Online segmentation and classification of modeled actions performed in the context of unmodeled ones

Dimitrios I. Kosmopoulos[1,3]
dkosmo@ics.forth.gr

Konstantinos Papoutsakis[1,2]
papoutsa@ics.forth.gr

Antonis A. Argyros[1,2]
argyros@ics.forth.gr

[1]Institute of Computer Science
FORTH, Greece

[2]Computer Science Department
University of Crete, Greece

[3]Dept. of Informatics Engineering
Technological Educational Institute
Crete, Greece

Abstract

In this work, we provide a discriminative framework for online simultaneous segmentation and classification of visual actions, which deals effectively with unknown sequences that may interrupt the known sequential patterns. To this end we employ Hough transform to vote in a 3D space for the begin point, the end point and the label of the segmented part of the input stream. An SVM is used to model each class and to suggest putative labeled segments on the timeline. To identify the most plausible segments among the putative ones we apply a dynamic programming algorithm, which maximises an objective function for label assignment in linear time. The performance of our method is evaluated on synthetic as well as on real data (Weizmann and Berkeley multimodal human action database). The proposed approach is of comparable accuracy to the state of the art for online stream segmentation and classification and performs considerably better in the presence of previously unseen actions.

1 Introduction

In this paper we deal with the problem of online segmentation of visually observable actions, i.e., we have to provide labels given the fact that the visual observations arrive stream-wise on a sequential fashion and we need to decide on the label shortly after they are received, without having available the full sequence.

The video segmentation has been traditionally treated separately from the classification step, however, these two problems are correlated and can be better handled considering simultaneously the low level cues and the high level models representing the candidate classes (see e.g., [23], [10]). Following that observation, generative models can build probabilistic models of actions and can give the posterior of assigning labels to observations. In case that an unknown activity appears, the posterior probability given the known classes will be low, so that it will easily signify a sequence of unknown observations. However, generative models rely on simplifying statistical assumptions for computing the joint probability

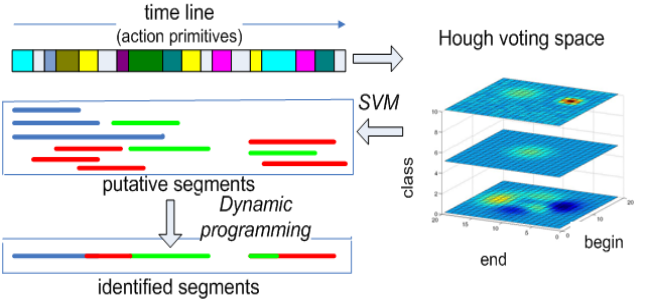


Figure 1: Overview of the proposed method: The action primitives in the considered time span vote in a 3D Hough voting space (begin-end-class). The SVM receives the votes and suggests the putative segments. The segments that maximise an objective function compose the final solution, which is found via dynamic programming.

of the states and the observed features, whereas a more general discriminative model may better predict the conditional probability of the states given the observed features. As a result, several researchers have investigated the use of discriminative models of actions such as Conditional Random Fields [17], Support Vector Machines [10], [25] or random forests [27], [6]. However, the discriminative models are not without problems, since they cannot easily handle untrained actions, since they were not part of their optimisation process.

In this work we seek to mitigate the aforementioned limitation of the discriminative methods, by employing a discriminative Hough transform. By collecting the votes generated by action primitives we detect putative segments, i.e., the time span as well as the action type associated to each of them using an SVM. In the following step we use the putative segments to assign labels to time instances, so that the observations are best explained; to this end we employ a dynamic programming algorithm. Figure 1 gives an overview of the method.

More specifically, the innovations of the proposed approach are: (a) A generic voting scheme in 3D space, which is defined by the start point, the end point and the class-specific label in order to segment the observation stream in an online fashion; (b) a generalised Hough transform to classify and segment time series online in a way that is decoupled from the observations. (c) a method to deal with unknown sequential patterns. (e) a discriminative framework for vote weighting in the aforementioned 3D voting space and (f) a dynamic programming method for label assignment in linear time.

The rest of the paper is organised as follows: In the next section we survey the related work. In section 3 we describe the proposed framework, which includes the generation of hypotheses via voting and the evaluation via dynamic programming. Section 4 describes the experimental results and section 5 concludes this work.

