Unsupervised Analysis of Activity Sequences Using Event-Motifs

Raffay Hamid, Siddhartha Maddi, Aaron Bobick, Irfan Essa
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
raffay, maddis, afb, irfan@cc.gatech.edu

ABSTRACT

We present an unsupervised framework to discover characterizations of everyday human activities, and demonstrate how such representations can be used to extract points of interest in event-streams. We begin with the usage of *Suffix Trees* as an efficient activity-representation to analyze the global structural information of activities, using their local event statistics over the entire continuum of their temporal resolution. Exploiting this representation, we discover characterizing event-subsequences and present their usage in an ensemble-based framework for activity classification. Finally, we propose a method to automatically detect subsequences of events that are locally atypical in a structural sense. Results over extensive data-sets, collected from multiple sensor-rich environments are presented, to show the competence and scalability of the proposed framework.

Categories and Subject Descriptors

[**Video and multimedia analysis**]: Machine Learning Techniques for Event Mining; [**Applications**]: Home surveillance, Crime prevention

1. INTRODUCTION & PREVIOUS WORK

Consider a household kitchen where activities such as cooking, setting up the table, washing the dishes *etc.* take place. Such active environments consist of certain key-objects interacting with each other. The interaction of these objects in a particular manner constitutes an event, while a sequence of such events constitutes an activity. Understanding what is happening in such active settings offers various ramifications, some of the more practical of which involve systems that can be used for automatic surveillance or supporting users in ubiquitous environments.

Starting from low-level perceptual inputs, several abstractions need to be addressed before one can get at some of the more interesting high-level questions. One of the key challenges is the transformation of semantically agnostic signals to some meaningful mid-level representation, that sufficiently encodes the structure of activities, while being reasonably robust to sensor-noise. Since for any fairly unconstrained setting, a prior model for the activity-structure is generally not at hand, it is imperative to have an efficient representation that facilitates the discovery of various activity-classes with minimal supervision. Equally important is the question of finding characterizations for these different kinds of activities that can in turn be used for activity classification and detection of structurally anomalous event subsequences that may be of interest (from hereon referred to as *Anomalies*). These anomalies can later on be shown to human observers for further analysis. In this work we attempt to elucidate the unsupervised discovery of activity-class characterization in the backdrop of everyday activity-analysis.

The main contributions of this work are:

- An efficient activity-representation that allows analysis of global structural information of activities, using their local event statistics.
- Unsupervised discovery and usage of recurrent event-*motifs* (see Section 2.2) in an ensemble-based framework for activity classification.
- Unsupervised detection of anomalies in event-streams.

A substantive majority of the previous work in the field of computational scene analysis has been focused on model based activity recognition where a set of target activities is explicitly modeled, followed by the learning of the model parameters given some training data (see *e.g.* [13] [7] [12]). Since these approaches make a strong assumption about the availability of *a priori* knowledge of the activity-structure, which is not always at hand, we are interested in discovering this structure with minimal supervision.

The perspective of detecting anomalies based on the dissimilarity from what is regular, has only recently been applied to the field of scene analysis [18] [14]. In this work however, we propose to extract the global structural information of activities simply by considering their *local* event-statistics. This allows us to analyze activities from a by-parts perspective as opposed to a wholistic one proposed in the previous approaches.

While ensemble-based classification of sequences using their motifs as features has been previously exploited for text classification [11], these motifs are mostly noise-free. We extend such an approach to the problem of activity classification where the motifs are sparse and noisy.

WOODSTOCK '97 El Paso, Texas USA

Figure 1: A Person interacts with a sink.

Previous work regarding novelty detection in sequences has focused on detecting subsequences that deviate from the local sequential statistics [1]. These approaches implicitly rely on highly structured sequences to leverage a consistent frequential signature. However, for the environments that we are interested in, such an assertion may not always hold. Our approach therefore assumes the normality of any event-pattern contained in the training set, exacting a more prudent use of the training data.

2. ACTIVITY-REPRESENTATION

Looking at an activity as a sequence of discrete events, two quantities emerge that are of fundamental interest: *Content* (*i.e.* the events that span the activity) and *Structure* (*i.e.* the order in which the constituent events are arranged). This treatment of an activity is similar to the representation of a document as a set of words - also known as the Vector Space Model (VSM) [10]. Such perspective of activities requires a set of events (an *event-vocabulary*) that spans the space of activities taking place in a context-setting.

