Structure from Statistics - Unsupervised Activity Analysis using Suffix Trees

Raffay Hamid, Siddhartha Maddi, Aaron Bobick, Irfan Essa
College of Computing - Georgia Institute of Technology
{raffay, maddis, afb, irfan}@cc.gatech.edu

Abstract

Models of activity structure for unconstrained environments are generally not available a priori. Recent representational approaches to this end are limited by their computational complexity, and ability to capture activity structure only up to some fixed temporal scale. In this work, we propose Suffix Trees as an activity representation to efficiently extract structure of activities by analyzing their constituent event-subsequences over multiple temporal scales. We empirically compare Suffix Trees with some of the previous approaches in terms of feature cardinality, discriminative prowess, noise sensitivity and activity-class discovery. Finally, exploiting properties of Suffix Trees, we present a novel perspective on anomalous subsequences of activities, and propose an algorithm to detect them in linear-time. We present comparative results over experimental data, collected from a kitchen environment to demonstrate the competence of our proposed framework.

1. Introduction & Previous Work

Consider a household kitchen where activities such as frying eggs and washing dishes take place. Analysis of what is happening in such environments remains an important question, that will impact development of systems for automatic surveillance and scene understanding. Our goal in this paper is to introduce a novel representation to facilitate unsupervised analysis of activities in complex settings.

We start with the assumption that an environment consists of various key-objects, the interactions amongst which constitute different ***events*** (see Figure 1). Events may have strong dependence on their preceding events over multiple durations [14]. While entering an unlit room *e.g.*, a person generally turns the light on after opening the door. However, while washing dishes, the event of turning the faucet on, is usually followed by rinsing the dishes, followed by turning the faucet off. A temporal conjunction of such variable-length event subsequences constitutes an ***activity***.

A majority of previous approaches for activity representation require explicit modeling of activity structure [20] [13] [19]. Because such models are generally not at hand *a priori*, representations that can encode activity structure with minimal supervision are needed.

Figure 1. A top view of a kitchen floor with some of the labeled key-objects. A person interacts with the kitchen sink.

To this end, there has been some recent interest towards extracting activity structure simply by computing their local event-statistics (see *e.g.* [21] using Vector Space Model [18], [26] using Latent Semantic Analysis [1], and [4] using n-grams [11] respectively). Each of these representations offers a certain bias, and entails a unique feature space. The representational competence of such feature spaces is limited by their ability to capture activity structure only up to some fixed temporal resolution. Moreover, since discriminative prowess of such approaches is a function of the order over which event-statistics are computed, it comes at an exponential cost of computational complexity [2]. In this work, we address these issues by proposing the usage of Suffix Trees [12] to efficiently extract the variable length event-subsequence of an activity, constructing a more discriminative feature space, and resulting in potentially better activity-class discovery and classification.

Previous work on vision based anomaly detection has mostly focused on model-based anomaly recognition [7] [17]. However, for reasons of small sample size and large variations amongst anomalies, such approaches generally do not scale well [16]. Anomalous activities have recently been defined based on their dissimilarity from regular activities [21] [25]. Such approaches however analyze activities in a wholistic manner [26], not considering the potentially important local structural irregularities. We address this problem by exploiting properties of Suffix Trees in a linear-time algorithm for detecting anomalous event subsequences of activities, that can be shown to human observers for further analysis.

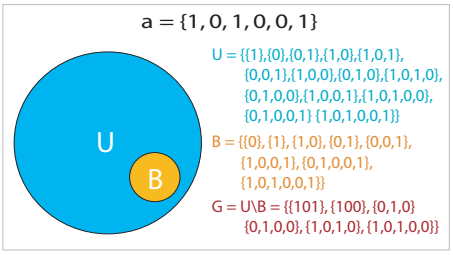


Figure 2. Illustration of Basis Subsequences - $\mathcal{U}$ represents the set of all contiguous subsequences in a, and $\mathcal{B}$ represents the set of basis subsequences of a. Every member of $\mathcal{G}$ occurs with the same frequency as some $b \in \mathcal{B}$ while being contained in b.

This paper improves upon the work in [5], which introduces Suffix Trees for activity analysis. Here we introduce a novel method to systematically control disjunction between classes while maintaining variable-duration symbol-dependence. We show how the feature-space of Suffix Trees is representationally equivalent to that of n-grams for all values of n. We empirically compare discriminative prowess and noise sensitivity of Suffix Trees. We highlight the significance of using a linear-cardinality feature-set to capture sequence-structure over a range of temporal scales. Finally, the notion of "minimal length anomalous subsequences" introduced here entails a novel formulation for detecting local structural anomalies.

