Temporal Context Analysis for Action Recognition in Multi-agent Scenarios

Yifan Zhang[1], Chunjie Zhang[2], Zhiqiang Tang[1], and Hanqing Lu[1]

[1] National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{yfzhang,hqlu}@nlpr.ia.ac.cn, zqtang2013@gmail.com
[2] School of Computer and Control Engineering,
University of Chinese Academy of Sciences, Beijing 100049, China
cjzhang@jdl.ac.cn

Abstract. In multi-agent scenarios such as sports videos, multiple actions are played by different players. Such actions do not necessary appear strictly sequentially but can happen in parallel. Approaches which only consider a single stream of actions are not competent to handle such scenarios. The temporal and causal relationships between the action streams such as “concurrence”, “mutually exclusion” and “triggering” need to be captured so as to correctly recognize the actions. In this paper, a novel method is presented for action recognition in multi-agent scenarios leveraged by analyzing the relationships among the temporal contextual actions. The multi-streams of actions are modeled by a Dynamic Baysian Network (DBN) containing several temporal processes corresponding to each type of action. Comparing to the Coupled Hidden Markov Model (CHMM), only the necessary interlinks between the temporal processes are built by a structure learning algorithm to capture the salient relationships. Empirical results on real-world video data demonstrate the effectiveness of our proposed method.

Keywords: Multi-agent action recognition, graphical model, structure learning.

1 Introduction

Recognizing human actions from videos is a task of obvious scientific and practical importance. In this paper we consider the problem of recognizing actions in multi-agent scenarios, which is challenging due to that the interactions between actions would lead to large state spaces and complicate the already uncertain low-level visual processing. In multi-agent scenarios, group or interactive actions from multiple players may occur sequentially or in parallel. It is not uncommon to have primitive actions with parallel streams. One action does not need to be completed before continuing on to the later one. For example, in the case of basketball, “the offensive player is *shooting* while the defender is *blocking* him” (shown in Fig. 1), which includes parallel actions. However, most of the existed

B. Huet et al. (Eds.): PCM 2013, LNCS 8294, pp. 760–768, 2013.

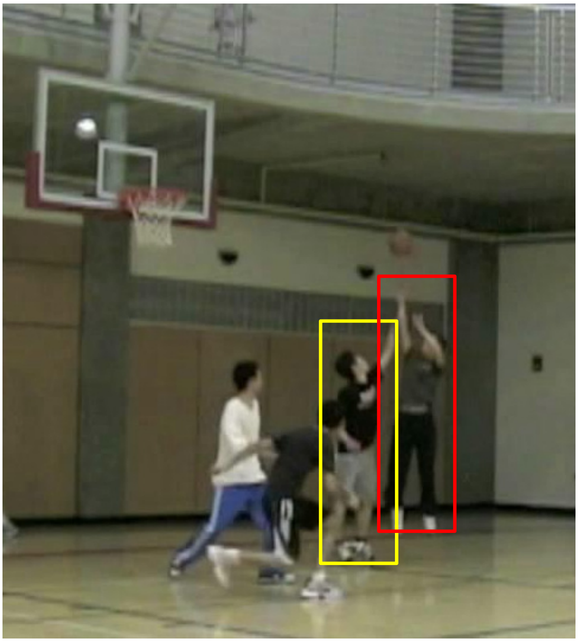

Fig. 1. Interactive actions in multi-agent scenario. The action *shooting* and *blocking* are highly likely to co-occur together. The frame is captured from the OSUPEL basketball dataset [2].

approaches such as Finite State Machines (FSMs) [5] or Hidden Markov Models (HMMs) [8] consider the scenario as a temporally ordered single stream of actions.

In multi-agent scenarios such as sports videos, one action could maintain certain relationships with the contextual ones which are governed by the domain knowledge and certain rules of thumb. These relationships may impose a temporal structure on constituent primitives, which can be leveraged to correctly recognize the actions. Besides the "concurrence" relationship between the parallel actions described above, there are two other important relationships: "mutually exclusion" and "triggering". If two actions are mutually exclusive, it means that their concurrence is inhibited. For example, "if one player is *dribbling*, he cannot be *holding* the ball at the same time". The "triggering" relationship indicates that one action is caused by another, such as "one player *passing* the ball leads to the other one *catching* the ball".

