A Novel Sequence Representation for Unsupervised Analysis of Human Activities

Raffay Hamid, Siddhartha Maddi, Amos Johnson, Aaron Bobick, Irfan Essa, Charles Isbell

College of Computing, Georgia Institute of Technology - Atlanta, GA, USA
{*raffay, maddis, amos, afb, irfan, isbell*}*@cc.gatech.edu*

Abstract

Formalizing computational models for everyday human activities remains an open challenge. Many previous approaches towards this end assume prior knowledge about the structure of activities, using which explicitly defined models are learned in a completely supervised manner. For a majority of everyday environments however, the structure of the *in situ* activities is generally not known *a priori*. In this paper we investigate knowledge representations and manipulation techniques that facilitate learning of human activities in a minimally supervised manner. The key contribution of this work is the idea that global structural information of human activities can be encoded using a subset of their local event subsequences, and that this encoding is sufficient for activity-class discovery and classification.

In particular, we investigate modeling activity sequences in terms of their constituent subsequences that we call event n-grams. Exploiting this representation, we propose a computational framework to automatically discover the various activity-classes taking place in an environment. We model these activity-classes as maximally similar activity-cliques in a completely connected graph of activities, and describe how to discover them efficiently. Moreover, we propose methods for finding characterizations of these discovered classes from a holistic as well as a by-parts perspective. Using such characterizations, we present a method to classify a new activity to one of the discovered activity-classes, and to automatically detect whether it is anomalous with respect to the general characteristics of its membership class. Our results show the efficacy of our approach in a variety of everyday environments.

Key words: Temporal Reasoning; Scene Analysis; Computer Vision.

1 Introduction

Consider a household kitchen where different activities, such as making omelets, washing dishes, or eating cereal *etc.*, can take place. Each one of these activities can be performed in many different ways. To build systems that can be

proactive and assistive in such environments, it is not plausible to learn each and every one of the *in situ* activities in a completely supervised manner. We are therefore interested in knowledge representations and manipulation techniques that allow computational systems to analyze human activities with minimal supervision.

The importance of these systems that can learn our everyday activities can be motivated by the variety of applications that they promise to offer. For instance, they have the potential to help us monitor peoples' health as they age, as well as in fighting crime through improved surveillance. Their medical applications include identifying and evaluating crucial parts of surgical procedures, and providing surgeons with useful feedback. Similarly, they can help us improve our productivity in office environments by detecting important events around us to enhance our involvement in various tasks.

One of the key challenges in building such perceptual systems is the big gap that exists between the low level sensory inputs such as pixel values or microphone voltages, and higher level inferences such as what dish is being prepared in a kitchen, or whether someone forgot to add salt in it *etc.* A natural way to bridge this gap is to have a set of intermediate characterizations that can appropriately channel the low-level perceptual information all the way to higher level inference stage. The granularity at which these intermediate characterizations should be defined presents a trade-off between how expressive the characterizations are, versus the robustness with which they can be detected through low-level sensory data. In the following, we define a set of such intermediate characterizations that we shall use throughout this paper.

1.1 Elements of Activity Dynamics

One way of looking at everyday environments is in terms of a set of perceptually detectable key-objects [22]. A key-object may be defined as:

Key-object: An object present in an environment that provides functionalities that may be required for the execution of activities of interest in that environment.

We assume that a list of key-objects for an environment is known *a priori*. An illustrative figure showing a list of key-objects in a kitchen environment is shown in Figure 1. Various operations on the key-objects can be used to define a set of perceptually detectable activity-descriptors. We call these descriptors Events which are defined as:

Event: A particular interaction among a subset of key-objects over a finite duration of time.

Figure 1 shows an example event of a person washing utensils in a sink.

Fig. 1. **Illustration of an Example Event** - A person shown washing some dishes in the sink of a kitchen.

Event Vocabulary: The set of interesting events that can take place in an environment.

An event vocabulary for a household kitchen may consist of events like person opens the fridge door, person turns the stove on, person turns the faucet on, *etc.* We assume that such an event vocabulary is known *a priori*.

Activity: A finite sequence of events.

To illustrate the notion of activities in an everyday environment, an example activity of making scrambled eggs is described below:

`Make Scrambled Eggs` = `Enter Kitchen` → `Turn Stove On` → `Get Eggs` → `Fry Eggs` → `Turn Stove Off` → `Leave Kitchen`

We assume that the start and end events of activities are known *a priori*, and that every activity must be finished before another is started, *i.e.* the question of overlapping activities is not included in our problem domain.

1.2 Main Hypothesis

We want to learn everyday human activities using some activity representation that does not require us to manually encode the structural information of these activities in a completely supervised manner. By structural information of an activity, we mean the various events constituting that activity, and the temporal order in which these constituent events are arranged. Our approach to this challenge is based on our hypothesis that we can learn the global structure of activities simply by using their local event subsequences. In particular, our main hypothesis states:

Hypothesis Statement: "*The structure of activities can be encoded using a subset of their contiguous event subsequences, and this encoding is sufficient for activity discovery and recognition*".

At the heart of our hypothesis is the question whether we can have an ap-

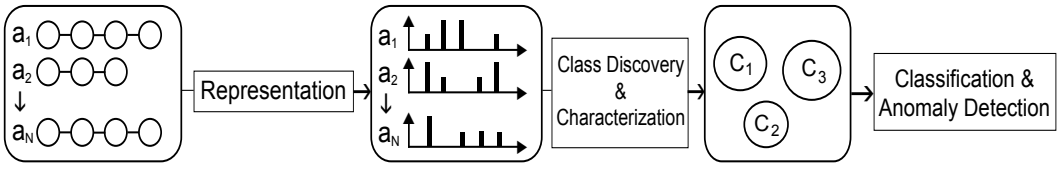


Fig. 2. **General Framework - 1**- Starting with a corpus of activities, we extract their contiguous subsequences using some activity representation. **2**- Based on the frequential information of these subsequences, we define a notion of activity similarity and use it to automatically discover different activity-classes. **3**- We characterize the discovered classes both at holistic and by-parts levels. **4**- We classify a test activity to one of the discovered classes, and compare it to the previous members of its membership class in order to detect anomalies.

propriately descriptive yet robustly detectable event vocabulary to describe human activities in a variety of everyday environments. Such intermediate sets of characterizations have been previously shown to exist for representing various temporal processes including speech [32], text documents [36], and protein sequences [4].

We posit that the key-objects in everyday environments pose a set of spatial and temporal constraints on the way we generally execute our activities in these environments [22]. For instance, one has to open a fridge before one can get milk out of it. Similarly, one must turn a stove on before one can use it to fry eggs, *etc.* We believe that these constraints can be used to construct a set of robustly detectable events that can appropriately describe the various activities taking place in an environment. These events can channel the low-level information detected from the sensors, in a manner that facilitates making useful higher-level inferences. This idea of learning activity structure by using statistics of their local event subsequences is essential to move us away from the traditional grammar driven approaches for activity modeling, and adopt a more data-driven perspective.

1.3 Key Contributions

The main contribution of this work is a data-driven perspective towards activity analysis. We view this approach towards automatic analysis of human activities in four principled ways:

1- Representation of activities in terms of their local event subsequences
2- Discovery of the various activity-classes in an environment
3- Characterization of the discovered activity-classes, and
4- Detection of activities that deviate from characteristics of discovered classes

A brief description of these main contributions follows. A block diagram illustrating the general overview of our proposed framework is given in Figure 2.