2 Related work

The simultaneous segmentation and classification of visual or other time series has gained in popularity recently. *Generative models* have been used extensively. In [5] a Bayesian nonparametric approach is presented for speaker diarisation that builds on the hierarchical Dirichlet process hidden Markov model. Typical approaches that exploit the hierarchical structure of time series to classify actions, are the hierarchical HMMs [4] or the layered

hidden Markov model [20]. The semi-Markov model, which explicitly captures the duration has also been employed [8], [19].

Dynamic time warping and its variations are also popular, e.g., [22] use a feature weighting variation for gesture recognition. [1] proposes a framework for gesture recognition and spatiotemporal gesture segmentation using a stochastic dynamic time warping combined with a variation of the Viterbi algorithm. Another line of research is followed by methods that seek to exploit the *co-occurrence of tasks* see, e.g., [29], [11]. Our method does not currently exploit this information, but this depends solely on how we treat the overlapping tasks that we recognise. In this paper we deal exclusively with the simpler problem of non-overlapping tasks. Recently a great deal of work was done on *deep learning*, e.g., convolutional neural networks [12], [14] and restricted Boltzman machines [9]. These methods can create a feature mapping in an unsupervised way and then they apply standard classification methods. Our method is agnostic to the employed features and could use these results.

In [2] a *discriminative framework* was proposed. The sequences were assigned to classes and segmented into subsequences using conditional random fields. The method requires the full sequence in advance and cannot operate in an online fashion. Similarly, conditional random fields were used in [24], [21]. In [17] hierarchical layers of latent variables were used to model sub-structures within actions. In [23] a discriminative approach was introduced under a semi-Markov model framework, and a Viterbi-like algorithm was devised to efficiently solve the induced optimisation problem. The segments that gave the best score were the selected ones. In [10] a joint segmentation and classification scheme was presented and it sought to maximise the confidence of the segment assignment. To this end a multi-class SVM was used and a dynamic programming approach was followed for efficient seeking of candidate segments and their evaluation. In [25] latent labels and state durations were optimised in a maximum margin approach. The results were very promising, but the authors made the assumption that the video sequences contain only instances of classes that were previously learned. These schemes have problems if segments belonging to previously unseen classes appear between the known ones because the dynamic programming scheme becomes inapplicable. A possible solution could be to model the content that does not belong to any of the known categories as a separate class, however that approach would not handle properly the unknown sequences that might appear.

Of some relevance to our method is the *anomaly detection* in time series. In contrast to segmentation of time series, anomaly detection is the identification of unknown patterns, i.e., behaviours that deviate from normal given the previously seen data. [15] used the one-class SVM discriminative model to detect novel observations and outliers after transforming the time series data to a vector, which is the required input to the SVM. Such approaches can be used offline, where the whole sequence is known. [13] proposes an on-line (causal) novelty detection method capable of detecting both outliers and regime change points in sequential time-series data using a metric based on extreme value theory. This method is more related to change point detection methods used (when a signal changes), e.g., in EEG analysis rather than to classification of more complex patterns like actions or gestures. Our approach is different from the above, since we do not care about the detection of abnormal sequences; our primary goal is to segment online (eventually with a short delay) some known sequential patterns, which could be occasionally interrupted by unknown or uninteresting sequences.

Related to our approach is the *Hough transform*. In [16] a discriminative Hough transform was used for object detection, where each local part cast a weighted vote for the possible locations of the object center. It was shown that the weights can be learned in a max-margin framework, which directly optimises the classification performance. Its resilience to noise

and the fact that multiple objects can be present simultaneously make the Hough transform a very attractive option, which can be generalised to time series and therefore to gesture and action recognition. We propose a discriminative Hough transform for time series analysis, where motion primitives are used instead of local descriptors. We deal with concurrent segmentation and classification in time-series instead of object detection in images where the voting is different. We vote in a 3D space which is defined by the time span and type of segment (begin point, end point and class label) and then use dynamic programming for the final label assignment. Another interesting approach for the problem of action recognition using Hough was presented in [27], where the action segmentation was coupled to the action positioning problem for a single actor. By considering features such as optical flow intensity and position, a Hough forest was built and then used to cast votes in real scenarios. Compared to that work we decouple the position estimation problem from the classification and segmentation problem, which reduces the dimensionality of the voting space. In [27] the actor was represented by a rectangle, while it is not clear how such a coupled framework would generalise for more complex problems involving high dimensional models (e.g., multiple actors, skeleton models, region descriptors). Hough transform was also used in [28]. However, it used voting for pose estimation unlike our work, which uses voting for actions.

