Unsupervised Co-Activity Detection from Multiple Videos Using Absorbing Markov Chain

Donghun Yeo, Bohyung Han, Joon Hee Han
Department of Computer Science and Engineering, POSTECH, Korea

Abstract

We propose a simple but effective unsupervised learning algorithm to detect a common activity (co-activity) from a set of videos, which is formulated using absorbing Markov chain in a principled way. In our algorithm, a complete multipartite graph is first constructed, where vertices correspond to subsequences extracted from videos using a temporal sliding window and edges connect between the vertices originated from different videos; the weight of an edge is proportional to the similarity between the features of two end vertices. Then, we extend the graph structure by adding edges between temporally overlapped subsequences in a video to handle variable-length co-activities using temporal locality, and create an absorbing vertex connected from all other nodes. The proposed algorithm identifies a subset of subsequences as co-activity by estimating absorption time in the constructed graph efficiently. The great advantage of our algorithm lies in the properties that it can handle more than two videos naturally and identify multiple instances of a co-activity with variable lengths in a video. Our algorithm is evaluated intensively in a challenging dataset and illustrates outstanding performance quantitatively and qualitatively.

Introduction

Activity detection refers to a technique to identify and localize one or more predefined classes of activities from input videos in temporal (sometimes, spatio-temporal) domain. Various algorithms have been proposed so far (Chen and Grauman 2012; Yuan, Liu, and Wu 2011; Duchenne et al. 2009; Tian, Sukthankar, and Shah 2013), but they are typically based on supervised learning techniques that utilize clean training data. On the other hand, co-activity detection, which is the problem of our interest, is a task to extract one or more streaks of frames containing a common activity from each video out of multiple ones without separate training procedure. Co-activity detection is a notoriously challenging problem; this is partly because co-activity is not a well-defined term and feature descriptors are not sufficiently discriminative, especially in an unsupervised setting. However, this problem has potential to be used in various computer vision applications such as automatic video cropping, video summarization, video annotation, visual surveillance, etc.

Co-activity detection has not been studied intensively yet, and there are only a few closely related works, which typically have critical limitations in scalability. (Chu, Zhou, and De la Torre 2012) formulates co-activity detection problem as an identification of the most similar subsequences given a pair of videos using branch-and-bound. Guo *et al*. (Guo et al. 2013) finds common activity through binary labeling on Markov random field, which is constructed based on dense trajectories. However, these algorithms are limited to processing only a pair of videos at a time and difficult to be extended for more videos. (Chu, Song, and Jaimes 2015) proposes a video co-summarization technique, which can be applied to co-activity detection by finding a complete bipartite subgraphs defined on a video pair. This algorithm is extended to multiple videos by aggregating pairwise results, but it does not consider multiple videos directly. A weakly supervised algorithm for co-activity detection is proposed in (Duchenne et al. 2009), but it assumes that co-activity has the same length and occurs once in each video. More importantly, it requires negative examples to detect co-activities. Another weakly supervised method is based on simple biclustering (Xiong and Corso 2012), which can also be applied to more than two videos, but this is not fully automatic since human agent should determine which cluster corresponds to co-activity in the final stage. There are several related problems in computer vision and pattern recognition, *i.e*., image co-segmentation (Joulin, Bach, and Ponce 2010; Kim and Xing 2012), video co-segmentation (Chiu and Fritz 2013), motif discovery (Minnen et al. 2007). However, these approaches are not flexible enough to be extended or converted to solve co-activity detection problem.

We propose an unsupervised algorithm to extract a subset of frames containing co-activity from each video using absorbing Markov chain. In our algorithm, we divide each video into multiple overlapping subsequences with a fixed length and construct a vertex corresponding to each subsequence. Edges are created for every pair of vertices that do not belong to the same video and also for temporally overlapped subsequences in each video. Note that the intra-sequence edges facilitate transitions between temporally adjacent vertices, which may be parts of co-activities due to temporal locality. This property is also useful to identify co-activity with variable lengths. In addition to this basic graph structure, we add an absorbing vertex that is connected from all other nodes in the graph. The weight of each inter-sequence edge is given by the similarity between two end vertices while the weights of intra-sequence edges and absorbing edges are defined differently, which will be discussed in detail later. Once the graph construction for an absorbing Markov chain is completed, we compute the absorption time of the Markov chain that starts from each vertex using fundamental matrix (Seneta 2006). In our formulation, the vertices that take more time to be absorbed correspond to the subsequences in co-activity and the identification of such vertices provides the final solution of co-activity detection.