1.3.1 Activity Representation

We propose a novel activity representation that considers activities in terms of their contiguous event subsequences of some fixed length. In particular, we consider activities as histograms of their event n-grams, where an n-gram is a contiguous activity subsequence of length n.

1.3.2 Activity-Class Discovery

Exploiting our activity representation, we propose a computational framework to automatically discover the various activity-classes taking place in an environment. We model activity-classes as maximally similar activity-cliques in a completely connected graph of activities, and show how to discover them efficiently.

1.3.3 Activity-Class Characterization

Finding characterizations of the discovered activity-classes is imperative for online activity classification as well as anomaly detection. In this regard, we propose methods for finding concise characterizations of these discovered activity-classes, both from a holistic as well as a by-parts perspective. From a holistic view, we formalize the problem as finding typical members of activity-classes that, to some measure, best represent all the members of the activity-class. On a by-parts level, we consider this problem as that of finding recurrent event subsequences in the member activities of an activity-class. We call these recurrent event subsequences *event motifs* (formally defined in Section 7.1, and find them in a way such that they are maximally mutually exclusive amongst the various activity-classes.

1.3.4 Activity Classification & Anomalous Activity Detection:

Using such characterizations, we present a method to classify a new activity instance to one of the discovered activity-classes, and to automatically detect if it is anomalous with respect to the general characteristics of its membership class. We also present an information theoretic method to explain the detected anomalies in a maximally informative manner.

1.4 Document Layout

This paper is a detailed exposition and extension of some of our preliminary work in [15] and [16]. We start in Section 2 by reviewing the previous work related to the problem at hand, pointing out how our approach is different from the previously proposed methods. We explain in Section 3 our proposed activity representation of event n-grams, and present an empirical analysis of their discriminative power and sensitivity to sensor noise as a function of class overlap. Exploiting event n-grams, in Section 4 we show how the notion of maximal cliques in edge-weighted activity-graphs can be used to efficiently discover activity-classes in an unsupervised manner. In Section 5, we explain

how these discovered activity-classes can be used for activity classification, anomalous activity detection as well as their explanation. Section 6 explains the experimental results for our proposed framework. The characterization of the discovered activity-classes for the purposes of online activity classification and anomaly detection is presented in Section 7. Section 8 explains the results for our proposed framework for event motif discovery. The conclusions and future directions of this work are explained in Section 9.

2 Related Work

The problem of automatic human activity analysis has been studied in various contexts, including computational perception [6], ubiquitous computing [11], as well as robotics [40]. Much has been written about activity decomposition and the role of knowledge in the perception of motion [5], where scientists have worked on understanding the psychological [43] as well as computational basis of how motion is perceived. [46] [44]. In the following we briefly review some of the previous work done in the scope of perceptual scene analysis, comparing how our work differs from these previous approaches.

2.1 Activity Representation

One of the key problems in building perceptual systems is finding activity representations that are efficiently computable. Most of the previous approaches towards this end assume that the structure of activities being modeled is known *a priori* (see *e.g.* [20] [25] [38] [26] [7] [24] and the references therein). However, such prior knowledge about activity structure is generally not at hand. These grammar driven modeling approaches are therefore limited to representing activities performed in relatively small-scale constrained environments, underscoring the motivation of our current work. Here we propose to treat activities as bags of event n-grams to extract their global structural information by using statistics of their local event subsequences.

2.2 Activity-Class Discovery

Discovering activity-classes using perceptual data has been explored in depth in the past. Our approach towards this problem is however novel in a few key aspects. The work in [13] and [31] for instance looks directly at perceptual signals to discover coherent classes of behaviors. Our work on the other hand adds an intermediate abstraction layer of events upon which the discovery process takes place. Since events are semantically more meaningful than the direct sensory signals, the activity-classes discovered based on events would potentially be more coherent and easily interpretable. Since event-monograms, as used in [47] and [39], do not capture the temporal information of activities, we propose to use higher order event n-grams. While work in [42] [29] has similar motivation of finding event patterns between activity sequences,

our framework goes beyond finding similarities between activities, and also addresses problems of class characterization as well as anomaly detection.

Unlike previous approaches, our framework models activity-classes as edge-weighted maximal cliques in a completely connected graph of some given activity-instances. Finding maximal cliques in edge-weighted graphs is a classic graph theoretic problem [2] [33]. In this paper we adopt the recently proposed approximate approach of iteratively finding dominant sets of maximally similar nodes in a graph (equivalent to finding maximal cliques) [30]. Besides providing an efficient approximation to finding maximal cliques, the framework of dominant sets naturally provides a principled measure of the cohesiveness of a class as well as a measure of node participation in its membership class.

2.3 Anomaly Detection

Most of the previous attempts to tackle the problem of finding activities that are anomalous have focused on a recognition based perspective towards the problem, where anomalous activities are explicitly modeled and learned in a supervised manner [18] [19]. For large-scale everyday environments however, anomalies are hard to completely define *a priori*. Rather than modeling anomalies themselves, in this work we propose to model the regular activity-classes and detect anomalous activities based on their distance from learned models of regular behaviors in the environment. Previous works that have similarly taken a detection based perspective towards finding anomalies [8] [31] have looked at it mostly from a generative perspective, and have not attempted to explain why an activity being detected as anomalous is in fact an anomaly. In contrast, our approach takes an instance-based view of activity-classes, and attempts to detect as well as explain in a maximally informative manner, why an activity is detected as an anomaly.

2.4 Activity-Class Characterization

A concise characterization of discovered activity-classes is imperative, both from a representational as well as a discriminative perspective. This is particularly important in situations where the start and end of different activities is not explicitly marked, and there is a need to perform online classification and anomaly detection. While previously proposed instance-based approaches in this regard [23] [39] focus on the representational aspects of the problem, they are not necessarily discriminative. Moreover, these approaches only consider activities at a global scale, not incorporating the more local information. To this end, we formalize this problem as finding predictably recurrent event motifs using variable memory Markov chains.

Numerous solutions to the problem of discovering important recurrent motifs in the fields of Bioinformatics and String Analysis have been previously proposed (see *e.g.* [27] [4] [9] and the references therein). Work done in [45] and [34]

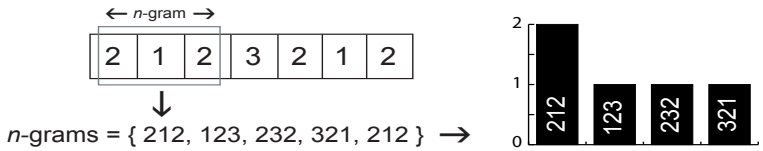


Fig. 3. **Illustration of n-grams -** Transformation of an example activity from sequence of events to histogram of event n-grams. Here the value of n is shown to be equal to 3.

presents techniques for learning variable memory Markov chains from training data in an unsupervised manner. Here, we extend the work done in [45] to handle data from multiple classes, finding motifs that are maximally mutually exclusive amongst activity-classes. Instead of sequentially finding individual subsequences and masking them out from the sequences as proposed in [3], our scheme simultaneously finds all the subsequences in the data in one pass, allowing to find partially overlapping subsequences.

3 Activity Representation - Activities as Bags of Event n-grams

Since models of activity structure for relatively unconstrained environments are generally not available *a priori* [10], representations that can encode this structure with minimal supervision are needed. Considering an activity as a sequence of discrete events [1] , two important quantities emerge:

1- Content - which events are constituting an activity, and
2- Order - the temporal arrangement of the constituent events.

We want to learn the content and order information of activities using an activity representation that does not require us to manually encode this information in a completely supervised manner.