2. Activity Representation

Activities are sequences of discrete events. As events can have dependence on preceding events over multiple durations [14], we model activities as temporal conjunctions of variable-length event subsequences. We are interested in representing an activity as counts of its constituent subsequences that subsume within themselves the rest of the subsequences of that activity. Consider *e.g.* the sequence:

$$a = \{x, y, z, p, q, r, x, y, z\} \tag{1}$$

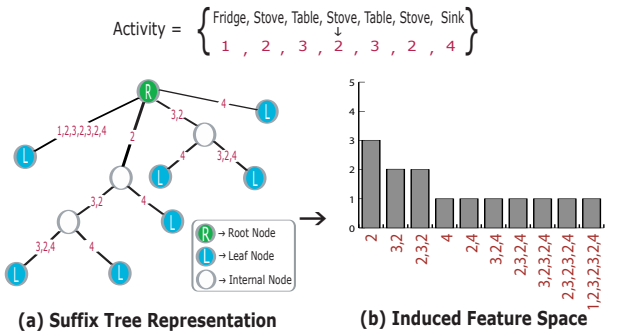
Note that subsequence x, y occurs with the same frequency as x, y, z, never occurring outside the context of x, y, z. In other words, x, y does not encode any extra structural information given the subsequence x, y, z and is therefore redundant from a representational perspective.

More formally, we represent a in terms of a subset $\mathcal{B}$ of its set of all subsequences $\mathcal{U}$, where $\forall\ w \in \mathcal{U}, \exists\ b \in \mathcal{B}$, *s.t.*:

- $f_a(w) = f_a(b)$
- w is a subsequence of b

where $f_a(w)$ stands for frequency of w in a. Any set $\mathcal{B}$, satisfying these properties is called a set of ***Basis Event Subsequences*** of a (see Figure 2 for an illustrative example). The aforementioned properties of $\mathcal{B}$ imply that $\forall\ w \in \mathcal{U} \setminus \mathcal{B}$,

$$f_a(w) = \max_{b \in \psi} f_a(b) \tag{2}$$



(a) Suffix Tree Representation **(b) Induced Feature Space**

Figure 3. Activity Representation - **(a)** Suffix Tree T for activity a, showing the trajectory of a person in a kitchen. **(b)** Activity a represented by counts of basis event subsequences of a, generated by traversing through the Suffix Tree T.

where $\psi \subset \mathcal{B}$, *s.t.* w is a subsequence of every member of ψ. Given $\mathcal{B}$, any subsequence $w \in \mathcal{U} \setminus \mathcal{B}$, does not provide any extra information about a, and is therefore redundant. Confining ourselves to subsequences with contiguous events, one way of extracting the basis event subsequences is to first enlist all of its constituent subsequences, followed by eliminating the ones that always appear as a prefix of any other subsequence. Such basis event subsequences encode the structural signature of an activity [6], and can therefore be used as discriminative features [10].

2.1. Treating Activities as Suffix Trees

Efficient extraction of the basis event subsequences of an activity is non-trivial, since their exhaustive search incurs exponential cost of computational complexity [3]. Drawing from previous work [5], we propose the usage of Suffix Tree T [12] as an activity representation to facilitate extraction of all basis event subsequences of a in time linear in $||a||$. T is a rooted, directed tree with each internal node having at least 2 children, and each edge labeled with a non-empty subsequence of a (see Figure 3-a). Starting from the root node, every basis subsequence in a can be generated by traversing through T, in time linear in $||a||$ [3]. A linear time algorithm for constructing T can be found in [23]. The feature space induced by T is spanned by the set of variable length basis event subsequences, $\mathcal{B}$, where $||b|| \in \mathcal{B}$ ranges over $[1 : ||a||]$, capturing the structure of a over multiple temporal scales (see Figure 3-b).

2.2. Representational Scope of Suffix Trees

Representations such as n-grams and Suffix Trees can be thought of as a means to extract different sequential features from a sequence. In this regard, two important questions emerge, *i.e.* how many features of an activity can a representation encode, and how succinct is this encoding.

We define the ***scope of a representation*** as the set of contiguous subsequences that can be extracted from a sequence using that representation. For instance, given a sequence a,

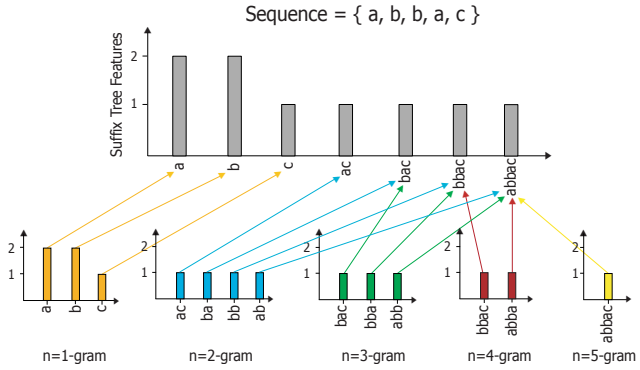


Figure 4. Representational Scope of Suffix Trees - Feature space induced by Suffix Trees embeds the feature spaces generated by $n = 1:5$-grams, implying greater representational scope of Suffix Trees than n-grams for any specific n.

the scope of 3-grams is the set of w, *s.t.* $\forall w \in a : ||w|| = 3$.