2.1 Event Vocabulary using Contextual Knowledge

As we are interested in systems that operate using perceptual inputs, the events must be defined such that they are detectable from some low-level perceptual primitives. Since convergence to any meaningful inference requires a set of *a priori* assumptions [16], we use contextual prior knowledge to construct a finite set of key-objects, the interaction amongst which govern the generation of different events. To exemplify our approach, consider Figure 1 showing an event from one of the environments that we analyzed (a household kitchen).

These key-objects are automatically tracked (explained in Section 6.2) over the entire duration of an activity. It is further assumed that at any instant in time the person is interacting with the nearest key-object. While there is no explicit knowledge encoded in the system about the nature of these interactions, our contextual priors facilitate the transformation of such agnostic features into semantically meaningful events.

2.2 Activities as Suffix Trees

As an improvement on the usage of Vector Space Model for activity-representation [14], recently there has been some interest in treating activities as bags of discrete event n-grams [4] [5], where activities are represented by sparse histograms of counts of event n-grams. It is evident that higher values of n would capture the temporal order information of events more rigidly, and would entail a more discriminative representation. Therefore, for a fixed value of n, such an approach falls short of capturing the activity-structure at multiple scales.

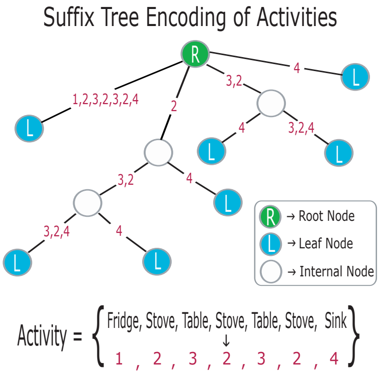

Figure 2: A Suffix Tree T defined on an activity sequence a can represent every subsequence in a with at most $2m$ nodes where m is the length of a.

As opposed to event n-grams, we propose the use of a data-structure called *Suffix Tree* [3] as an efficient activity-representation to analyze the global structural information of activities, using their local event statistics over the entire continuum of their temporal resolution. Unlike event n-grams, which are exponential in their complexity, a Suffix Tree T lets one enumerate *all* unique subsequences of an activity sequence a in time linear[1] in $||a||$. The figurative illustration of the transformation of an activity sequence into its equivalent Suffix Tree is shown in Figure 2. By construction, no two edges leaving a node in Suffix Trees can have labels that begin with the same symbol, implying that starting from the root node, every unique subsequence in a can be generated by traversing through T.

We define an *Event-Motif* in a as the event subsequence generated by starting from the root node in T, and traversing to a node. Note that the occurrence of all other event subsequences in a is completely dependent on the occurrence of event motifs of a [15]. These motifs therefore encode the structural signature of a and are of of fundamental interest for the representation of a.

2.3 Activity Similarity Metric

Following [4], our view of the similarity between a pair of activity sequences consists of two factors, the *core structural differences* and differences based on the *frequency of occurrence* of motifs. Note that here we are using event motifs as the basis to compute the similarity between two activity sequences as opposed the usage of event n-grams as proposed in [5].

Let a and b denote two sequences of events, and let their sets of corresponding motifs be denoted by H_a and H_b. For $x \in H_a \cup H_b$, let $f(x|H_a)$ and $f(x|H_b)$ denote the counts

[1] For any subsequence w occurring in a, $\exists$ a node n in T representing subsequence $\bar{w}$ *s.t.* w is a prefix of $\bar{w}$, and w occurs iff $\bar{w}$ does [15]. Thus while the upper bound on the number of subsequences for a is quadratic in $||a||$, the number of nodes in T is only linear in $||a||$.

of x in H_a and H_b respectively[2]. We define the similarity between two event sequences as:

$$sim(a,b) = 1 - \kappa \sum_{x \in H_a \cup H_b} \frac{|f(x|H_a) \; - \; f(x|H_b)|}{f(x|H_a) \; + \; f(x|H_b)} \qquad (1)$$

where $\kappa = 1/(||H_a|| + ||H_b||)$ is the normalizing factor, and $|| \cdot ||$ computes the cardinality of a set. While our proposed similarity metric: (1) conforms to the property of *Identity of indiscernibles*, (2) is *commutative*, and (3) is *positive semi-definite*, it does not however follow *Cauchy-Schwartz inequality*, making it a divergence rather than a true distance metric.

3. ACTIVITY-CLASS DISCOVERY

Given a context-setting with fairly structured activities, it is plausible to assume that the activity-instances occurring in the environment do not span the activity-space uniformly. Rather, there exist disjunctive activity-sets with high internal similarity while low similarity across the sets. This assertion is backed by the weak assumption that by exploiting the contextual prior knowledge, the detected events sufficiently encode the underlying structure of activities [9].