Our view of an activity is similar to how researchers in Natural Language Processing have looked at documents, *i.e.* as vectors of their constituent words (see Vector Space Model (VSM) [36]). While approaches such as VSM capture the content of a sequence in an efficient way, they completely ignore its order. Since the word content alone in documents often implies causal structure, ignoring order information of words is usually not a significant challenge. Activities however are not fully defined by their event-content alone; rather, there are preferred or typical event-orderings [27]. Therefore a model to capture event order in a more explicit manner is needed.

To this end we consider activities in terms of histograms of event n-grams where an n-gram is a contiguous subsequence of an activity. Each event n-gram is of a fixed size n. By sliding a window of length n over an activity, we can find all the event n-grams contained in it. We can then represent that

[1] Recall that we have defined an activity as a finite sequence of discrete events.

activity as counts of these extracted n-grams. For the illustrative example shown in Figure 3, the value of n is set equal to 3.

It is evident that higher values of n capture order information of events more precisely. However, as n increases, the dimensionality of the histogram space grows exponentially. For instance, given an event vocabulary of k events, n-grams with $n = 5$ would span an activity space with k^5 dimensions. For even moderate values of k, density estimation in such a space can be challenging. This highlights the importance of selecting a reasonable value of n which sufficiently captures event dependence in an environment, and yet induces a space that can be estimated from reasonable amounts of data.

3.1 Empirical Analyses of n-grams using Simulation Data

Representations such as n-grams can be thought of as a means to extract different sequential features from an activity sequence. It is essential to analyze how well can such a feature space discern between members of different classes with respect to some ground-truth notion of class-overlap. Moreover, since for any sensor-based perceptual system, the observations are always prone to sensor-noise, the efficacy of a representation is a function of how sensitive it is to sensor-noise. With this perspective at hand, we now present empirical analyses of n-grams in terms of their discriminative power and noise sensitivity as a function of class-disjunction and noise perturbation. The analyses presented here are based on simulated data, the details of which follow.

Events in human activities depend on preceding events over variable durations [28]. To simulate this variable length event dependence, we model activity-classes as variable memory Markov chains ($VMMC$) [45]. One way of encoding such a $VMMC$ is by using a probabilistic tree [14], where each node represents any one of the members of the event vocabulary, while each edge represents the probability of traversing to its child from its parent. The topology of a tree encodes the variable temporal dependence between different events. Given two identical trees, the sequences generated from them would have same statistical properties. However, as we increasingly perturb their edge probabilities, the resulting sequences generated would have increasingly different event statistics. Using this behavior to model the disjunction between the sequences of a pair of activity-classes, we first outline a novel algorithm regarding how to systematically control disjunction between activity-classes.

3.1.1 A Novel Method To Systematically Control Class Disjunction

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 a normal distribution with zero mean and unit variance ($\mathcal{N}(0, 1)$). $VMMC$ 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 4.

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 $\mathcal{N}(\mu, 1)$

3.1.2 Simulation Data:

For a symbol vocabulary $||\Sigma|| = 5$ and modal depth equal to 3, we generated 10 different topologies of *VMMCs*. 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 symbols, randomly selecting two-thirds for the training data and the rest for testing.

3.1.3 Discriminability Analysis

For data generated as described in § 3.1.2, and using similarity metric defined later (Equation 1), the nearest neighbor classification results are given in Figure 5-**a**. It is evident that for substantive class overlap, higher values of

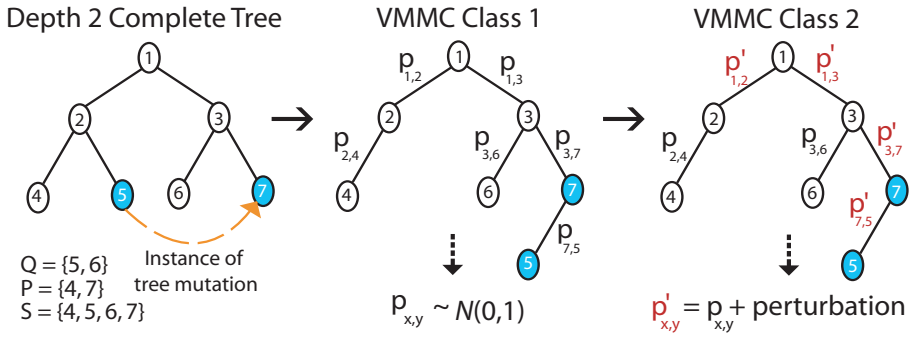
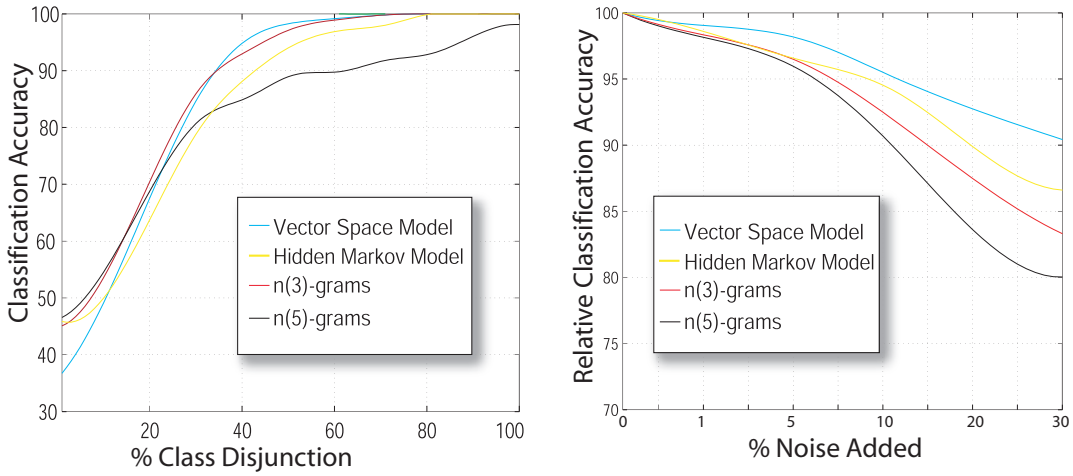


Fig. 4. **Illustration of Algorithm 1** - We begin by constructing a complete tree of depth d. $\mathcal{P}$ and $\mathcal{Q}$ are selected from leaf-set $\mathcal{S}$. Probabilities of VMMC-1 are sampled from $\mathcal{N}(0, 1)$. VMMC-2 is constructed by perturbing probabilities of VMMC-1.



(a) (b)

Fig. 5. **a - Discriminative Prowess** - Classification accuracy as a function of class-overlap. **b - Noise Sensitivity**- Classification for various representations relative to their noise free performance.

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.

3.1.4 Noise Sensitivity Analysis

We now analyze noise sensitivity of n-grams as a function of noise added as *Insertion*, *Deletion*, *Transposition* and *Substitution* of symbols. For data generated as described in § 3.1.2, 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 5-**b**. It is evident that representations that capture event order information more rigidly, are more sensitive to sensor noise. This underlines an inherent tradeoff between the ability of a representation to explicitly capture sequence-structure, and its robustness to sensor noise. It seems that tri-grams ($n = 3$) provide a reasonable balance between the two opposing factors. This is particularly true for relatively small class-overlap.

4 Unsupervised Activity-Class Discovery

We want to use the activity representation of event n-grams to automatically discover the various categories of human behaviors taking place in an environment. We assume that members of an activity-class generally share a set of

common properties that make them perceptually similar to each other, while making them different from members of other activity-classes. In order to discover such internally cohesive and externally disjunctive activity-classes, we first need to define some notion of activity similarity based on which we could formalize a method for activity-class discovery.