The occurrence of every $w \in a$ can be uniquely mapped to the minimum length $b \in \mathcal{B}$ induced by the Suffix Tree of a, which satisfies Equation 2 (for proof, see Appendix A). The space of all contiguous subsequences of a is spanned by n-grams where n ranges over $[1 : ||a||]$. Therefore, scope of Suffix Trees is equivalent to n-grams for all values of n, and greater than n-grams for a specific value of n (see Figure 4).

Furthermore, the number of nodes in T, and hence the cardinality of $\mathcal{B}$ is only linear in $||a||$ [23]. In contrast, it can be trivially shown that the upper bound on the cardinality of space of n-grams for all values of n is quadratic in $||a||$.

2.3. Empirical Analyses of Suffix Trees

We now present empirical analysis of representational competence of Suffix Trees using synthetic data. We compare their discriminative prowess, noise sensitivity and feature cardinality, with VSM, HMM's & n-grams.

Algorithm 1 Construct *VMMC's* $\mathcal{V}_1$ and $\mathcal{V}_2$

Require: Symbol vocabulary k, modal depth d, number of topological operations I, and $\%$ node perturbation η

Construct $\mathcal{V}_1$ as complete tree of depth d with leaf-set $\mathcal{S}$
Randomly construct $\mathcal{P} \subseteq \mathcal{S}$ where $||\mathcal{P}|| = ||\mathcal{S}||/2$
Construct $\mathcal{Q} \equiv \mathcal{S} \setminus \mathcal{P}$
for $i = 1$ to I **do**
  Sample a member of $\mathcal{Q}$. Detach it from its parent.
  Attach it to a randomly selected member of $\mathcal{Q}$.
end for
Sample edge probability of $\mathcal{V}_1$ from $\mathcal{N}(\mu, 1)$ distribution

Construct $\mathcal{V}_2$ as an exact copy of $\mathcal{V}_1$
Sample edge probability of $\eta\%$ nodes of $\mathcal{V}_2$ from $\mathcal{N}(\mu, 1)$

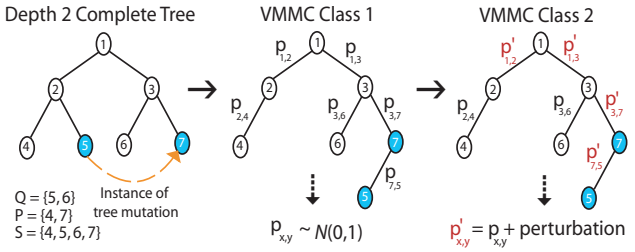


Figure 5. Illustration of Algorithm 1 - We begin by constructing a complete tree of depth d. $\mathcal{P}$ and $\mathcal{Q}$ are selected from leaf-set $\mathcal{S}$. Edge probabilities of VMMC-1 are sampled from $\mathcal{N}(0, 1)$. VMMC-2 is constructed by perturbing edge-probabilities of VMMC-1.

2.3.1 Systematic Control over Class-Disjunction

To simulate event dependence over a range of temporal durations, we model activity classes as Variable Memory Markov Chains (*VMMC*) [24], that can be encoded as probabilistic trees [3]. Since systematically controlling class disjunction while maintaining variable event dependence is non-trivial, here we outline a novel algorithm to this end.

We begin by constructing a complete tree T with depth equal to d. Randomly selecting half of the leaf-nodes of T, we iteratively attach them to its remaining half. The *VMMC* for class-1 is completed by assigning edge-probabilities of T by sampling from $\mathcal{N}(0, 1)$. *VMMC* for class-2 is constructed by first forming an exact copy of *VMMC* of class-1, followed by perturbing edge probabilities of top $\eta\%$ edge-paths of *VMMC* for class-1. The algorithm is outlined in Algorithm 1, and figuratively illustrated in Figure 5.

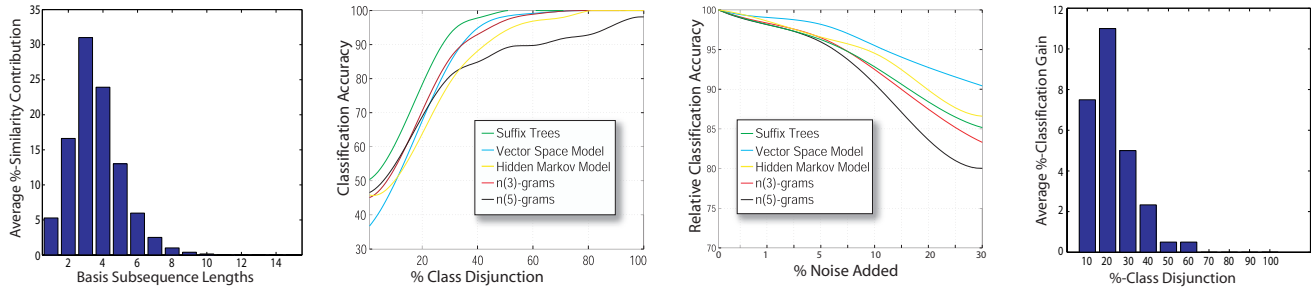
2.3.2 Similarity Contribution of Multi-length Features