3.1 Activity-Class as Maximal Clique

Starting off with a set of A activity-instances, we consider this activity-set as an undirected edge-weighted graph with A nodes, each representing the set of motifs for one of the A activity sequences. The weight of an edge is the similarity between a pair of nodes as defined in Section 2.3. We formalize the problem of discovering activity-classes as searching for edge-weighted maximal cliques[3]in the graph of A activity-instances. Indeed, in the past, some authors have argued that a maximal clique is the strictest definition of a cluster [2]. We proceed by finding a maximal clique in the graph, removing that set of nodes from the graph, and repeating this process iteratively with the remaining set of nodes, until there remain no non-trivial maximal cliques (with nodes ≥ 3) in the graph. The leftover nodes after the removal of maximal cliques are dissimilar from most of the (regular) nodes.

3.2 Maximal Cliques using Dominant Sets

Finding maximal cliques in an edge-weighted undirected graph is a classic graph theoretic problem. Because combinatorially searching for maximal cliques is computationally hard, numerous approximations to the solution of this problem have been proposed [17]. For our purposes, we adopt the approximate approach of iteratively finding *dominant sets* of maximally similar nodes in a graph (equivalent to finding maximal cliques) as proposed in [8]. Besides providing an efficient approximation to finding maximal cliques, the framework of dominant sets naturally provides a principled measure of the cohesiveness of an activity-class as well as a measure of node participation in its membership class. The formalization of dominant sets is summarized in Appendix A (for details, please see [8]).

4. ACTIVITY CLASSIFICATION

Given that an event motif partially captures the structure of an activity-class, we propose to use such motifs as features for an ensemble-based activity classification framework. Following [11], we use each motif w to construct a hypothesis of the form

$$g_w(a,\ell) = \begin{cases} c_{1,\ell} & \text{if } w \in a \\ c_{0,\ell} & \text{otherwise} \end{cases} \qquad (2)$$

where a is an activity and $c_{1,\ell}$ is the confidence that a containing w belongs to class ℓ. Also, $c_{0,\ell}$ is the confidence that a not containing w belongs to class ℓ. These confidences, computed by a weak learner, depend upon the fraction of examples in which a particular feature is present or absent from a certain class relative to all other classes. The weight of each example for this computation is based on its degree of importance to a particular class assigned by the AdaBoost.MH algorithm (for details, please see [11]). Given these confidence values, for each round of Boosting, a weak learner selects the feature that minimizes the empirical Hamming Loss of AdaBoost.MH algorithm.

4.1 Noisy Features

Sensor-noise and structural alteration of activities in a fairly unconstrained setting generally lead to instances of motifs with various structural mutations. To incorporate such variations upon motifs, we extend the algorithm proposed in [11] by modifying the weak hypothesis (Equation 2) to one that considers the occurrence of a motif in a probabilistic manner given as:

$$g_w(a,\ell) = \{1 - p(w \in a)\}c_{0,\ell} + p(w \in a)c_{1,\ell} \qquad (3)$$

where $p(w \in a)$ is the probability that w is contained in a, and is inversely proportional to the minimum *edit distance* between the motif and any subsequence occurring in a [3]. The probability that an activity a contains a motif w is given by $\exp(-y \cdot d/||w||)$, where $y \geq 0$ is an error sensitivity parameter and d is the minimum edit distance between w and a subsequence of a.

5. ANOMALY DETECTION

One of the key problems regarding novelty detection is the possibility of executing a certain activity in multiple legitimate ways, whose degree of legitimacy is independent of the number of times the activity is performed in any one of such ways. Consider for example, the activity of making coffee. The fact that generally people take coffee with cream does not make this particular way of making coffee any more normal than making coffee without cream. With this perspective at hand, we argue that given a set of legitimate activity sequences A, any subsequence of events is acceptable so long as it occurred in A. This approach puts the burden of proof on the legitimacy of the training data, resulting in fewer false negatives. Since such an approach is rather conservative towards the notion of regular, it will generate more false positives, leaving the final decision to the judgement of the human observer.

5.1 Defining an Anomaly

Intuitively, for an event stream a belonging to a particular class c, we consider an event in a to be an anomaly, if the event-subsequence observed since the previous anomaly in a

[2] Suffix Trees allows one to compute these frequencies in linear time [3]

[3] Recall that a subset of nodes of a graph is a clique if all its nodes are mutually adjacent; a *maximal* clique is not contained in any larger clique, whereas a *maximum* clique has largest cardinality.