The contributions of our co-activity detection algorithm are summarized as follows:

- We formulate a novel co-activity detection problem using absorbing Markov chain in a principled way. We claim that the use of absorbing Markov chain for this problem is conceptually reasonable and technically sound.
- Our algorithm can handle more than two videos naturally, which overcomes the common drawback of existing methods (Chu, Zhou, and De la Torre 2012; Guo et al. 2013; Chu, Song, and Jaimes 2015). It identifies multiple instances of a co-activity with variable lengths in a video without any modification of the original formulation.
- One can reproduce our results easily since our algorithm hardly involve various heuristics and require parameter tuning. We constructed a new dataset designated to co-activity detection problem, and our algorithm illustrates outstanding performance on challenging datasets according to our intensive evaluation.

Background

This section reviews absorbing Markov chain briefly and its properties employed in our algorithm.

Absorbing Markov Chain

Markov chain is a stochastic process that undergoes transitions from one state v_i to another state v_j with a transition probability obeying the Markov property as

$$\begin{aligned} &P(x_{T+1} = v_j | x_T = v_i, x_{T-1} = v_{i_{T-1}}, \ldots, x_1 = v_{i_1}) \\ &= P(x_{T+1} = v_j | x_T = v_i) \\ &= p_{ij}, \end{aligned} \quad (1)$$

where x_T denotes a random variable on a state space, $V = \{v_1, v_2, \cdots, v_M\}$. In other words, the state at time step $T+1$ depends only on the state at time step T, not any other preceding states, which results in transition probability from v_i to v_j denoted by p_{ij}. A transition matrix $\mathbf{P} \in \mathbb{R}^{M \times M}$ has an entry p_{ij} at its i-th row and j-th column.

Absorbing Markov chain is a Markov chain that has at least one absorbing state in the state space, which is the special state whose transition probabilities to other states are all zeros; formally, v_i is an absorbing state if $p_{ii} = 1$ and $p_{ij} = 0, \forall i \neq j$, and non-absorbing states are referred to as transient states. A random walker is supposed to reach one of absorbing states in the end and cannot escape from the absorbing state by its definition. Absorbing Markov chain has been studied in several computer vision problems, which include image matching (Cho, Lee, and Lee 2010), image segmentation (He, Xu, and Chen 2012) and saliency detection (Jiang et al. 2013).

Absorption Time

In absorbing Markov chain, we can compute the *absorption time*, the expected number of steps of a random walk, starting from a transient state v_i, before arriving at any absorbing state. This expected number of steps is absorption time.

To compute the absorption time of an absorbing Markov chain with M_t transient states and M_a absorbing states, we first re-enumerate the states to make the transition matrix have a canonical form as

$$\mathbf{P} = \begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \quad (2)$$

where $\mathbf{Q} \in \mathbb{R}^{M_t \times M_t}$ represents the transition probability matrix for all pairs of transient states and $\mathbf{R} \in \mathbb{R}^{M_t \times M_a}$ contains transition probabilities from transient states to absorbing states. The transition probabilities from absorbing states to transient states are given by an $M_a \times M_t$ zero matrix $\mathbf{0}$, and $\mathbf{I}$ is an identity matrix that represents transition probabilities between absorbing states.

Let q_{ij}^T be the probability of transition from v_i to v_j in T steps, and $\mathbf{Q}^T(i,j) = q_{ij}^T$. Then, the expected number of occurrences that a random walk starting from v_i visits v_j before arriving at one of absorbing states is given by the summation of q_{ij}^T, $T \in [0, \infty)$, which is estimated efficiently by the following equation:

$$\mathbf{F} = \sum_{T=0}^{\infty} \mathbf{Q}^T = (\mathbf{I} - \mathbf{Q})^{-1}, \quad (3)$$

where the output matrix $\mathbf{F}$ is referred to as fundamental matrix. The absorption time of a random walk starting from v_i in absorbing Markov chain, denoted by y_i, is obtained by adding the number of occurrences visiting transient states before absorption as

$$y_i = \sum_{j=1}^{M_t} f_{ij}, \quad (4)$$

where $\mathbf{F}(i,j) = f_{ij}$.

Co-activity Detection using Absorbing Markov Chain

Our algorithm identifies a co-activity from multiple videos by simply computing the absorption time in an absorbing Markov chain. This framework provides a reasonable solution to alleviate the drawbacks of existing co-activity detection algorithms and has potential to improve performance significantly. This section describes how absorbing Markov chain is related to co-activity detection and how we construct the graph to model the problem.