3 Proposed framework

3.1 Hypotheses generation via discriminative voting

In the discriminative voting framework we seek to identify simultaneously (a) the instances of classes C of sub-sequences in time series data, (b) the location $\mathbf{x}$ of the class-specific subsequence, in other words the begin and the end time point in the data.

Let $\mathbf{f}_t$ denote the feature vector observed at time instance t and let $S(C,\mathbf{x})$ denote the score of class C at a location $\mathbf{x}$ (the $(C,\mathbf{x})$ is a cell in a 3D voting space). The implicit model framework obtains the overall score $S(C,\mathbf{x})$ by adding up the individual probabilities $p(C,\mathbf{x},\mathbf{f}_t,l_t)$ over all observations within a time window (l_t indicates if observations in time t belong to the currently examined sliding window), i.e.,:

$$S(C,\mathbf{x}) = \sum_t p(C,\mathbf{x},\mathbf{f}_t,l_t) = \sum_t p(\mathbf{f}_t,l_t)p(C,\mathbf{x}|\mathbf{f}_t,l_t) \tag{1}$$

We define M action primitives, which result from clustering of the visual observation vectors $\mathbf{f}_t$. Let P_i denote the i-th action primitive. By assuming a uniform prior over features and time locations and marginalizing over the primitive entries we get:

$$S(C,\mathbf{x}) = \sum_t p(C,\mathbf{x}|\mathbf{f}_t,l_t) = \sum_{i,t} p(P_i|\mathbf{f}_t,l_t)p(C,\mathbf{x}|P_i,\mathbf{f}_t,l_t) \tag{2}$$

By observing that the primitives P_i depend only on the observed features $\mathbf{f}_t$ and not on the time location l_t that they appear, we can simplify $p(P_i|\mathbf{f}_t,l_t)$ to $p(P_i|\mathbf{f}_t)$. Similarly, $p(C,\mathbf{x}|P_i,\mathbf{f}_t,l_t)$ depends only on the matched primitive P_i and l_t and simplifies to $p(C,\mathbf{x}|P_i,l_t)$. Therefore we can write:

$$S(C,\mathbf{x}) = \sum_{i,t} p(P_i|\mathbf{f}_t)p(C,\mathbf{x}|P_i,l_t) = \sum_{i,t} p(P_i|\mathbf{f}_t)p(\mathbf{x}|C,P_i,l_t)p(C|P_i,l_t) \tag{3}$$

The term $p(P_i|\mathbf{f}_t)$ can be calculated by applying Bayes rule assuming uniform distribution for $\mathbf{f}_t$: $p(P_i|\mathbf{f}_t) \propto p(\mathbf{f}_t|P_i)/p(P_i)$. We use Gaussian mixture models (GMM) to represent the distributions of the observation vectors, and to express one primitive by one component of the GMM. The numerator can be simply obtained by evaluating the respective component of the GMM, while the denominator is given by the associated prior.

Returning to (3) the term $p(\mathbf{x}|C,P_i,l_t)$ gives the temporal distribution in the locations $\mathbf{x}$ given the class C and the location l_t of the primitive P_i. This can be modelled from the training samples. The third term is the weight of the primitive P_i emphasizing how confident we are that the primitive P_i at time t matches the class C as opposed to another class. If we assume that the $p(C|P_i,l_t)$ is invariant to the location l_t of the primitive we can simplify the term to $p(C|P_i,l_t) = p(C|P_i) \propto p(P_i|C)/p(P_i)$. The weight is defined independently for each primitive, so we refer to it as the naive-Bayes weights.

Our voting framework can be the basis for a discriminative voting scheme, for time series data. It is inspired by the framework presented in [16], which dealt with object detection. We can use maximum margin optimisation if we observe that the score $S(C,\mathbf{x})$ is a linear function of $p(C|P_i)$. By simplifying $p(C|P_i,l_t) = p(C|P_i)$ and by considering Eq.(3) we have:

$$\begin{aligned} S(C,x) = \sum_{i,t} p(P_i|\mathbf{f}_t)p(\mathbf{x}|C,P_i,l_t)p(C|P_i) = \sum_i p(C|P_i)\sum_t p(P_i|\mathbf{f}_t)p(\mathbf{x}|C,P_i,l_t) \\ = \sum_i w_i \times a_i(\mathbf{x}) = W_c^T A(\mathbf{x}) \end{aligned} \tag{4}$$