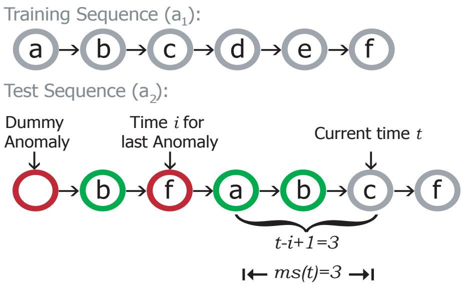

Figure 3: Figurative illustration of the notion of anomalies. At time t, the *match statistics* ms of a_2 is 3. This is because the longest subsequence in a_2 ending at t, which also occurs in a_1 is a, b, c. As the number of events in a_2 since the last anomaly till t is also 3, t is not an anomaly ($h_a(t) = 0$). Repeating this algorithm for the next event f would show that $t + 1$ is an anomaly.

does not occur in c. We consider a dummy anomaly at the beginning of every event stream.

Definition: Let S_c denote the set of all event-subsequences occurring in activities belonging to class c. For $1 \leq t \leq ||a||$, define

$$h_a(t) = \begin{cases} 1 & \text{if} \quad t = 0 \ \vee \ s \notin S_c \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where 1 represents an anomaly in a at t and $s = a[i, t]$ with $i = max\{0 \leq j < t : h_a(j) = 1\} + 1$.

5.2 Anomalies Using Match Statistics

We exploit the notion of *Match Statistics* of an activity sequence to efficiently detect anomalies. Slightly deviating from the standard definition of *match statistics* $ms(t)$ [3], here we define $ms(t)$ of a as the length of the longest subsequence in S_c that is a suffix of $a[1, t]$. We now explain how we can compute the function $h_a(t)$(Equation 4) using the *match statistics* of a.

Theorem 1 Let $1 \leq t \leq ||a||$ and let $i = max\{0 \leq j < t : h_a(j) = 1\} + 1$. Then $h_a(t) = 1$ iff $ms(t) < (t - i + 1)$

Theorem 1 forms the bridge between finding anomalies in an event sequence to computing its *match statistics*. Intuitively, Theorem 1 implies the occurrence of an anomaly in a test activity sequence a at t, if the length of the subsequence from t to the last anomaly exceeds the value of *match statistics* of a at t. (for proof, see Appendix B). Based on Theorem 1, an algorithm for finding anomalies is presented in 1. A figurative illustration of the notion of anomaly is shown in figure 3. *ms(t)* in Algorithm 1 can be computed in $O(||a||)$ given a Suffix Tree for the sequences in class c [3].

6. EXPERIMENTS & RESULTS

We demonstrate the competence of our proposed framework over an extensive data-set of activities collected in a Household Kitchen and a Loading Dock environments, comparing our results to the approach proposed in [4].

6.1 Experimental Setup for Household Kitchen

We deployed an orthographic projection camera in the ceiling of a household kitchen to record the activities of a user performing 7 classes of activities each constituting of 20 instances. The semantic labels of these activities are[4]: ($C1$) *Cook Dinner*, ($C2$) *Clean-up Kitchen after cooking*, ($C3$) *Put Dishes in Dishwasher*, ($C4$) *Remove Dishes from Dishwasher*, ($C5$) *Eat Cereal*, ($C6$) *Set-up Table*, and ($C7$) *Fry Egg* . Using our contextual prior knowledge, we identified 10 objects whose interactions with the user dictate the activity-structure. The floor layout of the kitchen along with the key objects are shown in Figure 4. Occurrences of all events contained between an Enter and and Exit event are considered as an independent activity.

Algorithm 1 Anomaly Detection

```
Require: A test sequence $a$ that belongs to class $c$ and a
  set of training sequences $A_c$ that also belong to class $c$.
  for $t = 1$ to $||a||$ do
      Compute $ms(t)$ for $a$ given $A_c$
  end for
  Let $lastAnomaly := 0$
  for $t = 1$ to $||a||$ do
      if $(t - lastAnomaly) > ms(t)$ then
          $h_a(t) = 1$
          $lastAnomaly = t;$
      else
          $h_a(t) = 0$
      end if
  end for
```

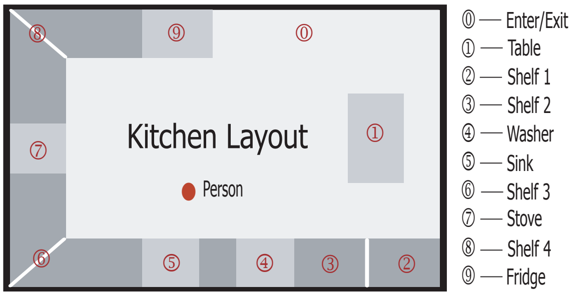

Figure 4: Orthographic view of kitchen layout with 10 key objects and a person are identified. Given the geometry of the scene, the key-locations for Enter and Exit events are the same.