Problem Formulation

In the proposed framework, we construct a directed graph to model semantic relationship between the subsequences belonging to different videos and impose temporal locality between the temporally adjacent subsequences within a video. The graph has the transient states, which correspond to the subsequences, while there is an additional absorbing state that is connected from all transient states.

The subsequences corresponding to co-activity should have common characteristics in their representations. We assign a large transition probability between the transient states with similar representations if they belong to different videos. In addition, the transition probability of an intra-sequence edge from a state to another is proportional to the sum of its outgoing inter-sequence edge weights. This strategy is useful to identify a co-activity with variable lengths by exploiting temporal locality. The transition probability from a transient state to the absorbing state is supposed to depend on the weights of inter-sequence edges originated from the transient state; roughly speaking, the transition probability tends to be negatively correlated to the inter-sequence edge weight. If an absorbing Markov chain follows these definitions of transition probabilities, the transient states corresponding to a co-activity are likely to have large absorption times while a random walk initiated from other transient states tends to arrive at the absorbing state quickly.

Graph Construction

Denote a set of N videos by $\mathscr{S} = \{S_1, S_2, \cdots, S_N\}$, where S_n $(n = 1, \ldots, N)$ is a set of fixed-length subsequences in each video that are partially overlapped in temporal domain. Then, we construct a directed graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ to identify the subsequences corresponding to co-activity based on absorbing Markov chain as follows.

Vertices There are two kinds of nodes in the graph; one is transient node and the other is absorbing node. Transient nodes denoted by $\mathsf{U} \subset \mathsf{V}$ are the union of all disjoint sets S_n, and contain all subsequences of all videos in $\mathscr{S}$ as

$$\mathsf{U} = \bigcup_{n=1}^{N} S_n = \{s_1, \ldots, s_{M_t}\}, \tag{5}$$

where M_t is the number of transient nodes and each element in S_n has one-to-one mapping to one of the elements in U. There is a single absorbing node denoted by a and the overall vertex set of the graph G is given by $\mathsf{V} = \mathsf{U} \cup \{a\}$.

Edges We define three categories of edges for convenience, where two categories are pertaining to the edges between transient nodes—inter-sequence and intra-sequence edges—and one is about the edges between transient nodes and the absorbing node.

The inter-sequence edges are created between all pairs of transient nodes that correspond to the subsequences belonging to different videos. Formally, a directed edge $e^{\text{inter}} \in \mathsf{E}_1$ connects from $s_i \in S_{n_1}$ to $s_j \in S_{n_2}, \forall (i,j), \forall (n_1, n_2)$, where $n_1 \neq n_2$, and a directed edge $e^{\text{intra}} \in \mathsf{E}_2$ connects from $s_i \in S_n$ and $s_j \in S_n, \forall (i,j), \forall n$. We add an edge $e^{\text{abs}} \in \mathsf{E}_3$ from every transient state $s_i \in \mathsf{U}$ to the absorbing state a. There is no self-loop and the overall edge set of the graph G is given by $\mathsf{E} = \mathsf{E}_1 \cup \mathsf{E}_2 \cup \mathsf{E}_3$.

The weight of each edge is proportional to transition probability; larger weight between vertices encourages more transitions from the origin node and the destination node. The weight of an edge between transient nodes is given by

$$w_{ij} = \begin{cases} \psi(s_i, s_j) & s_i \in S_{n_1}, s_j \in S_{n_2}, n_1 \neq n_2 \\ \sum_{s_k \in \Omega_i} \psi(s_i, s_k) & s_i, s_j \in S_n,\ s_j \in \Lambda_i \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where $\psi(\cdot,\cdot)$ denotes the similarity measure between two subsequences, Ω_i indicates a set of inter-sequence neighbors of s_i, and Λ_i is a set of temporal neighbors of s_i in both directions. Note that s_i and s_j in the same sequence are temporal neighbors if they have any overlapped frames. The weight of intra-sequence edge does not depend on the similarity between subsequences but is given a value proportional to the sum of inter-sequence edge weights originated from the outgoing vertex. It can be interpreted as *indirect* similarity, which is useful to improve temporal localization of co-activities. We does not use direct similarity for intra-sequence weight since the similarity between adjacent subsequences are typically high due to temporal coherency, while we expect high weights are observed only between the subsequences belonging to the same co-activity.

On the other hand, the weight from a transient node to the absorbing node is given by

$$w_{iM} = \theta - \frac{1}{|\Omega_i|} \sum_{s_k \in \Omega_i} \psi(s_i, s_k), \tag{7}$$

where $M = M_t + 1$ and the constant θ is computed by

$$\theta = \max_j \left[\frac{1}{|\Omega_j|} \sum_{s_k \in \Omega_j} \psi(s_j, s_k) \right]. \tag{8}$$

Note that w_{iM} is always non-negative due to the definition of θ. Intuitively, the weight from a transient node s_i to the absorbing node a is given by the difference between the maximum of average inter-sequence edge weights and average inter-sequence edge weights originated from s_i.