To capture these kinds of relationships, we propose to use a Dynamic Baysian Network (DBN) to represent the temporal structure on top of the actions. A DBN is a directed acyclic graphical model, which models the temporal evolution of a set of random variables X over time. In the DBN, each variable corresponds to one type of action, and its evolution forms a temporal process. To model the interactions between the actions, instead of being fully connected between the temporal processes, the interlinks are discovered by a structure learning algorithm so as to capture the salient temporal relationships while still controlling the complexity of the network.

Our main contribution is providing a system that efficiently and robustly recognizes actions in multi-agent scenarios by using (1) temporal context analysis for capturing meaningful temporal and causal relationships to disambiguate amongst noisy visual observations and (2) structure learning for discovering salient interlinks in the DBN model instead of fully connected and make it computationally tractable.

2 Related Work

There has been considerable research exploring how to represent and model multiple action interactions. The typical representations are Finite State Machines (FSMs) [5] or Hidden Markov Models (HMMs) [8], in which actions cause state transitions in a strictly sequential order and a successful transition through the stream implies the recognition of the action. Unfortunately, these type of approaches may not handle the scenarios with multiple streams of actions. A main difficulty with these approaches are that the system can only be in one state at a time, they cannot well represent parallel actions. Coupled Hidden Markov Model (CHMM) [1,7] was presented by Brand et al., which factorize the actions into two parallel transition processes to deal with the complexity in highly coupled T'ai Chi hand movements. However, the transition model structure of the CHMM is fully connected. Direct extensions of this model to multi-agent scenarios which contains multiple transition precesses may encounter computationally tractability issues.

Expert domain knowledge can be leveraged to create models for multi-agent actions. Ryoo and Aggarwal [9] model two person interactions by a context-free grammar (CFG), where high-level interactions are defined hierarchically using logical spatial and temporal predicates on sub-actions. The CFG parsing is not probabilistic and can be sensitive to low-level failures. Brendel et al. proposed Probabilistic event logic (PEL) [2] to capture temporal relationship between actions based on confidence weighted formulas. However, the temporal relationships must be known in advance in order to manually encode them into the event logic formulas.

Based on the limitations of the approaches mentioned above, we need to find a method which can automatically discover interactive relationships so as to model multiple streams of actions. In addition, it should be a probabilistic model which can handle the uncertainty brought by the low-level visual processing.

3 Relationships Modeling Using a DBN

We propose to use a DBN to model and learn the relationships among the actions. Let X^t represent a set of random variables at a discrete time slice t. A DBN is defined as $B = (G, \Theta)$, where G is the model structure, and Θ represents the model parameters, i.e., the conditional probabilistic distributions for all nodes. A DBN is a time-sliced model, as shown in Fig. 2(a), where each time slice is used to represent the snapshot of certain evolving temporal processes at a time instant. The neighboring time slices are interconnected by the links joining certain temporal nodes from two consecutive slices. Hence, to specify a DBN, we need to define the intra-slice topology (within a slice), the inter-slice topology (between two slices), as well as the parameters for the first two slices. Such a two-slice temporal Bayesian network is often called a 2TBN, as shown in Fig. 2(b). A 2TBN is used as a template which can be unrolled to the whole DBN.

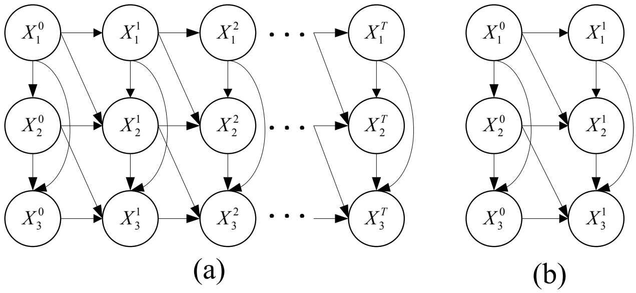

Fig. 2. (a) The DBN for $T + 1$ time slice, and (b) the corresponding 2TBN representation

Given a DBN model, the joint probability over all variables $X^0, ..., X^T$ is computed as follows:

$$P(x^0, ..., x^T) = P(x^0) \prod_{t=0}^{T-1} P(x^{t+1}|x^t), \tag{1}$$

where x^t represents the sets of values taken by the random variables X at time t, $P(x^0)$ captures the joint probability of all variables in the first slice, and $P(x^{t+1}|x^t)$ represents the transition probability.