6.2 Object Tracking

As can be observed from Figure 4, all key-objects except the person are static for the entire length of an activity, reducing our problem to automatic tracking of person. For this work, we implemented the person tracking framework proposed in [6]. For extracting the person from the background image, we learned a set of *Gaussian Mixture Models* for the chromatic contents of the background which are in turn used for computing the likelihood for the presence of the person in the image space. Given such likelihoods, we used a particle filter framework to search through the image space for converging to the maximum *a posteriori* position of the person. This *MAP* estimate in one frame is propagated to the next as the initial state of the particle filter for

[4] While our framework is unaware of these labels, we maintain this information for the performance analysis of our unsupervised class discovery algorithm as explained in Section 6.3.

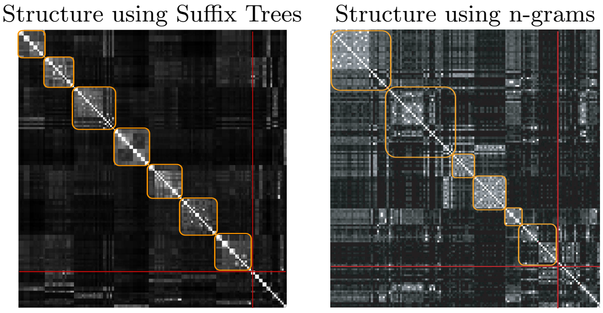

Figure 5: Visualization of Similarity matrices representing the discovered classes using Suffix Trees as compared to n-grams. White represents structurally similar activities while black stands for ones that are structurally different. Note that the similarity matrix obtained using Suffix Trees has compact and conjunctive white color as opposed to widely spread and ill-partitioned white color in the similarity matrix obtained by using n-grams.

the next iteration (for details, please see [6]).

6.3 Performance Analysis for Class Discovery

For every class that our framework discovered, the final class-label is assigned based on the labels of the majority of the class-members. Moreover, any two classes with the same class labels are merged. We considered two metrics for analyzing the performance of our approach regarding the goodness of each discovered cluster:

$$\%Fidelity = \frac{Cardinality\ of\ majority\ population}{Discovered\ cardinality\ of\ cluster} \quad (5)$$

$$\%Coverage = \frac{Cardinality\ of\ majority\ population}{Actual\ cardinality\ of\ cluster} \quad (6)$$

Intuitively, the metric % *Fidelity* captures how closely does a discovered cluster correlate with our semantic model of the corresponding activity-class, while % *Coverage* indicates how well does it span the activity space. With this approach, our proposed framework was able to discover all 7 clusters with average % *Fidelity* equal to 96.21% and an average coverage of 82.1% across the 7 discovered clusters. These results are enumerated in Table 1. To compare the competence of our framework with that proposed in [4], we implemented the n-grams based approach on our collected kitchen activities. The n-grams based approach discovered 6 of the 7 clusters failing to discover the activity-class *Fry Egg*, giving % *Fidelity* of 77% with an average coverage across the discovered 6 clusters equal to 74.1% (Table 2). The visualization for the extracted activity-structure using Suffix Trees compared with that obtained by using n-grams is shown in Figure 5.

6.4 Performance Analysis for Activity Classification

Out of the 140 activities, we randomly removed 50 sequences (using a uniform distribution), and trained the ensemble classifier on the remaining 90 activities using 7 rounds of boosting. The average classification results for 100 such independent trials are given in Table 3.

	D1	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
C1	14	0	4	0	0	0	0
C2	0	15	0	0	0	0	0
C3	0	0	17	0	0	0	0
C4	0	0	2	16	0	0	0
C5	0	0	0	0	17	0	0
C6	0	0	0	0	0	18	0
C7	0	0	0	0	0	0	18
Fidelity	100	100	73.91	100	100	100	100
Coverage	70	75	85	80	85	90	90

Table 1: Results for the discovered activity classes using Suffix Tree to extract the activity-structure. The labels C's represent the ground truth information about the 7 classes while labels D's represent the discovered classes.

	D1	**D2**	**D3**	**D4**	**D5**	**D6**
C1	20	0	0	0	0	0
C2	0	17	0	0	0	0
C3	4	5	11	0	0	0
C4	0	4	0	16	0	0
C5	2	3	0	1	8	4
C6	0	0	0	0	0	17
C7	6	2	0	8	0	0
Fidelity	62.5	54.8	100	64	100	81
Coverage	100	85	55	80	40	85

Table 2: Results for the discovered activity classes using n-grams to extract the activity-structure. Since no such cluster was discovered with the majority of the members belonging to activity-class *Fry Egg*, only 6 of the 7 classes could be discovered.