Since classifiers are generally functions of activity-similarity, which in turn is a function of the values of different sequential features, it is imperative to analyze the degree to which sequential features of varying lengths are contributive to a similarity metric. Following [5], the similarity between sequences a and b can be partitioned with respect to lengths of basis subsequences as follows:

$$\text{sim}(a, b) = 1 - \mathcal{R} \sum_{k} \sum_{x_k \in H_a \cup H_b} \frac{|f(x_k|H_a) - f(x_k|H_b)|}{f(x_k|H_a) + f(x_k|H_b)} \quad (3)$$

where f represents frequency, H_a and H_b are sets of corresponding basis subsequences in a and b, and $\mathcal{R}$ is normalizing coefficient. x_k are basis-subsequences of lengths k.

Simulation Data: For a symbol vocabulary $||\Sigma|| = 5$ and modal depth equal to 3, we generated 10 different topologies of *VMMC's*. For each topology, we generated sequences for 2 classes with percent overlap decreasing from complete overlap to complete non-overlap with increments of 10%. For each of these 100 trials, we generated 75 sequences each of length 100, randomly selecting two-thirds for the training data and the rest for testing.



(a) (b) (c) (d)

Figure 6. a - Basis-Subsequence Similarity Contribution - The average percentage similarity contribution of basis subsequences of different lengths for all test points and their respective nearest neighbors. **b - Discriminative Prowess** - Classification accuracy as a function of class-overlap. **c - Noise Sensitivity**- Classification for various representations relative to their noise free performance. **d - Linear-Cardinality Feature Set** - Average percentage classification gain using basis sub-sequences versus all n-grams.

We compute the contribution of length-sorted basis subsequences to the similarity between all test points and their corresponding nearest neighbors (Figure 6-**a**). The wide range of lengths contributive towards similarity clearly highlights the importance of incorporating sequential features over multiple lengths.

Note that probabilistic trees of modal-depth 3 were used to generate simulation data, which is the most contributive length of basis-subsequences towards the similarity metric, suggesting that Suffix Trees can be used to infer the predominant scale of temporal dependence.

2.3.3 Discriminative Prowess of Suffix Trees

For data generated as described in § 2.3.2, and using the similarity metric defined in Equation 3, the nearest neighbor classification results are given in Figure 6-**b**.

It is evident that for substantive class overlap, higher values of n seem to capture activity structure more rigidly, entailing a more discriminative representation. However, since accurate density estimation for higher value n-grams require exponentially greater amount of data, Vector Space Model seems to outperform 3- and 5-grams in cases where the 2 classes are more disjunctive. Nevertheless, compared to different representations, Suffix Trees offer greater representational competence for any amount of class overlap.

2.3.4 Noise Sensitivity Analysis

We now analyze noise sensitivity of Suffix Trees as a function of noise added as *Insertion*, *Deletion*, *Transposition* and *Substitution* of symbols. For data generated as described in § 2.3.2, we cumulatively added all four types of noises with a uniform prior on each, and noise likelihood ranging monotonically from 0 to 30%. Using noisy data, the classification results for different representations relative to their noise free performance is given in Figure 6-**c**.

It is evident that representations that capture event order information more rigidly, are more sensitive to sensor noise, making VSM most robust to noise perturbations. Note that at noise levels $> 10\%$, Suffix Trees outperform both 3- and 5-grams. This is because while n-grams capture sequence structure at fixed temporal resolution, the scale at which Suffix Trees encode this structure varies inversely with sequence entropy [22]. As entropy is a function of noise, at higher noise levels, Suffix Trees emulate the VSM more closely, making them more robust to noise than n-grams.

2.3.5 Significance of Linear-Cardinality Feature-Set

One naive way of incorporating multi-length features might be to conjoin the feature spaces induced by n-grams for all values of n. Besides being computationally inefficient, such over-complete feature space contains extraneous information that may reflect in relatively poor classification performance. Suffix Trees on the other hand efficiently generate a linear-cardinality multi-length feature set to encode sequence structure, naturally filtering out extraneous information [24]. For data generated in § 2.3.2, Figure 6-**d** corroborates this in terms of average percent classification gain while using Suffix Trees over $n = [1 : N]$-grams.

3. Results: Class Discovery & Classification

Exploiting Suffix Trees, we now explore the questions of unsupervised activity-class discovery and classification. We first briefly outline activity-class discovery (see [5] for details). This is followed by experimental details and results.

Activity Class Discovery: Following [5], we consider an activity-set as an undirected edge-weighted graph, with each node representing the histogram of basis event subsequences for one of the activity-instances. The edge-weights are equal to the similarity between its pair of activity-nodes defined by Equation 3. We formalize the problem of discovering activity-classes as searching for edge-weighted maximal cliques in the activity-graph. We proceed by finding a maximal clique, 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 [5] [15].