where $A^T = [a_1 a_2 ... a_M]$ (hereafter mentioned as the activation vector), and a_i is given by:

$$a_i(\mathbf{x}) = \sum_t p(\mathbf{x}|C,P_i,l_t)p(P_i|\mathbf{f}_t) \tag{5}$$

The weights W_c^T are class-specific and we notice that they can be optimised in a discriminative fashion to maximise the score for correct segmentations and labels. The discriminative nature of the optimisation may give much better results compared to the voting scheme based only on the estimation of $p(C|P_i)$. For a given training sequence that is observed we set the respective class-specific labels in the respective locations $\mathbf{x}_i$, i.e., at the bins that correspond to the correct begin/end points. The rest of the locations are defined to belong to an "idle" class. In other words, we define the ground truth labels $S(C,\mathbf{x}_i)$ for all possible locations $\mathbf{x}_i$ within a time window. For each of the locations $\mathbf{x}_i$ we need to find the activation vectors $A(\mathbf{x}_i)$, which are calculated by using Eq.(5). Given the labels $S(C,\mathbf{x}_i)$ and the respective $A(\mathbf{x}_i)$ we calculate the weights W_c using multiple one-versus-all binary SVM settings.

In *testing* we vote in the 3D space using Eq.(4) and then we apply the SVMs in a sliding time window to get the putative segments, considering only the segments that collected enough votes. As may happen in many cases, the local maxima in the Hough parameter space may be the result of noise and thus may not correspond to a real segment. Therefore an additional evaluation step is normally applied to eliminate some false positives using an HHM-like likelihood function. An illustrative example of the proposed hypotheses generation process and the additional evaluation step is shown in Fig.3(b)-(c) in the context of the proposed algorithmic steps, presented in Fig.1.

Generally we cannot exclude the possibility of previously unseen observations or patterns. In dynamic scenes it is almost certain that at some point we will come across some observations that will not be explainable by the existing models. Assigning a specific class

label to represent all the possible unknown classes is not the best solution, since the related model has to be really complex to cover the variety of possible observations and most importantly these observations are not known in advance.

3.2 Hypotheses evaluation via dynamic programming

The processing described in the previous section results into, say, K putative segments; K is many orders of magnitude smaller than the number resulting from brute force by considering all possible combinations of classes and begin-end points. However, these K segments are typically overlapping and belong to different classes. Their possible combinations are $O(K!)$ which can still be high and may not be evaluated exhaustively, even by using evolutionary algorithms as we noticed. We also noticed that in most of the cases, the discriminative framework proposes, among others, segments that are close to the ground truth. That fact motivated an approach that seeks to consider *only the proposed segments*, so that they can best explain the sequence of observations on the timeline. If for parts of the time line there are no proposed segments these parts remain unassigned and account for unknown observations.

We merge the proposed segments that overlap and have the same label, but typically there are also overlaps between segments of different labels, which compete for the same time windows. Assuming only one label for each time slot, we propose a variation of the Viterbi algorithm for linear-cost label assignment with regard to the number of input frames.

We define the likelihood δ_t, which is calculated after the optimal assignment of time instances to classes. The optimal sequence of classes for a time segment t=1..T, which contains overlapping candidate segments of different labels is given by the path $\psi_t = C_1, C_2, ..., C_t$. The initialisation of the likelihood δ_t for t=1 is then given by:

$$\delta_1(C_1) = \sum_{i=1}^{M} p(\mathbf{f}_1|P_i) \cdot p_b(P_i|C_1) \tag{6}$$

At time t, which accounts for the first t time instances we have:

$$\delta_t(C_t) = \max_{C_{t-1}}\{\delta_{t-1}(C_{t-1}) \cdot A(C_{t-1}, C_t)\} \cdot \sum_{i=1}^{M} p(\mathbf{f}_t|P_i) p(P_i|C_t) \tag{7}$$

where

$$A(C_{t-1}, C_t) = \begin{cases} \sum_{i=1}^{M}\sum_{j=1}^{M} p_{tr}(P_{i,t-1}, P_{j,t}|C_{t-1}) p(\mathbf{f}_{t-1}|P_{i,t-1}, C_{t-1}) p(\mathbf{f}_t|P_{j,t}, C_t) & if \quad C_t = C_{t-1} \\ \sum_{i=1}^{M}\sum_{j=1}^{M} p_e(P_i|C_{t-1}) p(\mathbf{f}_{t-1}|P_{i,t-1}, C_{t-1}) p_b(P_{j,t}|C_t) p(\mathbf{f}_t|P_{j,t}, C_t) & if \quad C_t \neq C_{t-1} \end{cases} \tag{8}$$

The first branch of Eq.(8) assumes no switching between different labels from t-1 to t. Therefore the transition likelihood $p_{tr}(P_{i,t-1}, P_{j,t}|C_{t-1})$ is involved, while the class remains the same (C_{t-1}). In time t-1 and t the primitives P_i and P_j are observed respectively, as indicated by the subscripts. There is no hard assignment of primitives, so all transitions are evaluated and weighted by the respective observation likelihoods.