	D1	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
ACR	83%	94%	78%	85%	91%	98%	92%

Table 3: Average Classification Results (ACR) for kitchen activities over 100 independent trials. In each trial 50 sequences were randomly removed from the data-set, training on the remaining 90 activities using 7 rounds of boosting.

6.5 Anomalies for Kitchen Domain

Out of the 140 activities that were partitioned into 7 activity-classes, 19 activities could not be clustered since they were structurally too different from the remaining activity sequences. For these 19 activities, our anomaly detection algorithm labeled approximately 15 percent of the events as anomalies in contrast to only 5 percent anomalies detected in the remaining 121 activities, indicating the coherency of our general approach. Moreover, we hand-labeled anomalous event-subsequences in these 19 activities and compared them with those detected by our system. These results are given in Table 4.

Analyzing these results semantically, 3 distinct types of anomalies emerge. The first ones are those that are actually alarming and must be brought under attention to some human personal. An example of this type of anomalies is when a user forgets to wash vegetables before cooking them. Another example is when the user forgets to put dishes in the sink after cooking a meal. The second type of anomalies are the ones where either the order of events is altered or

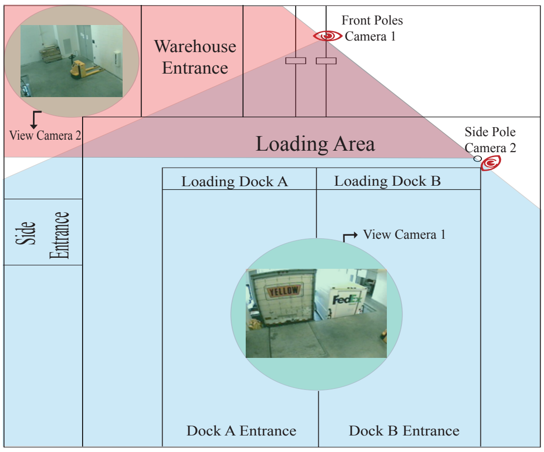

Figure 6: A schematic diagram of the camera setup at the loading dock area with overlapping fields of view (FOV). The FOV of camera 1 is shown in blue while that of camera 2 is in red. The overlapping area of the dock is shown in purple.

	True +ive	False -ive	False +ive
Rates	90%	10%	6%

Table 4: Anomaly Detection results for 19 atypical kitchen-activities. True positive rate is the fraction of anomalies labeled by the human observer that were also detected by the system. False negative rate is the fraction of anomalies labeled by the human observer not detected by the system. False positive rate is the fraction of events not labeled as anomalies by the human observer, that were detected by the system.

their frequency is changed. An example of such anomalies are when the user takes more than usual number of dishes out of a shelf while setting up the dinner table. The third type of anomalies are ones that are mostly benign and may not always be considered interesting to a human observer. These include tangential events that a user may do while performing another task (*e.g* drink water, rest on a chair *etc*). This underscores the need to have a human expert in an *active-learning* paradigm whose feedback could be used to learn what makes a surprise alarming/interesting.

6.6 Experimental Setup for Loading Dock

To test the generalizability of our proposed framework, we used the data for package deliveries activities at the loading dock of a *Barnes&Noble*® bookstore collected by [4]. In this data, two cameras with partially overlapping fields of view are used. A schematic diagram with sample views from the two cameras is shown in Figure 6. Daily activities from 9*a.m.* to 5*p.m.*, 5 days a week, for over one month are recorded. Based on the semantics of the activities taking place in that environment, an event vocabulary of 61 events was constructed. Every package delivery activity has a known starting event, *i.e. Delivery Vehicle Enters the Loading Dock* and a known ending event, *i.e. Delivery Vehicle Leaves the Loading Dock.* Based on the event vocabulary of 61 events, 150 package delivery activities were labeled.

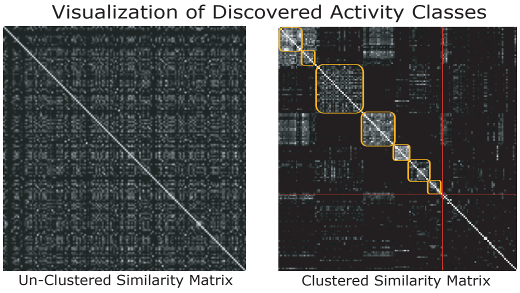

Figure 7: Visualization of similarity matrices before and after activity-class discovery for the Loading Dock environment.