3.1. Experimental Setup

It is argued that humans organize their surroundings to optimize execution of different activities [9]. This is particularly true for settings such as a household kitchen, where different key-objects provide a set of affordances, instrumental for completion of various activities. The order in which these objects are manipulated encodes the structural signature of activities, essential for activity analysis.

With this perspective at hand, we deployed a static camera in a kitchen ceiling to record a user's interaction with different key-objects known *a priori*. The floor layout of the kitchen along with the key-objects are shown in Figure 1. The user enacted 10 activity-classes each constituting of 10 activity instances. The directions and recipes for preparing dishes of different classes were taken from `http://www.recipeland.com/`.

These directions impose a set of temporal constraints, which require different events to be partially ordered in a particular manner. For instance, if a recipe uses chopped potatoes, then the events of washing the potatoes, getting a knife and chopping the potatoes must be performed before using the stove for cooking them. However, if a recipe uses chopped potatoes and onions, then the set of events to perform these two tasks may very well be interchanged. The locations where different ingredients were stored was maintained throughout the experiment, *e.g.* the silverware was kept in shelf 3, while spices were stored in shelf 2. At the end of each activity instance, every object used in performing the activity was placed back in its original location.

3.2. Automatic Event Detection

We assume the proximity of person with a particular key-object to imply an interaction between the person and the object. Each interaction longer than a particular duration was registered as an event of person interacting with a certain key-object. For this work, we implemented the tracking framework proposed in [8]. For extracting the person from background image, we learned *Gaussian Mixture Models* for the chromatic contents of the background, 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 image space for computing 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 the next iteration.

3.3. Performance Analysis of 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. Following [5], we considered two metrics for analyzing the performance of our approach regarding the goodness of each discovered cluster [11]:

$$\text{Precision (P)} = \frac{\text{Cardinality of majority population}}{\text{Discovered cardinality of cluster}} \quad (4)$$

$$\text{Recall (R)} = \frac{\text{Cardinality of majority population}}{\text{Actual cardinality of cluster}} \quad (5)$$

Note that activity-class discovery detailed in [5] only requires a certain notion of activity similarity, and is independent of the particular activity representation being used. This allows a modular approach to analyze the different activity representations in terms of unsupervised activity-class discovery. The performance of our method in comparison with *VSM*, 3- and 5-grams is shown in Table 1.

	Suffix Tree		**VSM**		**3-grams**		**5-grams**	
	P	*R*	*P*	*R*	*P*	*R*	*P*	*R*
Aloo Dam	55	60	55	50	50	60	55	60
Babka	50	40	-	-	56	50	38	30
Cereal	63	50	60	60	57	40	33	30
Fruit Salad	50	60	-	-	-	-	33	40
Omelet	63	70	-	-	-	-	-	-
Raita	47	70	-	-	18	70	33	70
Chicken	70	70	16	100	44	40	42	50
Setup Table	88	70	60	60	50	50	45	50
Green Salad	64	70	-	-	40	20	38	30
Wash Dishes	60	30	50	50	44	40	28	20
Average	**61**	**59**	**24**	**32**	**36**	**37**	**35**	**38**
% Discovery	**100**		**50**		**80**		**90**	

Table 1. Comparative performance for class discovery - Besides number of activity-classes discovered, Suffix Trees perform better than *VSM*, 3- and 5-grams based on Precision and Recall.

Since *VSM* does not capture activity structure rigidly, it produces a relatively fuller activity similarity matrix, resulting in the discovery of only 5 activity classes. As 3- and 5-grams capture activity structure in an increasingly more rigid fashion, they result in the extraction of 8 and 9 activity classes respectively. Capable of capturing activity structure at multiple temporal resolutions, Suffix Trees were able to discover all 10 activity-classes. The average precision and recall of the discovered activity-classes using Suffix Trees is also superior than other representations considered.

3.4. Classification Performance

For each activity-class discovered by Suffix Trees, two-thirds of class members were selected as training set while the remaining formed testing set. With 10 such constructed data sets, the average classification results using different representations[1] are shown in Table 2. The partially-ordered nature of activity sequences in kitchen domain dictates large within-class variation of activity-classes, which is reflected in the relatively modest average classification performance.

[1]For HMM representation, different number of states and Gaussian Mixture Models were tried to find the optimal parameter configuration.

	ST	3-gram	5-gram	VSM	HMMs
%-Accuracy	69	65	62	58	47

Table 2. Classification results using different activity representations - Average classification results seem to show that Suffix Trees perform better than other representations considered.

Nonetheless, the average classification performance of Suffix Trees outperforms rest of the representations considered.

4. Anomaly Detection

Using Suffix Trees for activity representation offers an interesting perspective on anomaly detection, which unlike previous approaches, calls for detecting event subsequences of an activity that have previously been unobserved in the membership class of that activity. This view is backed by the argument that since an activity can be executed in multiple legitimate ways, its regularity is independent of the number of times it is performed in any one of such ways.