The constructed graph structure based on a set of input videos is illustrated in Figure 1.

Identification of Co-activity Subsequence by Absorption Time

We now define a canonical transition matrix, $\mathbf{P} \in \mathbb{R}^{M \times M}$, of absorbing Markov chain based on the weights, which is given by

$$p_{ij} = w_{ij} \Big/ \sum_{k=1}^{M} w_{ik}, \tag{9}$$

where each row in $\mathbf{P}$ is normalized.

After defining the canonical transition matrix, we compute the absorption time of each subsequence using the fundamental matrix as in Eq. (4). The co-activity score of each

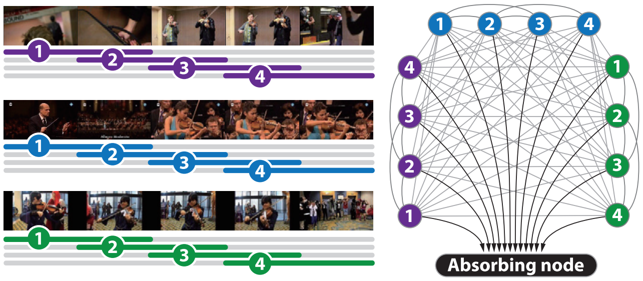


Figure 1: Visualization of the constructed graph for our absorbing Markov chain.

frame is computed by averaging the absorption times of subsequences containing the frame. The frames that have co-activity scores larger than a predefined threshold are identified as the co-activity frames. We cluster the frames in each video based on their absorption times using a Gaussian mixture model with two components, and determine the threshold by computing the optimal decision boundary in the estimated model through EM algorithm.

Implementation Details

We discuss crucial implementation issues in this subsection, which include feature descriptor, similarity measure between subsequences, and subsequence generation method.

Feature descriptor We employ the dense trajectory features, which are originally developed for activity recognition in (Wang et al. 2011), to represent subsequences. The dense trajectory features are composed of five histograms: histogram of trajectory shapes, histogram of HOGs, histogram of HOFs and two histograms of MBHs, which are denoted by $\phi_h(s_i)$ $(h = 1, \dots, 5)$ for subsequence s_i. All of these histograms are constructed based on bag-of-features model. Refer to (Wang et al. 2011) for more details about these features.

Similarity measure The similarity between s_i and s_j denoted by $\psi(s_i, s_j)$ in Eq. (6) is given by

$$\psi(s_i, s_j) = \exp\left(-\sigma \sum_{h=1}^{5} \frac{1}{\alpha_h} \chi^2(\phi_h(s_i), \phi_h(s_j))\right), \qquad (10)$$

where $\chi^2(\cdot,\cdot)$ denotes χ^2-distance between two histograms, α_h is a weighting factor of each histogram, and σ is the constant to control the effect of histogram distance. In our implementation, α_h is given by the average of χ^2-distances between all pairs of subsequences and σ is determined empirically to be 15.

Subsequence generation We simply set the length of a subsequence to 30, which is equal to the frame-per-second (fps) of the videos used in our experiment. The subsequences in a video are sampled every 10 frame and are partially overlapped with its neighbors.

Table 1: Details and statistics of Youtube co-activity dataset.

Class name (# of videos)	# of frames			# of co-activity frames			# of co-activity		
	avg.	max	min	avg.	max	min	avg.	max	min
Bench Press (10)	878.3	1809	329	452.0	879	149	1.0	1	1
Boxing Punching Bag (10)	2283.4	3161	1182	1246.8	1895	680	1.0	1	1
Clean&Jerk (10)	135.9	2187	471	437.3	891	195	1.0	1	1
Drumming (13)	154.3	3470	767	827.8	1933	248	1.3	3	1
Juggling Balls (12)	1105.3	1743	518	602.5	1228	184	1.2	2	1
Jumpping Rope (10)	1203.3	1894	573	641.6	1594	220	1.0	1	1
Pole Vault (10)	1518.2	2409	758	991.4	1654	197	1.0	1	1
Pommel Horse (10)	2293.8	3786	487	120.6	1650	200	1.0	1	1
Indoor Rock Climbing (10)	1897.1	4041	827	1271.6	2680	508	1.0	1	1
Soccer Juggling (10)	1098.4	2260	617	633.6	1380	170	1.0	1	1
Uneven Bars (10)	2071.4	2810	1475	1186.8	1560	980	1.0	1	1

Experiment

This section describes the dataset and evaluation protocols in our experiment and discuss the performance of our algorithm quantitatively and qualitatively.