	D1	D2	D3	D4	D5	D6	D7
ACR	87%	84%	94%	71%	72%	82%	81%

Table 5: Average Classification Results (ACR) for Loading Dock activities over 100 independent trials. In each trial 50 sequences were randomly removed from the data-set, training on the remaining 100 activities.

6.7 Activity-Class Discovery & Classification

We were able to discover 7 activity-classes composed of 108 activities. The visualization for these clusters is given in Figure 7. Analysis of the discovered classes reveals a strong structural similarity amongst the class members. For instance, all activities grouped in class 1 were *UPS*® deliveries, while activities in class 2 were mostly *Fed Ex*®. The fact that our framework was able to group semantically meaningful activities without any explicit knowledge implies the competence of our approach.

Out of the 150 activities, we randomly removed 50 sequences, and trained the ensemble classifier on the remaining 100 activities. The average classification results (ACR) for 100 such independent trials are given in Table 5. Since in a fairly unconstrained environment such as a Loading Dock, there is a substantial variance among activities, it is plausible to believe that as our training data starts spanning the activity-space more completely, the discovered activities would tend to be more discriminative resulting in an improvement in the ACR. Moreover, the rate of activities detected as anomalies would also converge to a smaller value.

6.8 Anomalies for Loading Dock

For the 42 atypical activities our anomaly detection algorithm labeled approximately 28 percent of the events as anomalies in contrast to 18 percent anomalies detected in the remaining 108 activities. Some of the truly alarming anomalies detected include a truck driving out without closing its back door, relatively excessive number of people unloading a truck, loading/unloading packages from the right door of the truck instead of the back door *etc.* As in the kitchen environment, some of these anomalies were false positives, for instance placing/removing more than usual number of packages in a delivery vehicle, successively opening and closing a truck door multiple times *etc.* This observation reiterates the potential to have a human expert whose feedback could be used to learn what makes an anomaly alarming/interesting.

7. DISCUSSION & FUTURE WORK

In this work, we attempted to investigate the problem of unsupervised structure analysis of event streams. The key question that we are interested in is whether there exists sufficient structural signature at a local temporal scale that can entail a reasonably disjunctive partitioning of the activity-space. Understanding this requires a notion as to *what* granularity of scale should the events be analyzed at. Lower granularity would result in more discriminative characterizations, but at the same time would be brittle, and therefore prone to sensor noise. Higher granularity will lead to representations more robust to sensor-noise, but with less discriminative prowess. The usage of Suffix Trees provides an efficient activity representation capable of analyzing event sequences over the entire continuum of their temporal scale. The fact that the complexity of this representation is only linear in the length of the activity sequence makes this representation all the more fitting.

The fact that event motifs partially capture the activity-class structure, supports their usage in an ensemble learning framework. Our proposed notion of probabilistic occurrence of event motifs allows such an approach to scale to environments with sensor noise.

The considerably large space of legitimate ways in which an activity can be performed, calls for being rather conservative towards ones notion of regular. By highlighting event subsequences not observed in the training set, our approach helps reduce the false negative rate, and leaves the final decision to an expert.

Currently, our framework is indifferent to the temporal duration for which an agent interacts with a scene object. In the future we intend to incorporate this temporal information within Suffix Tree framework, to facilitate the extraction of finer nuances of activity structure. We also plan to perform an empirical noise analysis of our Suffix Tree activity representation, to see how its performance degrades in the face of considerable levels of noise. Moreover, we are interested in exploiting multiple sensor-modalities so we could use a richer event vocabulary. Finally, we intend to explore how having a human user in an *active learning* paradigm could improve the detection rate of our proposed anomaly detection framework.

8. REFERENCES

[1] A. Apostolico, M. Bock, S. Lonardi, and X. Xu. Efficient detection of unusual words. *J. Computational Biology*, 7, 1/2:155–212, January.

[2] J. Auguston and J. Miker. An analysis of some graph theoretical clustering techniques. *J. ACM*, 17(4):571–588, 1970.

[3] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press; 1st edition, 1997.

[4] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman. Detection and explanation of anomalous activities: Representing activities as bags of event n-grams. In *IEEE CVPR*, 2005.

[5] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, and C. Isbell. Discovery and characterization of activities from event-streams. In *Conference on Uncertainty in AI (UAI)*, 2005.

[6] M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *ICCV*, 2001.

[7] D. Moore and I. Essa. Recognizing multitasked activities using stochastic context-free grammar, using video. In *AAAI*, 2002.