3.1 Structure Initialization

The DBN model learning is started by structure initialization. An initial 2TBN structure is derived by analyzing the relationships among temporal contextual actions in the training data. Based our definition, there are mainly three types of relationships for actions at two adjacent time slices: concurrence, mutually exclusion and triggering. Concurrence and mutually exclusion can be captured in the intra-slice topology, while triggering can be captured in the inter-slice topology.

In the training data, the "concurrence" relationship is discovered by the statistic of the concurrences between each pair of actions. The pairwise concurrence dependency between two actions is computed as follows:

$$P(A_i^t = 1 | A_j^t = 1) = \frac{N_{A_i^t + A_j^t}}{N_{A_j^t}}, \tag{2}$$

where $N_{A_i + A_j}$ is the total number of concurrences of action A_i and A_j in the same time slice, and N_{A_j} is the total number of occurrences of A_j in the whole database.

The "mutually exclusion" relationship is captured by the statistic of the absence of one action given the presence of the other one. The pairwise negative dependency between two actions is computed as follows:

$$P(A_i^t = 0 | A_j^t = 1) = \frac{N_{\neg A_i^t + A_j^t}}{N_{A_j^t}}, \tag{3}$$

where $N_{\neg A_i^t + A_j^t}$ is the total number of absences of A_i^t given the presence of A_j^t, and $N_{A_j^t}$ is the total number of occurrences of A_j^t in the whole database.

The "triggering" relationship indicates that one action's presence at the current time slice is caused by another one's presence at the previous time slice. We obtain the statistic of such situation in the training data. The pairwise causal dependency between two actions is computed as follows:

$$P(A_i^t = 1 | A_i^{t-1} = 0, A_j^{t-1} = 1) = \frac{N_{A_i^t + \neg A_i^{t-1} + A_j^{t-1}}}{N_{\neg A_i^{t-1} + A_j^{t-1}}}, \tag{4}$$

where $N_{A_i^t + \neg A_i^{t-1} + A_j^{t-1}}$ is the total number of the presences of A_i at the current time slice given its absence and the presence of A_j at the previous time slice, and $N_{\neg A_i^{t-1} + A_j^{t-1}}$ is the total number of the absence of A_i and the presence of A_j at the same time slice in the database.

We have obtained the three types of pairwise dependency for each pair of actions by temporal contextual analysis. The first two types of dependency are represented within a time slice. If they are higher than predefined thresholds, we assume that the two actions have strong dependency and build a link between the two nodes in the intra-slice topology. The last type of dependency is represented between two adjacent time slices. Similarly, if it is higher than a predefined threshold, we build a link between the two nodes in the inter-slice topology. This way, an initial 2TBN structure has been constructed.

3.2 Model Learning

After analyzing the contextual relationships, we obtain an initial DBN structure. Although it is our best guess based on the contextual analysis, it is necessary to use training data to refine it with a structure learning algorithm. The structure learning algorithm first defines a score that describes the fitness of each possible structure G to the observed data, and then, the best fitted network structure is identified with the highest score. The fitness score is defined as:

$$Score(G) = logP(D, G) = logP(G) + logP(D|G), \tag{5}$$

where $logP(G)$ is the log prior probability of the DBN structure. We do not give an equal prior to all possible structures. Instead, we assign a higher probability to the prior structure we initialized. $logP(D|G)$ is the log likelihood of the training data which can be approximated by the Bayesian information criterion (BIC) as follows:

$$logP(D|G) \approx logP(D|G, \hat{\Theta}) - \frac{d}{2}log(K), \tag{6}$$

where $\hat{\Theta}$ is the set of parameters of G which maximizes the likelihood of the training data D, d is the number of free parameters in G, and K is the number of training data D. The first term is used to measure how well the model fits the data, and the second term is a penalty term to punish the structure complexity. To obtain the model parameters $\hat{\Theta}$, we maximize the posterior distribution $p(\Theta|D, G)$ (MAP), given the training data D and the current structure G:

$$p(\Theta|D, G) = \prod_{i=1}^{n}\prod_{j=1}^{m_i} p(\theta_{ij}|D, G), \tag{7}$$

where n is the number of variables in G, and m_i is the number of all the parent instantiations for variable X_i. Since the training data set is complete, each parameter θ_{ij} can be calculated by a counting process. After we define the fitness score of the model, we can use an iterative way to learn the structure and parameters of the DBN. We firstly start with the initial DBN structure G_0, learn the parameters based on G_0 and compute the fitness score. Secondly, we generate the nearest neighbors of G_0 by adding, deleting, or reversing a single link, subjecting to the acyclicity constraint. Then we update G_0 with the structure which has the maximum score among the neighbors. The iteration process will be terminated until the score converges or the maximum iteration time is reached.

3.3 DBN Inference

Once the DBN model has been learned, each node is attached with an observation node so as to form a two-layer model. The top layer encodes the actions and their temporal and causal relationships. The bottom layer comprises a set of observation nodes that ingest the preliminary detection from low-level features. During action recognition, the node in the top layer are hidden and must be inferred from the observations in the bottom layer. The inference is conducted by finding the most probable explanation (MPE) of the evidence.

Let $A^t_{1:n}$ represents all the nodes for actions at time t, where n is the number of action nodes. Given the available evidence until time t: $O^{1:t}_{A_{1:n}}$, the action nodes are inferred over time by maximizing the probability $p(A^t_{1:n}|O^{1:t}_{A_{1:n}})$. The probability can be factorized by performing the DBN updating process as described in [6], and the inference can be solved by the widely used junction tree inference algorithm.

By DBN inference, we can employ the contextual information and mutual dependencies in a holistic way to refine the isolated preliminary detection results, and thus correctly recognize the actions in the multiple action streams.

4 Experimental Results

To evaluate the performance of our action recognition methods in multi-agent scenario, we choose the OSUPEL basketball dataset [2]. The OSUPEL basketball dataset is publicly available and it consists of multiple players playing against each other in a real basketball court. There are six primitive actions in the dataset: Pass, Catch, Hold Ball, Shoot, Jump and Dribble. This dataset is suitable for evaluating localization and recognition of multiple primitive actions characterized by rich spatiotemporal constraints.

Before discussing the recognition results in the multiple action streams, we first briefly describe the method on how to get the visual evidence of the actions from low-level features. The computed tracks of the players in the videos have been already provided in the dataset. We extract features from the bounding box of the computed tracks and use an HMM to detect each action separately. The features are derived from the histogram of oriented gradients (HOG)[3] and histogram of oriented optical flow (HOF) [4]. We train HMMs for each action class and use them to detect actions in the video separately. Please note that the preliminary detection results are not satisfying. The performance of the noisy detectors on the OSUPEL basketball data are summarized in Table 1. They can only be considered as noisy evidences to infer the true occurrence of each action.

Table 1. Preliminary action detection performance

	Dribble	Jump	Shoot	Pass	Catch	Hold
Recall	0.52	0.33	0.20	0.25	0.24	0.49
Precision	0.86	0.63	0.43	0.74	0.67	0.64

We feed the preliminary detection results as the observation values into the bottom layer of the DBN model so as to infer the state of the hidden nodes in the top layer. Fig. 3 shows the inference results on an example sequence with 10000 frames in the the OSUPEL basketball dataset. It contains 6 plots corresponding to 6 action classes. In each plot, the ground truth of the action occurrences is shown in the first row; the inference results are shown in the second row; the observations from the low-level detectors are shown in the third row. It is clear that using the holistic inference based on the contextual information, our model can correct the missing errors and false alarms in the preliminary detection results, and thus improve the action recognition accuracy.

