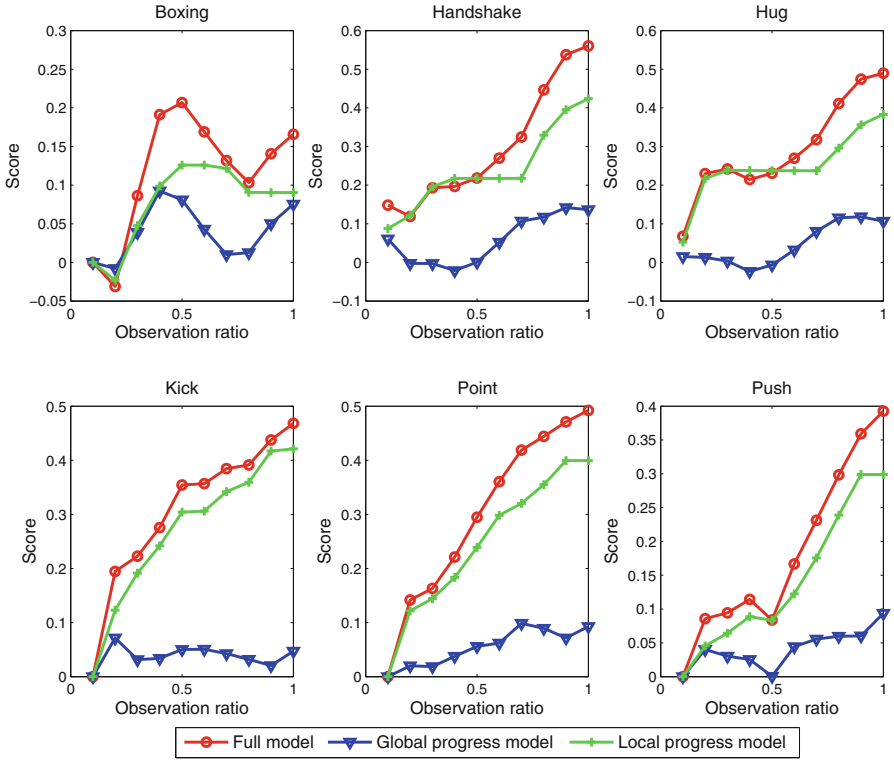

Fig. 6.5 Contributions of the global progress model and the local progress model to the prediction task

Fig. 6.6 Examples of segments in "handshake" and "hug". Segments $k = 6, 8, 10$ in the two classes are visually similar

especially, the segments in the second half of the videos, where people return to their starting positions (see Fig. 6.7). However, because MTSSVM models both the segments and the 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. (6.5). We set C to 0.5, 5, and 10, and test MTSSVM on all parameter

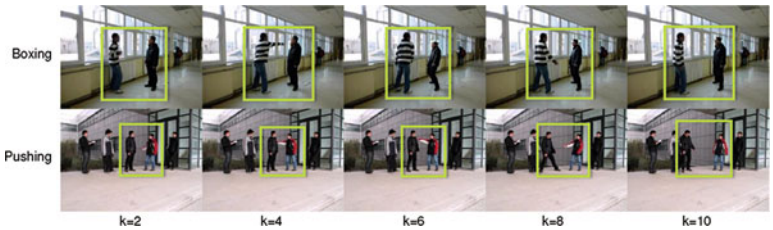

Fig. 6.7 Examples of visually similar segments in the "boxing" action (*Top*) and the "pushing" action (*Bottom*) with segment index $k \in \{2, 4, 6, 8, 10\}$. *Bounding boxes* indicate the interest regions of actions

Table 6.2 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

combinations with observation ratios 0.3, 0.5, and 0.8. Results in Table 6.2 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.

Finally, we also evaluate the importance of each component in the MTSSVM, including the Constraint (6.7), the Constraint (6.8), the local progress model [LPM in Eq. (6.4)], and the global progress model [GPM in Eq. (6.3)]. We remove each of these components from the MTSSVM, and obtain four variant models, the no-cons2 model [remove the Constraint (6.7) from MTSSVM], the no-cons3 model [remove the Constraint (6.8)], the no-LPM model [remove the LPM and Constraint (6.8)], and the no-GPM model [remove the GPM and Constraint (6.7)]. We compare MTSSVM with these variants with parameter C of 1 and 100. Results in Fig. 6.8 show that the GPM is the key component in the MTSSVM. 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 the 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 no-cons3 model and the no-LPM model, they cannot capture useful information for differentiating action classes.