Dataset

The proposed algorithm is evaluated in two datasets. Since there is no designated datasets for unsupervised co-activity detection, we constructed a new co-activity dataset based on the videos for 11 activities, which are collected from YouTube. There are at least 10 videos in each class, and all videos in each class contain one or more instances of the co-activity corresponding to class label. The ground-truths for co-activities are annotated manually for individual frames. The details and statistics of the dataset are summarized in Table 1, and videos can be found in a project page[1]. This dataset is referred to as the YouTube co-activity dataset afterwards. The other dataset is THUMOS14 (Jiang et al. 2014), which is a large scale dataset for temporal action detection and contains more than 200 temporally untrimmed videos for 20 activity classes. This is a very challenging dataset since the best known supervised action detection algorithm achieves only 14.7% in mean average precision.

Algorithms for Comparison

To evaluate the performance of our algorithm, AMC, which stands for absorbing Markov chain, we compare with a few external baseline algorithms, which include video co-segmentation (VCS) (Guo et al. 2013), Temporal Commonality Discovery (TCD) (Chu, Zhou, and De la Torre 2012), and video co-summarization (CS) (Chu, Song, and Jaimes 2015) techniques

VCS performs a co-activity detection based on trajectory matching, which is an intermediate goal of its full algorithm, video co-segmentation. TCD is a branch-and-bound technique to identify the best matching variable length subsequences, which overcomes the limitation of naïve exhaustive search for common activity retrieval. On the other hand, CS detects the co-activity from many pairs of videos, where it aggregates the pairwise results to detect co-activity from multiple videos. However, except CS, the other two methods are not capable of handling more than two videos. We

[1] http://cv.postech.ac.kr/coactivity/

Table 2: Accuracy of individual co-activity classes in YouTube co-activity dataset.

	(a) All videos in each class are tested at the same time.											
	Precision				Recall				F-measure			
	AMC	AMC-	PR	CS	AMC	AMC-	PR	CS	AMC	AMC-	PR	CS
BP	.75	.63	**.79**	.60	.83	**.93**	.55	.32	**.79**	.75	.65	.42
BPB	**.68**	.64	67	.55	**.93**	.91	.35	.32	**.78**	.75	.46	.40
C&J	.50	.41	**.66**	.34	.80	**.87**	.43	.52	**.62**	.56	.52	.41
Dr	.70	.68	**.83**	.61	**.95**	**.95**	.36	.28	**.81**	.79	.50	.38
JB	**.88**	.75	.74	.57	.85	**.94**	.32	.42	**.86**	.84	.45	.48
JR	**.99**	.98	.94	.60	**.75**	.69	.37	.36	**.85**	.81	.53	.45
PV	.87	.81	**.92**	.64	.95	**.97**	.46	.30	**.91**	.88	.61	.41
PH	.65	.59	**.90**	.64	.84	**.94**	.55	.38	**.73**	.72	.68	.48
IRC	.64	.61	**.93**	.68	**.61**	.36	.17	.23	**.63**	.45	.29	.34
SJ	**.73**	.69	.65	.67	.93	**.95**	.20	.52	**.82**	.80	.31	.59
UB	.72	.65	**.93**	.60	**.90**	.70	.24	.29	**.80**	.68	.38	.39
Avg.	.74	.68	**.81**	.59	**.85**	.84	.36	.36	**.78**	.73	.49	.43