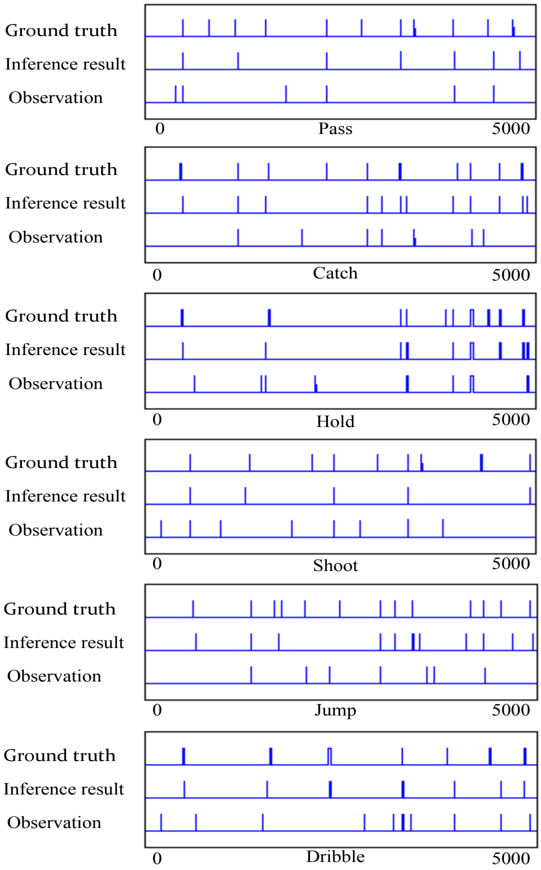

Fig. 3. Comparison of inference results with ground truth and observations from low-level detector on an example sequence in the OSUPEL basketball dataset

The performance of our model on the whole dataset is demonstrated in Table 2. Comparing to Table 1, we can find that both the precision and recall have been improved. To compare with our method, we also implement a CHMM [1,7] model, the structure of which is fully connected between all the temporal processes. The performance of CHMM model on action recognition is also shown in Table 2. Since the CHMM model cannot learn an adequate transition model, it performs worse than our model as expected. This demonstrates that the structure learning reduces the number of unnecessary parameters and caters for better network structure discovery.

5 Conclusions

We have proposed a DBN model for action recognition in multi-agent scenarios. Three kinds of mutual relationships between the interactive actions, concurrence,

Table 2. Inference performance using CHMM and our method

		Dribble	Jump	Shoot	Pass	Catch	Hold
CHMM	Recall	0.53	0.32	0.23	0.29	0.36	0.53
	Precision	0.83	0.61	0.44	0.72	0.69	0.66
Our method	Recall	0.54	0.36	0.27	0.32	0.34	0.56
	Precision	0.86	0.60	0.44	0.73	0.70	0.67

mutual exclusion and triggering, can be successfully captured in our model, so as to compensate the limitation of the low-level visual detectors. An advanced structure learning algorithm has been presented to discover meaningful and salient dependencies in order to construct a computationally tractable network. Currently, our model is time slice based. In the following work, we intend to extent our model to temporal interval based to make it more expressive and be able to capture more complex relationships.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant No. 61202325, the President Fund of UCAS, the Open Project Program of the National Laboratory of Pattern Recognition, China Postdoctoral Science Foundation:2012M520434, 2013T60156.

References

1. Brand, M., Oliver, N., Pentland, A.: Coupled hidden markov models for complex action recognition. In: CVPR Conference Proceedings (1997)
2. Brendel, W., Fern, A., Todorovic, S.: Probabilistic event logic for interval-based event recognition. In: CVPR Conference Proceedings (2011)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR Conference Proceedings, pp. 886–893 (2005)
4. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)
5. Hongeng, S., Nevatia, R.: Multi-agent event recognition. In: ICCV Conference Proceedings, pp. 84–91 (2001)
6. Korb, K., Nicholson, A.: Bayesian Artificial Intelligence. Chapman and Hall/CRC (2004)
7. Kratz, L., Nishino, K.: Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In: CVPR Conference Proceedings (2009)
8. Lv, F., Nevatia, R.: Recognition and segmentation of 3-D human action using HMM and multi-class adaBoost. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 359–372. Springer, Heidelberg (2006)
9. Ryoo, M.S., Aggarwal, J.K.: Recognition of composite human activities through context-free grammar based representation. In: CVPR Conference Proceedings (2006)