The second branch accounts for switching between different labels, so the termination likelihood p_e is involved for label C_{t-1}. A new segment with different label begins in t, so the begin likelihood p_b for the new class label C_t is involved. These are weighted by the

respective observation likelihoods given the classes C_t and C_{t-1} and the respective primitives. Again all possible combinations are evaluated.

In Eq.(7) only the right factor needs to be evaluated for each class, while the left part was calculated in the t-1 step. The overall cost of the calculation is linear and is applicable for online applications. To retrieve the sequence of assignments we keep track of the argument that maximised (7) through the array ψ_t, which is given by:

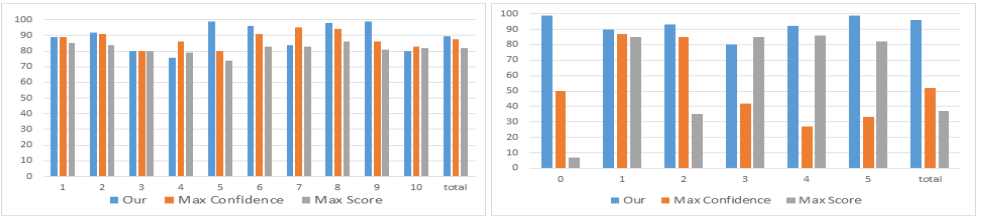
$$\psi_t(C_t) = \underset{C_{t-1}=1..L}{\arg\max}\{\, \delta_{t-1}(C_{t-1}) \cdot A(C_{t-1}, C_t)\} \tag{9}$$

An illustrative example of the hypotheses evaluation process is shown in Fig.3(d) for the two experimental configurations that are provided, resulting the final classification and segmentation action segments. The proposed dynamic programming algorithm differs from the typical Viterbi algorithm because the transition between labels on the time line has to be treated differently in the case that the same segment continues (label remains the same - first case in Eq.(8)) than the case that a new segment begins (label changes - second case in Eq.(8)).

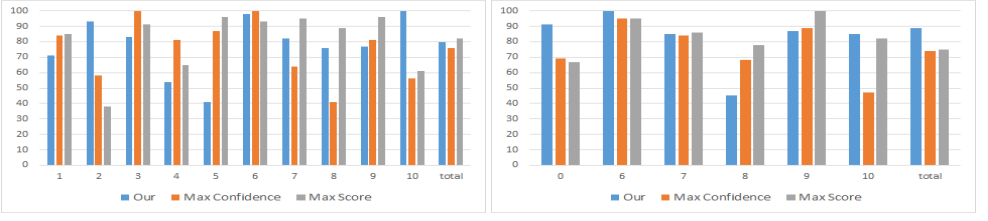
4 Experimental results

To verify the validity of our method we have experimented with synthetic as well as a real dataset from the field of visual action recognition. For our comparisons we implemented (using the CVX [8]) the optimisation scheme that maximises the confidence for segments similarly to [10] (hereafter denoted as MaxConfidence or MC) and the scheme that maximises the overall score similarly to [23] (hereafter denoted as MaxScore or MS); these are state of the art methods that do online segmentation, like our method does.