4.1 Activity Similarity Metric

Due to the spatial and temporal constraints imposed by the key-objects in an environment, human activities tend to have partially ordered sequences of events. Our desired notion of similarity between activities should consider this partially ordered nature of activities, and we want to use the representation of event n-grams as a means to this end. In particular, our view of similarity between a pair of activity sequences consists of two factors:

1- The structural differences, and
2- The frequential differences

The structural differences relate to the distinct n-grams that occurred in either one of the activities in an activity-pair, but not in both. For such differences, the number of mutually exclusive n-grams is of fundamental interest. Similarly, if a particular n-gram is present in both the sequences, the only discrimination that can be drawn between the sequence-pair is purely based on the frequency of the occurrence of that n-gram. This intuition can be formalized as follows.

Let A and B denote two activities, and let their corresponding histograms of event n-grams be denoted by H_A and H_B. Let Y and Z be the sets of indices of n-grams with counts greater than zero in H_A and H_B respectively. Let α_i denote different n-grams, and $f(\alpha_i|H_A)$ and $f(\alpha_i|H_B)$ denote the counts of α_i in A and B respectively. We define similarity between two activities as:

$$\text{sim(A,B)} = 1 - \kappa \sum_{i \in Y,Z} \frac{|f(\alpha_i|H_A) \; - \; f(\alpha_i|H_B)|}{f(\alpha_i|H_A) \; + \; f(\alpha_i|H_B)} \tag{1}$$

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

4.2 Activity-Class Discovery

It is argued that while facing a new piece of information, humans first classify it into an existing class [35], and then compare it to the previous class members to understand how it varies in relation to the general characteristics of the membership class [37]. Using this perspective as our motivation, we represent

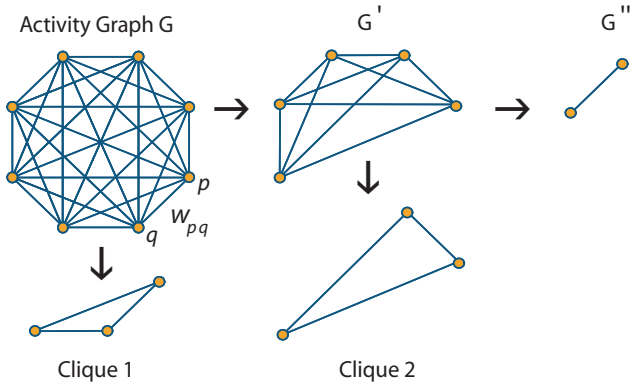


Fig. 6. **Illustration of Activity-Class Discovery -** Activity-instances are represented as a completed connected, edge-weighted activity graphs G. The edge-weight $w_{p,q}$ between nodes p and q is computed using Equation 1. Maximal cliques of activity-nodes are iteratively found and removed from the activity-graph, until there remain no non-trivial maximal cliques. These maximal cliques correspond to activity-classes comprising of mutually similar activity instances.

an activity space by a set of mutually disjunctive classes, and then detect a new activity as a regular or an anomalous member of its membership class.

4.2.1 Activity-Classes as Maximal Cliques

Given K activity-instances, we consider this activity-set as an undirected edge-weighted graph with K nodes, each representing a histogram of n-grams of one of the K activity-instances. The weight of an edge is the similarity between a pair of nodes as defined in Equation 1. We can now formalize the problem of discovering activity-classes as searching for edge-weighted maximal cliques [2] in the graph of K activity-instances [2]. We begin by finding the first maximal clique in the activity-graph, followed by removing that set of nodes from the graph, and iteratively repeating this process with the remaining set of nodes, until there remain no maximal cliques in the graph. The leftover nodes after the removal of maximal cliques are dissimilar from most of the regular nodes, and are considered as being anomalous (see Figure 6 for illustration).

4.2.2 Maximal Cliques using Dominant Sets

As combinatorially searching for maximal cliques in an edge-weighted undirected graph is computationally hard, numerous approximations to the solution of this problem have been proposed [33]. 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 [30]. Besides providing an efficient approximation to finding maximal cliques, the framework of dominant sets provides a principled measure of cohesiveness of

[2] A subset of nodes is a *clique* if all its nodes are mutually adjacent; a *maximal* clique is not contained in any larger clique; a *maximum* clique has largest cardinality.

a class as well as a measure of node participation.

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 $A = (a_{ij})$ defined as:

$$a_{ij} = \begin{cases} \text{sim}(i,j) \text{ if } (i,j) \in \text{ E} \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{2}$$

Here $\text{sim}(i, j)$ is computed using our proposed notion of similarity as defined in Equation 1. To quantize the cohesiveness of a node in a cluster, we define its "average weighted degree". Let $S \subseteq V$ be a non-empty subset of vertices and $i \in S$, such that,

$$\text{awdeg}_{\text{S}}(i) = \frac{1}{||\text{S}||} \sum_{j \in \text{S}} a_{ij} \tag{3}$$

and

$$\Phi_{\text{S}}(i,j) = a_{ij} - \text{awdeg}_{\text{S}}(i) \quad \text{for } j \notin \text{S} \tag{4}$$

Intuitively, $\Phi_{\text{S}}(i, j)$ measures the similarity between nodes j and i, with respect to the average similarity between node i and its neighbors in S. Note that $\Phi_{\text{S}}(i, j)$ can either be positive or negative.

We now consider how weights are assigned to individual nodes. Let $S \subseteq V$ be a non-empty subset of vertices and $i \in S$. The weight of i with respect to S is given as:

$$w_{\text{S}}(i) = \begin{cases} 1 & \text{if } ||\text{S}|| \ = \ 1 \\ \sum\limits_{j \in \text{S} \setminus \{i\}} \Phi_{\text{S}\setminus\{i\}}(j,i) w_{\text{S}\setminus\{i\}}(j) & \text{otherwise} \end{cases} \tag{5}$$

Moreover, the total weight of S is defined as

$$W(S) = \sum_{i \in S} w_S(i) \tag{6}$$

Intuitively, $w_S(i)$ gives a measure of the overall similarity between vertex i and the vertices of $S \setminus \{i\}$ with respect to the overall similarity among the vertices in $S \setminus \{i\}$. We are now in a position to define *dominant sets*. A non-empty

sub-set of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq S$, is said to be *dominant* iff:

- $w_{\mathrm{S}}(i) > 0$, $\forall\ i \in \mathrm{S}$, *i.e.* internal homogeneity
- $w_{\mathrm{S}\cup\{i\}}(i) < 0\ \forall\ i \notin \mathrm{S}$, *i.e.* external inhomogeneity.

Effectively, we can state that the dominant set in an edge-weighted graph is equivalent to a cluster of vertices in that graph. Since solving Equation 5 combinatorially is infeasible, we use a continuous optimization technique of replicator dynamics (for details, see [30]).