[8] M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *CVPR*, 2003.

[9] E. Rosch, C.Mervis, W. Gray, D. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8, 1976.

[10] G. Salton. *The SMART Retrieval System - Experiment in Automatic Document Processing.* Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[11] R. E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[12] Y. Shi, Y. Huang, D. Minen, A. Bobick, and I. Essa. Propogation networks for recognizing partially ordered sequential action. In *In Proceedings of IEEE Conference on CVPR*, 2004.

[13] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *PAMI*, 20:1371–1375, 1998.

[14] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *PAMI*, 22(8):747–757, 2000.

[15] E. Ukkonen. Constructing suffix trees on-line in linear time. In *Proc. Information Processing 92, Vol. 1, IFIP Transactions A-12 484-492*, 1994.

[16] S. Watanabe. *Pattern recognition: Human and mechanical.* Wiley & Sons, New York, 1985.

[17] Y. Weiss. Segmentation using eigen vectors: a unifying view. In *ICCV*, 1999.

[18] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Proc. of IEEE CVPR*, 2004.

APPENDIX

Appendix A.

Let the data to be clustered be represented by an undirected edge-weighted graph with no self-loops $G = (V, E, \vartheta)$ where V is the vertex set $V = \{1, 2, ...K\}$, $E \subseteq V \times V$ is the edge set, and $\vartheta : E \rightarrow \mathbb{R}^+$ is the positive weight function. The weight on the edges of the graph are represented by a corresponding $K \times K$ symmetric similarity matrix $A = (a_{ij})$ *s.t.* $(a_{ij}) = sim(i, j)$ iff $(i, j) \in E$ and 0 otherwise. The function *sim* is computed using our proposed notion of similarity as described in §2.3. Let $Q \subseteq V$ be a non-empty subset of vertices and $i \in Q$, such that $awdeg_Q(i) = \frac{1}{||Q||} \sum_{j \in S} a_{ij}$. Moreover, for $j \notin Q$, we define Φ_Q as $\Phi_Q(i, j) = a_{ij} - awdeg_Q(i)$. Let $S \subseteq V$ *s.t.* $||Q|| \neq 0$ and $i \in Q$. The weight of i w.r.t. Q is given as:

$$w_Q(i) = \begin{cases} 1 & if\ ||Q|| = 1 \\ \sum\limits_{j \in Q \setminus \{i\}} \Phi_{Q \setminus \{i\}}(j, i) w_{Q \setminus \{i\}}(j) & otherwise \end{cases} \quad (7)$$

The total weight of Q is defined to be $W(Q) = \sum_{i \in Q} w_Q(i)$. A non-empty sub-set of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq Q$, is said to be *dominant* iff:

1. $w_Q(i) > 0$, $\forall\ i \in S$, i.e. internal homogeneity

2. $w_{Q \bigcup \{i\}}(i) < 0 \; \forall \; i \notin S$, i.e. external inhomogeneity.

Optimization techniques such as *Replicator Dynamics* can be used to precipitate solving Equation 7 [8].

Appendix B.

Proof of Theorem 1: Here we prove:

1. $ms(t) < (t - i + 1) \Rightarrow h_a(t) = 1$
2. $h_a(t) = 1 \Rightarrow ms(t) < (t - i + 1)$

1. Suppose $ms(t) < (t - i + 1)$. Note that $a[i, t]$ is a suffix of $a[1, t]$. Since $ms(t)$ represents the longest suffix of $a[1, t]$ that is contained in S_c, if $a[i, t]$ is contained in S_c then $(t - i + 1) = ||a[i, t]|| \leq ms(t)$. From our supposition $ms(t) < (t - i + 1)$, therefore $a[i, t]$ cannot be contained S_c. Thus $h_a(t) = 1$.
2. To show that $h_a(t) = 1 \Rightarrow ms(t) < (t - i + 1)$ we will prove its contrapositive that $ms(t) \geq (t - i + 1) \Rightarrow h_a(t) = 0$. Suppose $ms(t) \geq (t - i + 1)$. Note that $a[i, t]$ is a suffix of $a[1, t]$. By the definition of $ms(t)$, $a[t - ms(t) + 1, t]$ is contained in S_c and is the longest such subsequence. Since $ms(t) \geq (t - i + 1)$, $a[i, t] = a[t - (t - i + 1) + 1, t] \subseteq a[t - ms(t) + 1, t]$. Since $a[i, t] \subseteq a[t - ms(t) + 1, t]$ and $a[t - ms(t) + 1, t] \in S_c$, $a[i, t] \in S_c$. Hence $h_a(t) = 0$.