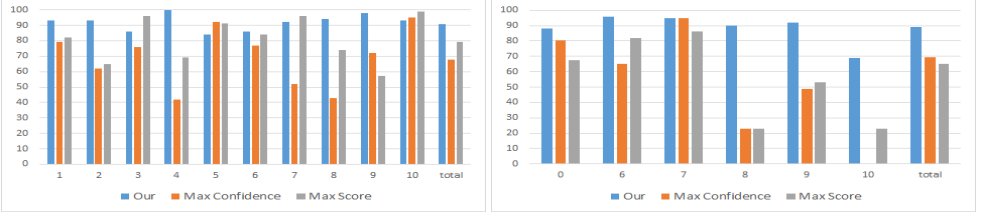
Synthetic data We generated a dataset of 2D sequences. We created randomly ten HMM models, each composed of up to three different states, by random definition of means, covariances, priors and state transitions. Then for each of them we performed sampling and we produced 100 sequences of length between 450 and 750 each. These sequences were concatenated at random order to form bigger sequences consisting of one instance per class. Given the dataset we investigated two different settings. To make our method comparable to existing work, we initially made the assumption of a multiclass problem, where all the knowledge was given in advance to the system, i.e., no instances stemming from unknown latent classes appeared. This implies that a label from the known set of labels had to be assigned to every frame. We used 50% of the sequences for training and the rest for testing. Fig.2a presents per class classification accuracies of our method, MC and MS, on a frame-by-frame basis. The size of the sliding window was defined by the maximum action length. Our method shows similar or higher accuracy, i.e., 89.44% vs. 87.65% and 81.71% of MC and MS respectively. Our method is agnostic about the existence of instances stemming from unknown sequential patterns; therefore small gaps falsely assigned to novel observations may appear. This is the largest source of error, i.e., segments that have actually larger duration are detected as shorter because the longer ones are sometimes not suggested as putative segments. On the contrary, MC and MS assume that all observations are known and that there are no gaps between the putative segments. Next, we assumed that some of the observed actions resulted from unknown latent classes. MC and MS are not able to classify instances of previously untrained actions; therefore, to make a fair comparison we trained an HMM for each known action and we checked the likelihood of each segmented sub-sequence using the



(a) Synthetic dataset



(b) MHAD dataset



(c) Weizmann dataset

Figure 2: Classification accuracy (%) per class for the three experiments when training with the full dataset (left) and with a subset (right). Class 0 denotes the novel observations.

respective model. For low likelihoods we classified the actions as unknown. We excluded the instances of five classes from training and we learned the rest. Our method gave promising results, exhibiting accuracy 96.16% (assignment of instances from 6-10 to 0 were considered true). The best results for MC and MS were 52.42% and 37.34% respectively, and were obtained by using a threshold of -10^{-7} for length-normalised log likelihood and M=30. Their inferior performace can probably be attributed to the requirement for continuous labeling of the whole timeline, which inevitably couples together the actions and necessitates a suboptimal postprocessing step to detect unseen actions; in contrast, our method is optimised to classify known and detect unknown actions without that constraint.

Berkeley dataset. The next experiment is related to visual recognition of actions, which involve the whole human body. To this end we use the Berkeley Multimodal Human Action Database (MHAD) [26], which consists of temporally synchronised and geometrically calibrated data. The dataset contains about 660 action sequences which correspond to about 82 minutes. In the original dataset the different actions were provided as segments. For the purpose of identifying actions in continuous data we concatenated those videos. We did not consider the action *sit down/stand up* as a separate action, but as the composition of the actions *sit down* and *stand up*; this apporach is justified by the continuous recognition that we do. Therefore we actually classified only ten different actions. All available 3D skeleton

joints were used to build our representation of human motion for each frame. We used the 3D orientation of each joint with respect to its ancestor in the kinematic skeletal chain based on quaternions. In addition, the 3D coordinates of each skeletal joint with respect to the hips joint were computed for each mocap record. Finally, the distance of the hips joint to the ground plane was also incorporated to the feature set for each frame. The last two subsets of our feature representation were normalised for each actor of the dataset, given the total length of the skeletal chain. The feature vector had 220 dimensions 103 representing all the joint angles and 117 representing all the joint positions. We trained with the first 7 subjects and tested with the last 5 ones as in [26]. Each subject repeated the same actions 4 times. Fig.2b gives the per class accuracy. Our method gave overall accuracy 79.58% compared to 76.22% of MC and 82.46% of MS when training with all classes. Essentially all methods gave comparative results. For reference purposes, we mention that the best results reported in [26] using the same skeletal data was 79.93% using use kernel-SVM with the χ^2 kernel for classification. However, the setting was different, i.e., unlike ours, the classification was performed on segmented videos and the *sit down/stand up* action was treated separately. We then examined the effect of unknown sequences, by using the same postprocessing step as in the synthetic experiment to make the competing methods comparable to ours. We trained with classes 6-10 and then tested using the same data as in the previous experiment. Illustrative experimental results are demonstrated in Fig.3. Our method had an overall accuracy of 89.04%, outperforming MC with 74.25% and MS with 75.14% (threshold -10^{-5}, M=40), which verifies the merit of the proposed method. Moreover, the supplemental material accompanying the paper [1] demonstrates results based videos compiled based on the Berkeley dataset, in the same experimental configuration as described in Fig.3.