Consider for example, the activity of making coffee. Assuming that a majority of people take coffee with cream, does not make this particular way of taking coffee any more regular than taking coffee without cream. With this perspective at hand, we argue that given a class of legitimate activity sequences A, any subsequence of events in a test sequence a classified as an instance of A, is regular so long as it occurred in A. Conversely, any subsequence of a is anomalous if it never appears in any of the previously observed members of A.

This approach exacts a more prudent use of training data, resulting in fewer false negatives. While the approach does not necessarily capture different semantic connotation of an anomaly, it is nevertheless useful in highlighting points of interest in an event-stream, leaving the final decision to the judgment of a human observer.

4.1. Defining an Anomaly

Given a test activity sequence a, classified as an instance of activity class A, let S be the set of all subsequences $\in A$. A subsequence $s \in a$ is said to be ***anomalous*** iff:

1. $s \notin S$, and
2. $\nexists\, s^{'}$, such that $s^{'} \subset s$ and $s^{'} \notin S$

Intuitively, we are interested in finding minimal length subsequences of an activity a that do not appear in any other previously observed activity in the membership class of a.

Since an exhaustive search for such previously unobserved minimal length anomalous subsequences is exponentially hard, we exploit the notion of *Match Statistics* of an activity a (explained below), to detect anomalous event subsequences of a in time linear in $||a||$ [3].

4.2. Anomalies Using Match Statistics

Let A and S be defined as before. Then following [3] we define the following two quantities:

Match Statistics: $\text{ms}(t)$ of a is the length of the longest subsequence in S that is a prefix of $a[t : ||a||]$.

Reverse Match Statistic: $\overline{\text{ms}}(t)$ of a is the length of the longest subsequence in S that is a suffix of $a[1 : t]$.

In other words, $a[t : (t + \text{ms}(t) - 1)]$ is the longest subsequence in a starting at index t that is contained in S. Similarly, $a[(t - \overline{\text{ms}}(t) + 1) : t]$ is the longest subsequence in a ending at index t that is contained in S. We now attempt to form the bridge between the *match statistics* and *reverse match statistics* of a to the anomalous subsequences of a.

Theorem 1: Let $s = a[i, j]$ for $1 \leq i < j \leq ||a||$. Then s is anomalous iff: (1) $\text{ms}(i) = j - i$ and, (2) $\overline{\text{ms}}(j) = j - i$ (for proof, see Appendix B).

Here $||s|| = j - i + 1$. Intuitively, a minimal length subsequence is not contained in training data if: (a) it is itself not contained in training set, (b) all its prefixes and (c) suffixes are present in training set. First condition of Theorem 1 implies criteria (a) and (b) are satisfied, while second condition guarantees criteria (b) and (c) are met. An algorithm to find anomalous event-subsequences is given in Algorithm 2. The *match* and *reverse match statistics* in Algorithm 2 can be computed in $O(||a||)$ given a Suffix Tree for the activity sequences of the membership class A [3].

Example Case: A figurative illustration of the notion of anomaly is shown in Figure 7. For $i = 1$, the longest subsequences in x that is a prefix of $y[1 : ||y||]$ is b. Therefore $\text{ms}(i = 1)$ of y is 1. Moreover, since the checking condition of Theorem 1 defines $j = \text{ms}(i) + i$, thus for $i = 1$, $j = 2$. As the longest subsequence in x that is a suffix of $y[1 : j = 2]$ is u, $\overline{\text{ms}}(j = 2)$ of y is 1. Since $\text{ms}(i = 1) = \overline{\text{ms}}(j = 2) = j - i = 1$, according to Algorithm 2, at $i = 1$, anomalyList(1) = $\{bu\}$. Similarly, anomalyList(2) = $\{bu, ua\}$.

Now consider the case for $i = 3$. As the longest subsequences in x that is a prefix of $y[3 : ||y||]$ is a, b, c, therefore $\text{ms}(i = 3)$ of y is 3. Similarly, $\overline{\text{ms}}(j = 6)$ of y is 1. Since

Algorithm 2 Find anomalous subsequences in activity a

```
Let anomalyList(0) = {∅}
for i = 1 to ||a|| do
    Let j = i + ms(i)
    if ms̄(j) = j − i then
        Add (i, j) to anomalyList(i)
    end if
end for
return anomalyList
```

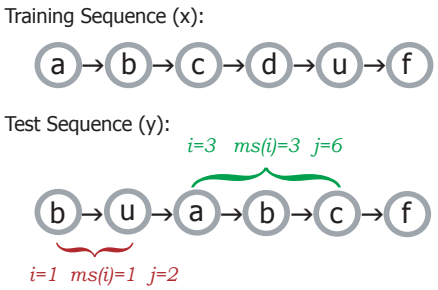


Figure 7. Notion of anomaly - Explanation of anomalous subsequences in terms of *match* and *reverse match statistics*.