	(b) Every pair of videos in each class is tested.																	
	Precision						Recall						F-measure					
	AMC	AMC-	PR	CS	TCD	VCS	AMC	AMC-	PR	CS	TCD	VCS	AMC	AMC-	PR	CS	TCD	VCS
BP	**.82**	.77	.79	.54	.61	.55	.71	.70	.36	**.88**	.34	.63	**.74**	.71	.48	.65	.42	.57
BPB	**.68**	.67	.66	.51	.58	.57	.76	.74	.29	**.80**	.11	.44	**.71**	.69	.39	.61	.18	.47
C&J	.55	.58	**.61**	.39	.46	.55	.68	.51	.22	**.80**	.39	.62	**.58**	.49	.31	.50	.40	.55
Dr	**.77**	.73	.76	.53	.22	.54	.74	.74	.29	**.80**	.08	.43	**.74**	.73	.39	.62	.12	.46
JB	**.75**	.63	.61	.47	.61	.52	**.71**	.57	.23	.64	.30	.43	**.71**	.58	.31	.52	.39	.44
JR	**.87**	.82	.84	.54	.70	.68	.76	.67	.36	**.82**	.35	.71	**.80**	.72	.47	.63	.44	.68
PV	.89	.86	**.90**	.65	.56	.79	.84	.85	.27	**.92**	.15	.65	**.86**	.85	.40	.75	.23	.67
PH	**.88**	.82	.85	.56	.53	.41	.71	.58	.32	**.83**	.16	.24	**.78**	.65	.45	.66	.23	.29
IRC	.71	.73	**.79**	.59	.78	.50	.43	.39	.15	**.73**	.24	.29	.52	.48	.24	**.65**	.35	.34
SJ	**.74**	.70	.68	.59	.40	.53	.69	.64	.22	**.86**	.13	.44	**.70**	.65	.32	.65	.19	.43
UB	.77	.80	.80	.57	**.85**	.22	.60	.33	.11	**.91**	.32	.08	.65	.43	.19	**.68**	.45	.11
Avg.	**.77**	.74	.75	.54	.57	.53	.69	.61	.26	**.82**	.23	.45	**.71**	.64	.36	.63	.31	.46

received the source codes of all other algorithms from the original authors.

In addition to these external methods, we implement two variations of our algorithm, which are referred to as AMC- and PageRank (PR). AMC- is designed to test the contribution of intra-subsequence edges, and does not have no intra-sequence edges. PageRank algorithm (Page et al. 1999) is based on the Markov chain without absorbing states, where the stationary distribution of Markov chain employed to identify the subsequences belonging to co-activity. The implementation of PageRank method is exactly same with our full algorithm, AMC, except that there is no absorbing state.

Evaluation Metrics

We employ precision and recall scores for quantitative comparison, which are computed based on the labels generated by the algorithms and manually annotated ground-truths. The formal definitions of precision and recall are given respectively by

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad \text{and} \quad \text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{11}$$

where TP, FP, and FN denote the number of frames corresponding to true positives, false positives, and false negatives, respectively. F-measure is a harmonic mean of precision and recall, which is defined as

$$\text{F-measure} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{12}$$

Results

Table 2(a) illustrates the quantitative results for all videos in each co-activity class of YouTube co-activity dataset, where our full algorithm denoted by AMC is compared with two internal baselines, AMC- and PR, and an external method denoted by CS. The performance of AMC is outstanding in terms of F-measure, and outperforms all other algorithms, particularly CS, in all classes with significant margins. For reference, BC achieves 0.72 in F-measure (0.73 in both precision and recall) when we assume that the oracle selects the cluster corresponding to co-activities, which is the most optimistic scenario for BC. Also, it turns out that adding intra-sequence edges and absorbing state is useful to improve accuracy; this is based on the performance observation of AMC- and PR, respectively. PR tends to have high precision while it performs poorly in terms of recall.

Table 2(b) summarizes the quantitative results of all applicable algorithms to pairs of videos in each class of YouTube co-activity dataset, where the average accuracy of all pairs of videos in each class is presented in the table. In addition to the algorithms used in the previous experiment, VCS and TCD are employed for evaluation. The average accuracy of our algorithms including AMC- and PR is lower than the one presented in Table 2. This is probably because our co-activity detection approach has merit to use many videos at the same time by identifying the common properties among videos more effectively but rejecting false alarms caused by accidentally similar representations between two videos. CS illustrates better accuracy in two video cases due to high recall scores, but this is because CS typically predicts a large portion of video frames as co-activity and our dataset tends to have many co-activity frames in videos. The performance of AMC is still best among all compared algorithms and the gaps against other algorithms are typically larger than the ones observed in Table 2(a). Note that the performance of BC (0.64 in F-measure) is also substantially worse than AMC even with oracle prediction in the final stage. The pair-wise experiment also shows that temporal locality and absorbing state are still crucial factors, which is consistent with the results from the experiments based on all videos as a whole.

Although TCD and VCS are proposed for the detection of co-activity only in a pair of videos, they are outperformed by the algorithms applicable to more than two videos, and such low accuracy results from the following reasons according to our observation. The objective of TCD may not be appropriate in practice; matching scores are often very high only in the parts of co-activities, not in the entire durations. Although VCS relies on trajectory matching, trajectory estimation is not reliable in YouTube co-activity dataset. Consequently, VCS often fails to find good matching pairs.