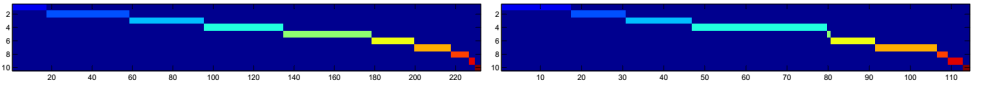
Weizmann dataset Finally we used the classification database of the Weizmann dataset [7], consisting of 9 actors performing 10 different actions. Based on the aligned binary foreground masks that are also available in the dataset, we computed the Zernike moments up to order 20 [18]. We exploited only the even polynomials (excluding zero-based polynomials) and concatenated the resulting amplitude and phase values resulting in feature vectors of size 220. In the following, we applied a GMM-based clustering to build a dictionary of M=30 motion primitives that represent the notion of spatiotemporal keywords in our framework. As in [10] we concatenated each set of videos of a subject into a single longer sequence; consequently, we composed 9 long videos of contiguous actions and used 5 of the videos for training and 4 for testing. The results per class are in Fig.2c. We used threshold= -10^{-5}. Training with all data yielded 90.82%, 67.64% and 79,31% for our method, MC and MS respectively; partial training with classes 1-5 yielded 89,23%, 69.34% and 65.14% for the three methods. That confirms the results from the previous experiments. For reference purposes we mention that in [10] 94% accuracy was reported for the MC, which is justified by the use of spatiotemporal features, while in our experiments we used only spatial features.

5 Conclusion

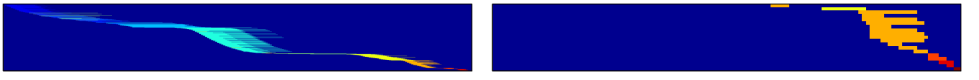
We presented a framework for simultaneous segmentation and classification of sequential data interrupted by unknown actions and we have applied it on synthetic and visual action streams. In all cases our method performed similarly or better than the competing discriminative methods when a "closed world" assumption was true. When the actions of interest

[1] http://youtu.be/LxIiTFDavpg

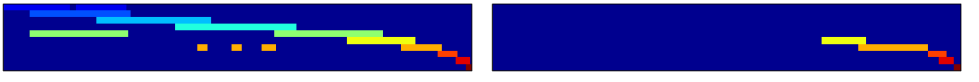
were interrupted by previously unseen actions our method was still able to classify them and detect the unknown ones. MaxScore and MaxConfidence gave inferior performance mainly due to the fact that they enforce continuity of labeling and the inevitable post-processing proved to be relatively ineffective. To our knowledge our discriminative method is the first one for online simultaneous segmentation and classification having this property.



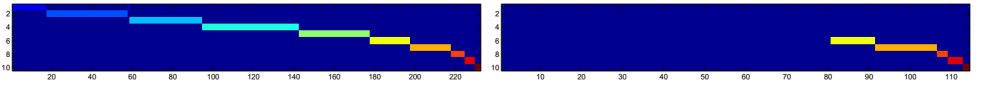
(a) Ground truth action segments



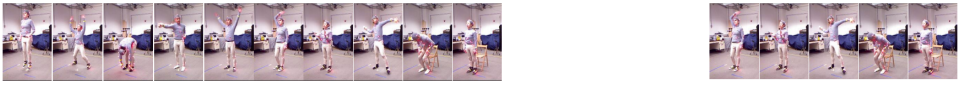
(b) Putative action segments based on hypotheses generation (prior to HMM-like likelihood evaluation).



(c) Putative action segments and their classes based on hypotheses generation.



(d) Final segmentation and classification of actions based on hypotheses evaluation.



(e) Key-frames of the recognised actions are presented in temporal sequence, for each of the test videos.

Figure 3: Sample results from the Berkeley dataset. Actions $1-10$ are illustrated as color-coded segments in figures (a) - (d). Horizontal axis represents the time-line in frames. On the left column, modeling and training of the proposed method is applied using all actions $1-10$, performed by subjects $1-7$. Testing is applied on an image sequence which contains each action once performed by subjects $8-12$. The final result is illustrated in the left sub-figure of (d), matching the ground truth in (a). On the right column, modeling and training using actions $6-10$ for subjects $8-12$ was performed. A test sequence was compiled concatenating all available actions once, from subjects $1-7$. The final result, shown in right sub-figure of (d), demonstrates the segmentation and classification of the modeled actions in the context of the unmodeled actions $1-5$, which are considered as unknown in that case, thus no recognition results are present during the first five actions. In (e), corresponding key-frames of the recognised actions are illustrated. (The figure is best viewed in color.)