5 Activity Classification and Anomaly Explanation

Given $||C||$ discovered activity-classes, we are interested in finding if a new activity instance is regular or anomalous. Each member j of an activity-class c has some weight $w_c(j)$, that indicates the participation of j in c. We compute the similarity between a new activity-instance τ and the previous members of each class by defining a function $A_c(\tau)$ as:

$$A_c(\tau) = \sum_j sim(\tau, j) w_c(j) \qquad \forall j \in c \tag{7}$$

Here $w_c(j)$ is the same as defined in Equation 5. A_c represents the average weighted similarity between the new activity-instance τ and any one of the discovered classes c. The selected membership class c^* is found as

$$c^* = \arg\max_{\forall c} A_c(\tau) \tag{8}$$

Once the membership decision of a new test activity has been made, we now focus our attention on deciding whether the new class member is regular or anomalous. Intuitively speaking, we want to decide the normality of a new instance based on its closeness to the previous members of its membership activity-class. This is done with respect to the average closeness between all the previous members of its membership class. Let the function $\Gamma(\tau)$ be:

$$\Gamma(\tau) = \sum_{j \in c^*} \Phi_{c^*}(j, \tau) w_{c^*}(j) \tag{9}$$

where Φ is defined by Equation 4. We define a new class member τ as regular if $\Gamma(\tau)$ is greater than a particular threshold. The threshold on $\Gamma(\tau)$ is learned by mapping all the anomalous activity instances detected in the training activity-set to their closest activity-class (using Equation 7 & 8), and computing the value of Γ for both regular and anomalous activity instances. We can now observe the variation in false acceptance rate and true positives as a function of the threshold Γ. This gives a "Receiver Operating Curve" (ROC). The area under ROC is indicative of the confidence in our detection metric $\Gamma(\tau)$ [21].

Based on our tolerance for true and false positive rates, we can choose an appropriate threshold.

5.1 Anomaly Explanation

Explanation of the detected anomalous activities is a function of characterization of the general properties of an activity-class. One way of characterizing these properties is to find the best representative or typical member of a class [23]. The question of typicality is closely related to the similarity of a node to other members of a class. The problem has been previously approached as finding the node with min-max distance from other nodes [12], or the node with maximum in-degree [17]. Such approaches however either assume the clusters to be well behaved, or take a very local view of a node's similarity to its neighbors.

5.2 Activity-Class Modeling

Following [23], we propose the idea of typical nodes (mentioned as "authoritative sources" in [23]) and "similar to typical (STT)" nodes (mentioned as "hubs" in [23]). Typical and STT nodes exhibit a mutually reinforcing relationship - a good STT node is one which is closer to a Typical node, while a Typical node is one closer to more STT nodes. Following [23], we associate a non-negative Typicality weight x^p and a non-negative STT weight y^p to each node in the cluster where p denotes the index of nodes in a cluster. Naturally, if p is closer to many nodes with large x values, it should receive a large y value. On the other hand if p is closer to nodes with large y values, it should receive large x value. We define two coupled processes to update weights x^p and y^p iteratively, *i.e.*

$$x^p \leftarrow \sum_{q:(q,p)\in E} y^q \quad \text{and} \quad y^p \leftarrow \sum_{q:(q,p)\in E} x^q \tag{10}$$

As we iterate the above two equations k times in the limit $k \leftarrow \infty$, x^p and y^p converge to x^* and y^*. The node which has the largest component in the converged vector x^* would correspond to the node which has the greatest Typical weight and hence is the best representative of the nodes of clusters. x^* can be computed from the Eigen Analysis of the matrix $A^T A$ where A is the symmetric similarity matrix of all the nodes of the cluster. Essentially x^* is the principal eigenvector (the one with greatest corresponding Eigen value) of $A^T A$, the largest component of which corresponds to the Typical Node of the cluster (for proof, see [23]).

5.2.1 Explanatory Features

For large scale surveillance systems, it is imperative to find the features that can be used to explain an anomalous activity in a maximally-informative manner. We are interested in features of an activity-class that have minimum en-

tropy, and occur frequently. The entropy of an n-gram indicates the variation in its observed frequency, which in turn indicates the confidence in the prediction of its frequency. The frequency of occurrence of an n-gram suggests its participation in an activity-class. We want to analyze the extraneous and the pertinent features in an activity sequence that made it anomalous with respect to the most explanatory features of the regular members of the membership activity-class. We now construct our approach mathematically (a figurative illustration is given in Figure 7).

Let α_i denote a particular n-gram i for an activity, and c denote any of the $||C||$ discovered activity-classes. If R denotes the typical member of c as described in §5.2, and τ denotes a new activity-class member detected as being anomalous, then we can define the difference between their counts for α_i as:

$$D(\alpha_i) = f_R(\alpha_i) - f_\tau(\alpha_i) \tag{11}$$

where $f(\alpha_i)$ denotes the count of an n-gram α_i. Let us define the distribution of the probability of occurrence of α_i in c as:

$$P_c(\alpha_i) = \frac{\sum\limits_{k \in c} f_k(\alpha_i)}{\sum\limits_{i=1}^{M} \sum\limits_{k \in c} f_k(\alpha_i)} \tag{12}$$

where M represents all the non-zero n-grams in all the members of activity-class c. Let us define multi-set χ_c^i as:

$$\chi_c^i = \{f_k(\alpha_i) | k \in c\} \tag{13}$$

We can now define probability $Q(x)$ of occurrence of a particular member $x \in \chi_c^i$ for α_i in c as:

$$Q(x) = \psi \sum_{j \in c} \begin{cases} 1 \ if \ f(\alpha_i) = \ x \\ 0 \qquad \text{otherwise} \end{cases} \tag{14}$$

where ψ is the normalization factor. Let us define Shannon's Entropy of a tri-gram i for an activity-class c by $H_c(\alpha_i)$ as:

$$H_c(\alpha_i) = \sum_{x \in \chi_c^i} Q_c(x) ln(Q_c(x)) \tag{15}$$

We can now define the notion of *predictability*, $\text{PRD}_c(\alpha_i)$, of the values of tri-gram α_i of cluster c as:

$$\text{PRD}_c(\alpha_i) = 1 - \frac{H_c(\alpha_i)}{\sum\limits_{i=1}^{M} H_c(\alpha_i)} \tag{16}$$

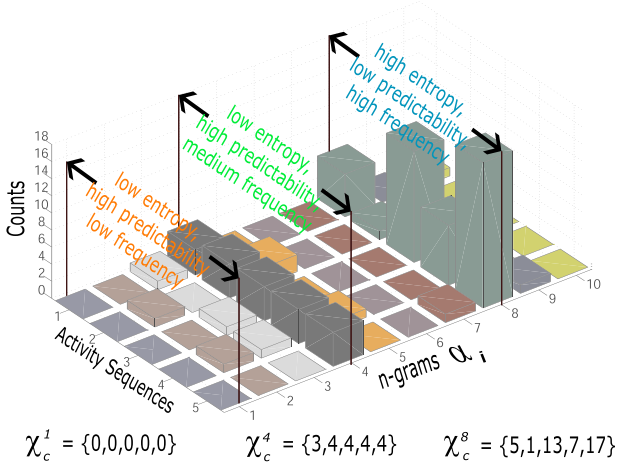


Fig. 7. Five simulated activity sequences are shown to illustrate the different concepts introduced in § 5.2.1. α_1 has low value of P_c, its entropy H_c is low and therefore its predictability is high. α_4 has medium P_c, its entropy H_c is also low and its predictability is high. Finally α_8 has high P_c, but its entropy H_c is high which makes its predictability low. α_1 could be useful in explaining the extraneous features in an anomalous activity, while α_4 could be useful in explaining the features that were deficient in an anomaly.

It is evident from this definition, that α_i with high entropy $H_c(\alpha_i)$ would have high variability, and therefore would have low predictability.