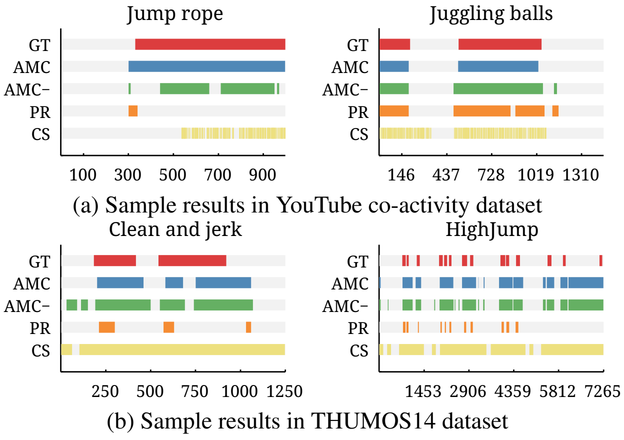
The performance in THUMOS14 dataset is illustrated in Table 3, where AMC outperforms other methods in both input scenarios and we can still observe the benefit of intra-

Table 3: Average accuracy over all classes in THUMOS14 dataset (All: all videos, Pair: pairwise videos)

	Precision					Recall					F-measure				
	AMC	AMC-	PR	CS	TCD	AMC	AMC-	PR	CS	TCD	AMC	AMC-	PR	CS	TCD
All	.28	.29	**.30**	.28	-	**.82**	.73	.16	.57	-	**.40**	.39	.19	.36	-
Pair	**.30**	.28	.27	.25	.22	.57	.44	.08	**.63**	.07	**.35**	.29	.11	.32	.08

sequence edges and absorbing state. However, overall accuracies of all algorithms are significantly are worse in this dataset. It suggests that THUMOS14 dataset is too wild to estimate co-activities in an unsupervised manner although our algorithm presents reasonably good performance. Note that the VCS code fails to report results due to its high memory requirement, which is partly because the longest video in THUMOS14 dataset contains more than 50K frames.

Figure 2 illustrates several qualitative co-activity detection results of all applicable algorithms to more than two videos, where *x*-axis denotes frame index and algorithm names including ground-truths are listed in *y*-axis. Each color bar visualizes the temporal configurations of detected co-activity frames. Our algorithm localizes co-activities better than other ones, and identify multiple instances in a video effectively. Figure 3 presents that our algorithm also works better than all others for a pair of videos although some methods such as (Chu, Zhou, and De la Torre 2012; Guo et al. 2013) are specialized in two video case only.



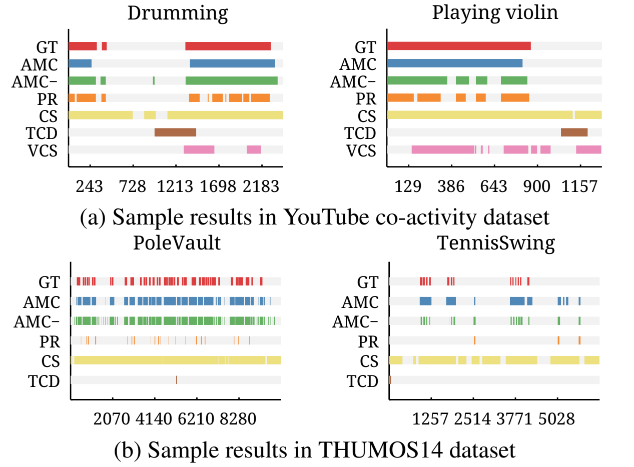
(a) Sample results in YouTube co-activity dataset

(b) Sample results in THUMOS14 dataset

Figure 2: Qualitative co-activity detection results when all videos in each class are tested.

The accuracies with respect to the variations of subsequence length are presented in Table 4. Table 5 shows the results with the variations of subsequence overlap ratio given the subsequence length 30. Both factors affect overall performance marginally; the F-measures of our algorithm hardly change and are still better than other methods. Note that our parameter choices are not necessarily best ones.

For more comprehensive evaluation, we present more results in our project page, where the performance in UCF (Soomro, Zamir, and Shah 2012) and Hollywood (Laptev et al. 2008) dataset is discussed as well.



(a) Sample results in YouTube co-activity dataset

(b) Sample results in THUMOS14 dataset

Figure 3: Qualitative co-activity detection results when every pair of videos in each class is tested.