Acknowledgement This research has received funding from the EU 7th Framework Programme (FP7/2007-2013) under grant agreement No. 288146, HOBBIT.

References

[1] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1685–1699, Sept 2009.

[2] S. P. Chatzis, D. Kosmopoulos, and P. Doliotis. A conditional random field-based model for joint sequence segmentation and classification. *Pattern Recognition*, 46: 1569–1578, 2013.

[3] T.V. Duong, H.H. Bui, D.Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 838–845 vol. 1, 2005.

[4] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Mach. Learn.*, 32(1):41–62, July 1998. ISSN 0885-6125.

[5] E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. A Sticky HDP-HMM with Application to Speaker Diarization. *Annals of Applied Statistics*, 5(2A):1020–1056, 2011.

[6] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33 (11):2188–2202, November 2011. ISSN 0162-8828.

[7] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[8] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. 2008.

[9] Ge. E. Hinton, S. Osindero, and Y-W Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. ISSN 0899-7667.

[10] M. Hoai, Z.Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3265–3272, 2011.

[11] H. Izadinia and M. Shah. Recognizing complex events using large margin joint low-level event model. In *Proceedings of the 12th European conference on Computer Vision - Volume Part IV*, ECCV'12, pages 430–444, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-33764-2.

[12] Sh. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1): 221–231, 2013. ISSN 0162-8828.

[13] H. Joo Lee and S. J. Roberts. On-line novelty detection using the kalman filter and extreme value theory. In *ICPR*, pages 1–4. IEEE, 2008. ISBN 978-1-4244-2175-6.

[14] Q. Le, J. Ngiam, Z. Chen, D. Chia, P. Koh, and A. Ng. Tiled convolutional neural networks. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *NIPS*, pages 1279–1287. 2010.

[15] J. Ma and S. Perkins. Time-series novelty detection using one-class support vector machines. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, pages 1741–1745 vol.3, July 2003.

[16] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1038–1045. IEEE, 2009. ISBN 978-1-4244-3992-8.

[17] L. Morency, A. Quattoni, and T. Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[18] R. Mukundan and K. R. Ramakrishnan. *Moment Functions in Image Analysis: Theory and Applications*. World Scientific, New York, 1998.

[19] P. Natarajan and R. Nevatia. Coupled hidden semi markov models for activity recognition. In *Motion and Video Computing, 2007. WMVC '07. IEEE Workshop on*, pages 10–10, 2007.

[20] N. Oliver, A. Garg, and E. Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *Comput. Vis. Image Underst.*, 96(2): 163–180, November 2004. ISSN 1077-3142.

[21] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29 (10):1848–1852, 2007. ISSN 0162-8828.

[22] M. Reyes, G. Dominguez, and S. Escalera. Feature weighting in dynamic time warping for gesture recognition in depth data. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1182–1188, Nov 2011.

[23] Q. Shi, Li Wang, Li Cheng, and A. Smola. Discriminative human action segmentation and recognition using semi-markov model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[24] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Conditional models for contextual human motion recognition. *Comput. Vis. Image Underst.*, 104(2):210–220, 2006. ISSN 1077-3142.

[25] K. Tang, Li Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1250–1257, 2012.

[26] R. Vidal, R. Bajcsy, F. Ofli, R. Chaudhry, and G. Kurillo. Berkeley mhad: A comprehensive multimodal human action database. In *Proceedings of the 2013 IEEE Workshop on Applications of Computer Vision (WACV)*, WACV '13, pages 53–60, Washington, DC, USA, 2013. IEEE Computer Society. ISBN 978-1-4673-5053-2.

[27] A. Yao, J. Gall, and L. Van Gool. A hough transform-based voting framework for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2061 –2068, june 2010.

[28] T-H. Yu, T-K. Kim, and R. Cipolla. Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3642–3649, June 2013.

[29] Y. Zhu, N.M. Nayak, and A.K. Roy-Chowdhury. Context-aware activity recognition and anomaly detection in video. *Selected Topics in Signal Processing, IEEE Journal of*, 7(1):91–101, 2013.