We define the explainability of an n-gram $\alpha_i \in c$ that was frequently and consistently present in the regular activity-class as:

$$\xi_c^P(\alpha_i) = \mathrm{PRD}_c(\alpha_i) P_c(\alpha_i) \tag{17}$$

Intuitively, ξ_c^P indicates how much an α_i is instrumental in representing a activity-class c.

Similarly, we can define the explainability of $\alpha_i \in c$ in terms of how consistently was it absent in representing c.

$$\xi_c^A(\alpha_i) = \mathrm{PRD}_c(\alpha_i)(P_c^{\max}(\alpha_i) - P_c(\alpha_i)) \tag{18}$$

where $P_c^{\max}(\alpha_i)$ is the maximum probability of occurrence of any α_i in c.

The first term in both Equation 17 and 18 indicates how consistent α_i is in its frequency over the different members of a class. The second term in Equation 17 and 18 dictates how representative and non-representative α_i is for c respectively.

Given an anomalous member of a activity-class, we can now find the features that were frequently and consistently present in the regular members of the

activity-class, but were deficient in the anomaly τ. To this end, we define the function Deficient(τ) as:

$$\text{Deficient}(\tau) = \arg\max_{\alpha_i}[\xi_c^P(\alpha_i)D_c(\alpha_i)] \tag{19}$$

Similarly, we can find the most explanatory features that were consistently absent in the regular members of the membership activity-class but were extraneous in the anomaly. We define the function Extraneous(τ) as:

$$\text{Extraneous}(\tau) = \arg\min_{\alpha_i}[\xi_c^A(\alpha_i)D_c(\alpha_i)] \tag{20}$$

We can explain anomalies based on these features in two ways. Firstly, we can consider features that were deficient from an anomaly but were frequently and consistently present in the regular members. Secondly, we can consider features that were extraneous in the anomaly but were consistently absent from the regular members of the activity-class.

6 Results: Class Discovery, Classification & Anomaly Explanation

To test the competence of our proposed framework, experiments on data-sets collected from three everyday environments were performed. The explanation of the experimental setups and results obtained in these settings are presented in the following.

6.1 Experimental Setup - Loading Dock Area

We collected video data at the Loading Dock Area (LDA) of a retail bookstore. We installed two cameras with partially overlapping fields of view. A schematic diagram with sample views from the two cameras is shown in Figure 8. Different delivery activities take place in this environment, and to get the reader better situated with the dynamics of this environment, some of the events from one of the collected activity are shown in Figure 9. Daily activities from 9a.m. to 5p.m., 5 days a week, for over one month were recorded, during which we collected 195 activities. Of these, 150 were randomly selected as our training set, while the remaining 45 were used as our testing set. We carefully identified 10 key-objects in the environment, whose various interactions constituted an event vocabulary of 61 events. Events of the 150 training activity instances were were manually annotated using our pre-defined event-vocabulary. For testing activities, we hand-tracked the key-objects and built low-level event detectors that used these object-tracks for semi-automatic event detection in test videos.

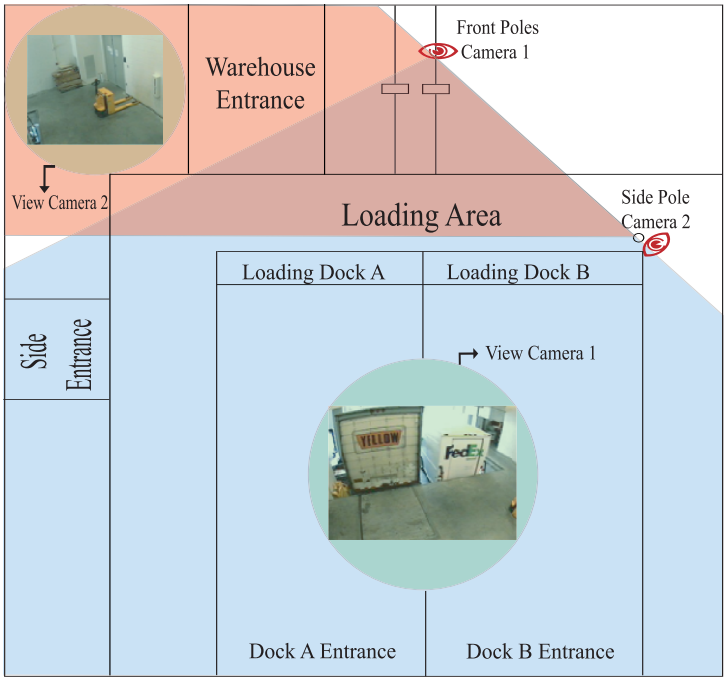


Fig. 8. A schematic diagram of the camera setup at the loading dock area with overlapping fields of view.

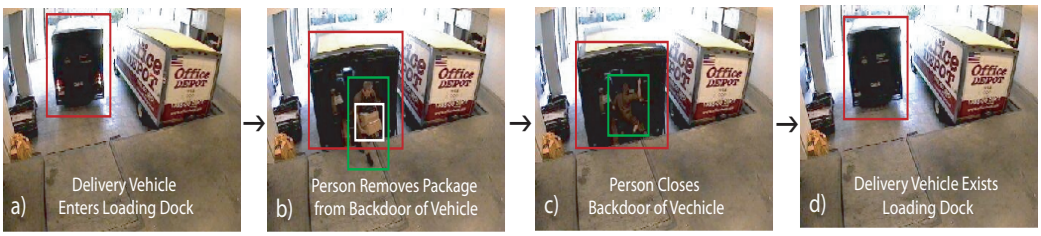


Fig. 9. **Key Frames of Example Events -** The figure shows an delivery activity in a loading dock area. Only Camera 1 is being shown here. The key-objects whose interactions define these events are shown in different colored blocks.

6.2 Analysis of Discovered Activity-Classes - Loading Dock Area

Out of the 150 training activities, we discovered 7 activity-classes, with 106 activities as part of any one of the discovered classes, while 44 activities being different enough to be not included into any class. The visual representation for the similarity matrices of the original 150 activities and the re-arranged activities in 7 classes is shown in Figure 10.

Analysis of these discovered activity-classes reveals a strong structural similarity amongst the class members. For instance, the most cohesive of the discovered classes was the one where all the UPS deliveries were clustered. It must be pointed out that there was no explicit information about the company-labels of the delivery vehicles in our vocabulary. The reason we were able to discover all UPS deliveries as a cohesive activity-class is because the activity-structure

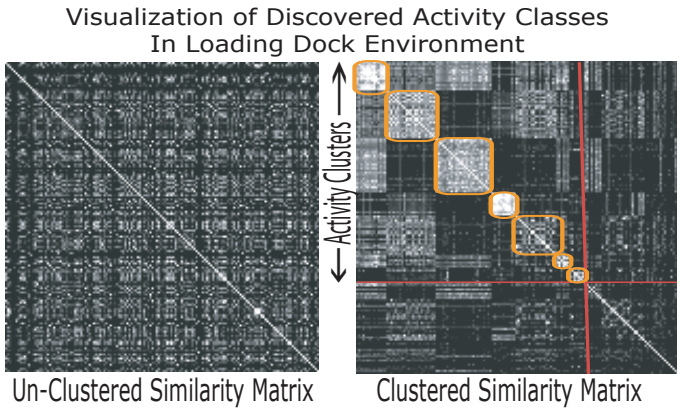


Fig. 10. Each row represents similarity of an activity with the entire training date. White implies identical similarity. Black represents complete dissimilarity. Activities ordered after the red cross line in the clustered similarity matrix were dissimilar enough to be considered anomalous

Class Index	Class Description
Class 1	UPS® delivery-vehicles that picked up multiple packages using hand carts.
Class 2	Pickup trucks and vans that dropped off a few packages without needing a hand cart.
Class 3	Delivery trucks that dropped off multiple packages, with multiple people using hand-carts.
Class 4	A mixture of car, van, and truck delivery vehicles that dropped off one or two packages without needing a hand cart.
Class 5	Delivery-vehicles that picked up and dropped-off multiple packages using a motorized hand cart and multiple people.
Class 6	Van delivery-vehicles that dropped off one or two packages without needing a hand cart.
Class 7	Delivery trucks dropped off multiple packages using hand carts.