Table 4: Impact of subsequence length to accuracy

subsequence length		All videos			Pairwise		
		Precision	Recall	F-measure	Precision	Recall	F-measure
YouTube	30	**.74**	**.85**	**.78**	**.77**	.69	.71
	60	.72	.82	.77	.76	**.72**	**.72**
	90	.73	.79	.75	.74	.71	.71
THUMOS14	30	**.28**	.82	.40	**.30**	.57	.35
	60	**.28**	**.84**	**.41**	**.30**	.64	**.37**
	90	.27	.81	.40	.29	**.65**	**.37**

Table 5: Impact of subsequence overlap ratio to accuracy

subsampling rate (# of overlapped frames)		All videos			Pairwise		
		Precision	Recall	F-measure	Precision	Recall	F-measure
YouTube	10 (20)	**.74**	**.85**	**.78**	**.77**	**.69**	**.71**
	20 (10)	.71	.85	.77	**.77**	.67	.69
	30 (0)	.72	.85	.77	**.77**	.66	.69
THUMOS14	10 (20)	.28	.82	.40	**.30**	**.57**	**.35**
	20 (10)	.31	.80	**.41**	**.30**	.55	.34
	30 (0)	.31	.80	**.41**	**.30**	.54	.34

Conclusion

We proposed an unsupervised co-activity detection algorithm based on absorbing Markov chain. Our algorithm handles more than two videos naturally, is capable of identifying multiple co-activity instances in a video and runs fast once feature descriptor is computed for each subsequence. We compared our algorithm with existing techniques related to unsupervised co-activity detection in two datasets; the YouTube co-activity dataset was constructed for the purpose and THUMOS14 dataset was employed additionally for performance evaluation. The proposed technique achieves outstanding performance in both datasets compared to existing ones without ad-hoc heuristics, parameter tuning, and complex post-processing.

Acknowledgement This work was supported by Samsung Electronics Co., Ltd.

References

Chen, C.-Y., and Grauman, K. 2012. Efficient activity detection with max-subgraph search. In *CVPR*, 1274–1281.

Chiu, W.-C., and Fritz, M. 2013. Multi-class video co-segmentation with a generative multi-video model. In *CVPR*, 321–328.

Cho, M.; Lee, J.; and Lee, K. M. 2010. Reweighted random walks for graph matching. In *ECCV*, 492–505.

Chu, W.-S.; Song, Y.; and Jaimes, A. 2015. Video co-summarization: Video summarization by visual co-occurrence. In *CVPR*, 3584–3592.

Chu, W.-S.; Zhou, F.; and De la Torre, F. 2012. Unsupervised temporal commonality discovery. In *ECCV*, 373–387.

Duchenne, O.; Laptev, I.; Sivic, J.; Bach, F.; and Ponce, J. 2009. Automatic annotation of human actions in video. In *ICCV*, 1491–1498.

Guo, J.; Li, Z.; Cheong, L.-F.; and Zhou, S. Z. 2013. Video co-segmentation for meaningful action extraction. In *ICCV*, 2232–2239.

He, P.; Xu, X.; and Chen, L. 2012. Constrained clustering with local constraint propagation. In *ECCV*, 223–232.

Jiang, B.; Zhang, L.; Lu, H.; Yang, C.; and Yang, M.-H. 2013. Saliency detection via absorbing markov chain. In *ICCV*, 1665–1672.

Jiang, Y.-G.; Liu, J.; Roshan Zamir, A.; Toderici, G.; Laptev, I.; Shah, M.; and Sukthankar, R. 2014. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/.

Joulin, A.; Bach, F.; and Ponce, J. 2010. Discriminative clustering for image co-segmentation. In *CVPR*, 1943–1950.

Kim, G., and Xing, E. P. 2012. On multiple foreground cosegmentation. In *CVPR*, 837–844.

Laptev, I.; Marszalek, M.; Schmid, C.; and Rozenfeld, B. 2008. Learning realistic human actions from movies. In *CVPR*, 1–8.

Minnen, D.; Isbell, C. L.; Essa, I.; and Starner, T. 2007. Discovering multivariate motifs using subsequence density estimation. In *AAAI*, 615–620.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

Seneta, E. 2006. *Non negative matrices and Markov chains*. Heidelberg: Springer.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: a dataset of 101 human actions classes from videos in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence* abs/1212.0402.

Tian, Y.; Sukthankar, R.; and Shah, M. 2013. Spatiotemporal deformable part models for action detection. In *CVPR*, 2642–2649.

Wang, H.; Kläser, A.; Schmid, C.; and Liu, C.-L. 2011. Action recognition by dense trajectories. In *CVPR*, 3169–3176.

Xiong, C., and Corso, J. J. 2012. Coaction discovery: segmentation of common actions across multiple videos. In *Proceedings of the Twelfth International Workshop on Multimedia Data Mining*, 17–24.

Yuan, J.; Liu, Z.; and Wu, Y. 2011. Discriminative video pattern search for efficient action detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(9):1728–1743.