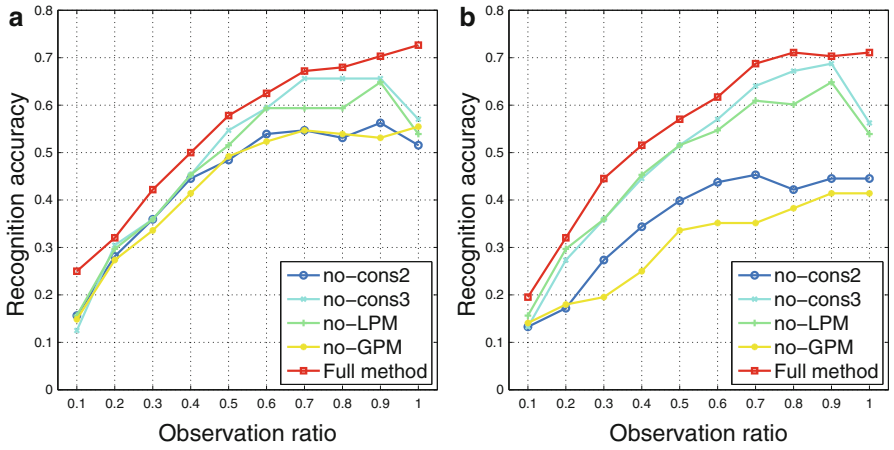

Fig. 6.8 Prediction results of each component in the full MTSSVM with C parameter (**a**) 1 and (**b**) 100

5 Summary

We have proposed the 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 SSVM 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.

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. Mach. Learn. **77**(1), 27–59 (2009)
6. Kitani, K.M., Ziebart, B.D., Andrew Bagnell, J., Martial Hebert, M.: Activity forecasting. In: ECCV (2012)
7. Kong, Y., Jia, Y., Fu, Y.: Learning human interaction by interactive phrases. In: ECCV (2012)
8. Kong, Y., Jia, Y., Fu, Y.: Interactive phrases: semantic descriptions for human interaction recognition. In: TPAMI (2014)

9. Kong, Y., Kit, D., Fu, Y.: A discriminative model with multiple temporal scales for action prediction. In: ECCV (2014)
10. Li, K., Hu, J., Fu, Y.: Modeling complex temporal composition of actionlets for activity prediction. In: ECCV (2012)
11. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: CVPR (2011)
12. Niebles, J.C., Chen, C.-W., Fei-Fei, L.: Modeling temporal structure of decomposable motion segments for activity classification. In: ECCV (2010)
13. Raptis, M., Sigal, L.: Poselet key-framing: a model for human activity recognition. In: CVPR (2013)
14. Raptis, M., Soatto, S.: Tracklet descriptors for action modeling and video analysis. In: ECCV (2010)
15. Ryoo, M.S.: Human activity prediction: early recognition of ongoing activities from streaming videos. In: ICCV (2011)
16. 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)
17. Ryoo, M., Aggarwal, J.: UT-interaction dataset, ICPR contest on semantic description of human activities. http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html (2010)
18. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: ICPR, vol. 3, pp. 32–36. IEEE, New York (2004)
19. Shapovalova, N., Vahdat, A., Cannons, K., Lan, T., Mori, G.: Similarity constrained latent support vector machine: an application to weakly supervised action classification. In: ECCV (2012)
20. Shi, Q., Cheng, L., Wang, L., Smola, A.: Human action segmentation and recognition using discriminative semi-Markov models. Int. J. Comput. Vis. **93**, 22–32 (2011)
21. Tang, K., Fei-Fei, L., Koller, D.: Learning latent temporal structure for complex event detection. In: CVPR (2012)
22. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. J. Mach. Learn. Res. **6**, 1453–1484 (2005)
23. 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)
24. Wang, Z., Wang, J., Xiao, J., Lin, K-.H., Huang, T.S.: Substructural and boundary modeling for continuous action recognition. In: CVPR (2012)
25. Yao, B., Fei-Fei, L.: Action recognition with exemplar based 2.5d graph matching. In: ECCV (2012)
26. Yao, B., Fei-Fei, L.: Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. IEEE Trans. Pattern Anal. Mach. Intell. **34**(9), 1691–1703 (2012)
27. Yu, T.-H., Kim, T.-K., Cipolla, R.: Real-time action recognition by spatiotemporal semantic and structural forests. In: BMVC (2010)