$\text{ms}(i=3) \neq \overline{\text{ms}}(j=6) \neq j-i$, therefore, at $i=3$, anomalyList(3) = anomalyList(2) = $\{bu, ua\}$.

4.3. Performance of Anomaly Detection

A set of anomalous activities with access to ground-truth about the number of anomalies in each activity sequence and their respective locations, was constructed by randomly selecting 5 of the 10 activities for each of the 10 activity-classes, and modifying some of the steps of their recipes. Using this ground-truth information, the performance rates of our anomaly detection framework are given in Table 3. Being rather conservative towards the notion of regular, our framework is able to correctly extract 84% of the true anomalies. This naturally comes at the cost of relatively high false positive rate of 47%, which can be addressed by using event detectors more robust to sensor noise.

4.4. Analysis of Detected Anomalies

An analysis of detected anomalies reveals that anomalies involving key-objects that serve a unique purpose are detected almost perfectly, while those involving objects with overloaded affordances can be missed by our approach. For instance, the key-objects sink and stove each offer only one affordance. Therefore anomalies where the person forgets to wash something before cutting it, or pre-heating the stove at the beginning of cooking, are always detected. However, the anomaly of forgetting to add salt for example, kept in shelf 3 might be missed if the person went to shelf 3 earlier to get some other condiment. This underscores the importance to define events at a level sufficient to describe different types of anomalies that can occur in an environment.

5. Discussion & Future Work

In this work, we present a novel representation for unsupervised activity analysis in sensor-rich environments. We specifically investigate if there is sufficient structural signature at a local temporal scale that can entail a reasonably disjunctive partitioning of the activity space. This requires a notion of the granularity of scale at which events should be analyzed. Lower granularity results in more discriminative characterizations, and would therefore be prone to sensor noise. Higher granularity would 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 the usage of Suffix Trees all the more appropriate.

Class Labels	Added Anomalies	Total Detected	Correctly Detected	% True +	% False +
C1	15	25	13	87	48
C2	7	18	6	86	61
C3	10	16	9	90	44
C4	16	27	13	81	52
C5	12	14	8	67	43
C6	20	15	12	60	20
C7	16	24	13	81	46
C8	5	12	5	100	58
C9	11	25	11	100	56
C10	17	26	15	88	42
Average	-	-	-	**84 %**	**47 %**

Table 3. Anomaly Detection Performance - Column 2 shows ground-truth information about number of anomalies added per-class. Column 3 represents total number of anomalies per-class detected. Number of true anomalies detected are given in column 4. % true and false positive rates are listed in column 5 and 6.

The considerably large space of legitimate ways in which an activity can be performed, requires a relatively conservative notion of activity regularity. By highlighting event subsequences not observed in the training set, our approach helps reduce the false negative rate, and leaves the final decision to a human expert. Since Suffix Trees allow one to enumerate such structurally atypical event subsequences in linear time, they are appropriate not just for activity-class discovery and classification, but also for anomaly detection.

In our future work, we intend to incorporate the duration for which a person interacts with a key-object in the environment. Moreover, we are interested in exploiting multiple sensor-modalities so we could use a richer event vocabulary.

6. Acknowledgements

We would like to thank Spencer Brubaker, Sooraj Bhat, Matthew Mullin and Zsolt Kira for many insightful discussions. We also acknowledge the support of DARPA contract number SUBC 03-000214, as a part of CALO project.

Appendix A. Representational Scope

Given a finite length sequence a, such that $||a|| = N$, let $\mathcal{F}$ be the set of all contiguous subsequences of a. $\mathcal{F}$ is spanned by n-grams of a, where n ranges from $[1:N]$. If $\mathcal{F}_x$ is the space spanned by x-gram, then

$$\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \cdots \cup \mathcal{F}_N \tag{6}$$

Claim: Suffix Trees induce a unique surjective mapping $\mathcal{M} : \mathcal{F} \rightarrow \mathcal{B}$.

Proof: We first prove $\exists$ a unique mapping $\mathcal{M} : \mathcal{F} \rightarrow \mathcal{B}$ induced by Suffix Trees, and then show that $\mathcal{M}$ is surjective.

Equation 2 implies that for any $w \in \mathcal{F}$, $\exists\ \psi \subset \mathcal{B}$, *s.t.* $\forall \psi_i \in \psi$, $f(w) = f(\psi_i)$, and $w \subseteq \psi_i$. By construction, for the particular $\mathcal{B}$ induced by Suffix Tree, not only is $w \subseteq \psi_i$, but also w is a prefix of $\psi_i\ \forall \psi_i \in \psi$. Moreover, by construction each $\psi_i \in \psi$ is unique. The previous two statements imply that each $\psi_i \in \psi$ is of unique length. Thus $\forall w \in \mathcal{F}$, $\exists$ a unique mapping $\mathcal{M} : \mathcal{F} \rightarrow \mathcal{B}$ such that **i:** $f(w) = f(b)$ for some $b \in \mathcal{B}$, **ii:** $w \subseteq b$, and **iii:** $\nexists$ any $\overline{b} \in \mathcal{B}$ with $w \subseteq \overline{b}$ and $||\overline{b}|| < ||b||$.