Table 1

Description for the Discovered Classes in Loading Dock.: A brief description of the various discovered classes in the Loading Dock Environment are given in terms of the different distinguishing features.

induced by a UPS delivery by the virtue of where the truck docks, how many packages are delivered, in what manner are they delivered *etc.*, is reflected in our similarity metric, and is picked up by the discovery algorithm. This anecdotal evidence is an indication that the perceptual bias introduced by us in terms of the event-vocabulary, is successfully manipulated at the higher-level discovery algorithm. A brief description of the discovered activity-classes is given in Table 1.

6.3 Experimental Setup - Residential House Environment

To test our proposed algorithms on the activities in a residential house environment, we deployed 16 strain gages at different locations in a house, each with a unique identification code. These transducers register the time when the resident of the house walk over them. The data was collected daily for

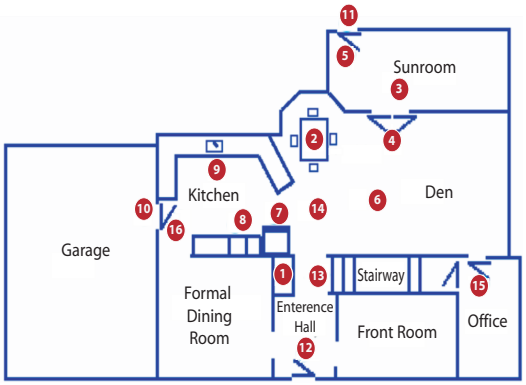


Fig. 11. A schematic diagram of the strain-gage setup in the house scenario. The red dots represents the positions of the strain gages.

almost 5 months (151 days - each day being considered as an individual activity). Whenever the person passed near a transducer at a particular location, it was considered as the occurrence of a unique event. Thus our event vocabulary in this environment consists of 16 events. Figure 11 shows a schematic top-view of this environment.

6.4 Analysis of Discovered Activity-Classes - Residential House

Out of the 151 activities captured over a little more than 5 months, we found 5 activity-classes (maximal cliques), with 131 activities as members of any one of the discovered class, and 20 activities being dissimilar enough not to be a part of any non-trivial maximal clique. A brief description of the discovered activity-classes is given Table 2.

A closer analysis of these classes show the general behavior of the person depending on how long did the person spend in the house, what parts of the house did he spent most of his time at while he was inside, and what were the most frequent location-transitions that he made. These behaviors seem to correlate with other physical information not encoded in the data, such as what day of the week it was *etc.* This demonstrates that the proposed system can be very useful for monitoring the everyday activities of senile individuals to see if there are any anomalous patterns.

6.5 Experimental Setup - Household Kitchen

One of the main reasons of exploring this environment was to study how our framework performs when the events are detected in a completely automatic manner using low-level pixel information. To this end, we deployed a top-down static camera in a household kitchen to record a users interactions with different key-objects known *a priori*. 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/`.

Class Index	Class Description
Class 1	Activities lasting for the entire length of days where the person's trajectory spans the entire house space. Most of the time was spent in the area around the Kitchen and the Dining Table.
Class 2	The person moves from from kitchen to the stairway more often. Further more, as opposed to cluster 1, the person does not go from the Office to the Sum Room area.
Class 3	The person spends more time in the areas of Den and the living-room. Moreover, he visits the Sun-room more often.
Class 4	The person spends most of the day in Kitchen and Dining Room. The duration for which she stays in the house is smaller for this class.
Class 5	The person moves from Dining Room to the Sun Room more often. The duration for which she stays in the house is significantly smaller than any other activity-class.

Table 2

Description for the Discovered Classes in Residential House: A brief description of the various discovered classes in the Residential House Environment are given in terms of the different distinguishing features.

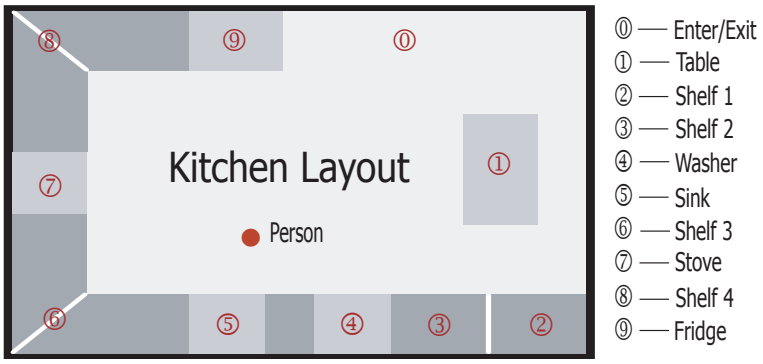


Fig. 12. A schematic diagram showing the kitchen floor layout, and the location of the considered key-objects.

The floor-layout of the kitchen and the key-objects are shown in Figure 12.

6.5.1 Automatic Event Detection in Household Kitchen Environment

One of the imperatives of exploring this environment was to see how our framework performs when the events are detected completely automatically using the low-level pixel values. For this setup, 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 a previously proposed tracking framework [41]. 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 filter for next

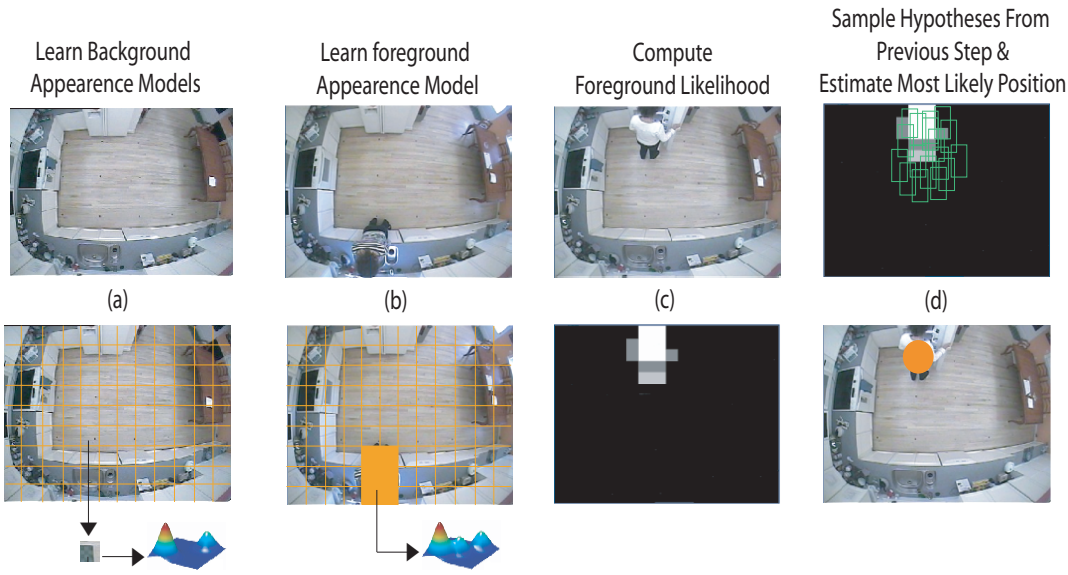


Fig. 13. **Figurative Illustration of Person Tracking**: (1) The background image is divided into multiple regions, and a gaussian mixture model for the chromatic content of each region is learned. (2) These background models are used to subtract the foreground from the background, and another set of gaussian mixture models are learned for the chromatic content of the foreground. (3) During testing, the likelihood of a portion of the image belonging to foreground is computed using the background and foreground appearance models. (4) A fixed number of most likely hypotheses (particles) are sampled from the previous frame, and are re-distributed using a motion model. These hypotheses are weighted using foreground likelihood in test image, and used to infer position of the person in current frame.

iteration. This process is figuratively illustrated in Figure 13.