As $\mathcal{B} \subseteq \mathcal{F}$, each b is contained in $\mathcal{F}$ and is mapped to itself from $\mathcal{F} \rightarrow \mathcal{B}$. Moreover, $||\mathcal{B}|| \leq ||\mathcal{F}||$. Therefore, the unique mapping $\mathcal{M} : \mathcal{F} \rightarrow \mathcal{B}$, is surjective. Q.E.D.

Appendix B. Proof Theorem 1

($\Longrightarrow$) Suppose s is an anomalous subsequence, *i.e.* $a[i,j] \notin S$. Then, $\text{ms}(i) < j - i + 1$, otherwise $a[i,j]$ would be contained in S. Similarly, $\overline{\text{ms}}(j) < j - i + 1$.
Suppose for a contradiction, that $\text{ms}(i) < j - i$. Since $\text{ms}(i) < j - i$, $a[i, i + \text{ms}(i) - 1] \subset a[i, i + (j - i) - 1] = a[i, j - 1]$ and by definition, $a[i, i + \text{ms}(i) - 1]$ is the longest substring in a starting at i that is contained in S, $a[i, j - 1]$ is not contained in S. But since $a[i,j]$ is an anomalous substring, $a[i,j]$ does not contain any substring that is not in S. This however contradicts the fact that $a[i, j-1] \subset a[i,j]$ and $a[i, j-1] \notin S$. Hence $\text{ms}(i) \geq j - i$. Since we have shown that $\text{ms}(i) < j - i + 1$, therefore $\text{ms}(i) = j - i$. Similarly, we can show that $\overline{\text{ms}}(j) = j - i$.

($\Longleftarrow$) Suppose $\text{ms}(i) = j - i$ and $\overline{\text{ms}}(j) = j - i$. We prove $a[i,j]$ satisfies both conditions of being anomalous:

1. By definition, $a[i, i + \text{ms}(i) - 1] = a[i, i + (j - i) - 1] = a[i, j - 1]$ is the longest subsequence of a starting at i that is contained in S. Hence $a[i,j] \notin S$.
2. By definition of match statistic, $a[i, i + \text{ms}(i) - 1] = a[i, i + (j - i) - 1] = a[i, j - 1] \in S$. Hence all subsequences of $a[i, j - 1] \in S$. Similarly, since $a[j - \overline{\text{ms}}(j) + 1, j] = a[j - (j - i) + 1), j] = a[i + 1, j] \in S$, all subsequences of $a[i + 1, j] \in S$. Since all subsequences of $a[i, j - 1]$ and of $a[i + 1, j] \in S$, therefore all proper subsequences of $a[i,j]$ also $\in S$.

Q.E.D.

References

[1] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *JASIS*, 41:391–407, 1990.

[2] W. E. Grimson. The combinatorics of local constraints in model-based recognition and localization from sparse data. *Journal of the ACM*, 33(4):658–686, 1986.

[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. Bobick, and I. Essa. Unsupervised analysis of activity sequences using event-motifs. In *VSSN '06*, 2006.

[6] M. Hammouda and M. S. Kamel. Efficient phrase-based document indexing for web document clustering. *IEEE Trans. on KDE*, 16(10):1279–1296, 2004.

[7] S. Hongeng and R. Nevatia. Multi-agent event recognition. In *In Proc. of IEEE ICCV*, 2001.

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

[9] D. Kirsh. The intelligent use of space. *Journal of Artificial Intelligence*, 73, 1995.

[10] C. Largeron. Prediction suffix trees for supervised classification of sequences. *Pattern Recognition Letters*, 2003.

[11] C. Manning and H. Schtze. *Foundations of Statistical Natural Language Processing*. 1999.

[12] E. McCreight. A space-economical suffix tree construction algorithm. *Journal of the ACM*, pages 262–272, 1976.

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

[14] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *IEEE ICMI*, 2002.

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

[16] D. Pelleg and A. W. Moore. Active learning for anomaly and rare-category detection. In *NIPS*, 2005.

[17] R. Russo and M. Shah. A computer vision system for monitoring production of fast food. In *ACCV*, 2002.

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

[19] Y. Shi, A. Bobick, and I. Essa. Learning temporal sequence model from partially labeled data. In *IEEE CVPR*, 2006.

[20] 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.

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

[22] W. Szpankowski. Unexpected behavior of typical suffix trees. In *Proc. of 3rd ACM-SIAM*, 1992.

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

[24] M. Weinberger, J. Rissanen, and M. Feder. A universal finite memory source. In *IEEE Trans. Inform. Theory, vol. IT-41, pp. 643–652, 48*, 1995.

[25] L. Zelnik-Manor and M. Irani. Event-based video analysis. In *Proc. of IEEE CVPR*, 2001.

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