6.6 Analysis of Discovered Activity-Classes - Household Kitchen

The purposes of conducting this experiment was to explore how many of the original activity-classes that we know are present in our activity-corpus can n-grams extract for different values of n. 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 were merged. We ran the discovery algorithm for different values of thresholds, and the best obtained results are given in Table 3.

As can be observed that as the value of n increases, the n-grams are able to capture the activity structure more explicitly, resulting in the recovery of more number of activity-classes. The quality of the recovered classes also increases with the increase in the value of n.

Note that this trend however cannot continue indefinitely. This is because with higher values of n the sparsity of the data would increase to a point where the structural signature of activities present in the data might be lost. Therefore,

	1-grams		3-grams		5-grams	
	P	*R*	*P*	*R*	*P*	*R*
Aloo Dam	55.5	50.0	50.0	60.0	54.5	60.0
Babka	-	-	55.5	50.0	37.5	30.0
Cereal	60.0	60.0	57.1	40.0	33.3	30.3
Fruit Salad	-	-	-	-	33.3	40.0
Omelet	-	-	-	-	-	-
Raita	-	-	17.9	70.0	33.3	70.0
Chicken	16.3	100.0	44.4	40.0	41.6	50.0
Setup Table	60.0	60.0	50.0	50.0	45.4	50.0
Green Salad	-	-	40.0	20.0	37.5	30.0
Wash Dishes	50.0	50.0	44.4	40.0	25.0	20.0
Average	**24.1**	**32.0**	**35.9**	**37.0**	**34.1**	**38.0**
% Discovery	**50**		**80**		**90**	

Table 3

Comparative performance for Class Discovery - The table shows the discovered number of activity-classes while using the n-gram representation for different values of n.

there is a need to first discover the predominant mode of temporal dependence of events in an environment, which could be used to set an optimal value of n. We leave the discovery of an optimal value of n as part of our future work.

6.7 Detected Anomalies - Loading Dock Environment

We now present a detailed explanation of how using the initially detected anomalous activities in the Loading Dock Area (see Section 6.2), we can learn a threshold for detecting new anomalous activity-class members, validate how intuitive are these detected anomalies from a human view-point, and explain in what ways are the detected anomalies different from the regular members of their membership classes.

6.7.1 Analysis of Detected Anomalies

Analyzing the detected anomalous activities reveals that there are essentially two kinds of activities that are being detected as anomalous, (1) ones that are truly alarming, where someone must be notified, and (2) those that are simply unusual delivery activities with respect to the other regular activities. Key-frames for three of the truly alarming anomalous activities are shown in Figure 14. Figure 14-a shows a truck driving out without closing it's back door. Not shown in the key-frame is the sequence of events where a loading-dock personnel runs after the delivery vehicle to tell the driver of his mistake. Figure 14-b shows a delivery activity where a relatively excessive number of people unload the delivery vehicle. Usually only one or two people unload a delivery vehicle, however as can be seen from Figure 14-b, in this case there were five people involved in the process of unloading. Finally, Figure 14-c shows the unusual even of a person cleaning the dock-floor.

a b c

Fig. 14. **Anomalous Activities** - (a) shows a delivery vehicle leaving the loading dock with its back door still open. (b) shows an unusual number of people unloading a delivery vehicle. (c) shows a person cleaning the loading dock floor.

6.7.2 User Study For Detected Anomalies

To analyze how intuitive the detected anomalies are to humans, a user test involving 7 users was performed. First 8 regular activities for a subject were selected so she could understand the notion of a regular activity in the environment. 10 more activities were selected, 5 of which were labeled as regular by the system while the rest of the 5 were detected as anomalies. Each of the 7 users were shown these 10 activities and asked to label every one of them as a regular instance or an anomaly based on the regular activities previously shown. Each of the 10 activities were given labels based on what the majority agreed upon. 8 out of 10 activities labeled by the users, corresponded with the labels of the system. The probability of the system choosing the correct label 8 out of 10 times by chance is 4.4% [3]. This highlights the interesting fact that the anomalies detected by the proposed system fairly match the natural intuition of human observers.

6.8 Noise Analysis of n-grams in Loading Dock Environment

The results presented thus far were generated using activities with hand-labeled events. However, using low-level vision sensors to detect these events will generate noise. This invites the question as to how well would the proposed system perform over noisy data. In the following, the noise analysis to check the stability and robustness of the proposed framework is presented; allowing one to make some predictions about its performance on data using low-level vision.

Given the discovered activity-classes and the learned detection threshold using the training set of 150 activity-instances, various types and amounts of noise to the 45 test sequences was added, and the following two tests were performed:

[3] Given that the probability of correctly choosing the true label by simply guessing is 0.5, the binomial probability states that chance of an 8/10 success is $C_8^{10}(0.5)^8(0.5)^2 \approx .0439$

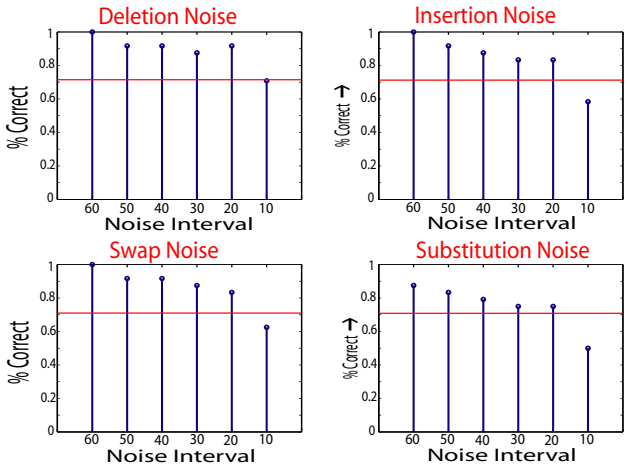


Fig. 15. Noise Analysis - Loading Dock Area: Each graph shows system-performance under synthetically generated noise using different generative noise models.

(1) ***Regular Classification Rate:*** Percent activities classified as regular members in the 45 ground truth test activities maintain their correct activity-class and regular-membership labels in the face of noise.

(2) ***Anomaly Detection Rate:*** Percent of 45 ground truth test activities detected as anomalies still get detected as anomalies in the face of noise.

Different amounts of noise using four types of noise models, Insertion Noise, Deletion Noise, Substitution Noise and Swap Noise was synthetically generated. We generated one noisy event-symbol using a particular noise model, anywhere within a window of a time-period for each activity in the testing data set. For instance Insertion Noise of time period 10 would insert one event-symbol between any two consecutive event-symbols, every 10 symbols. The classification performance of the proposed system under such noise model is shown in Figure 15. The system performs robustly in the face of noise and degrades gracefully as the amount of noise increases. Likewise, the anomaly detection capability of our system in the face of synthetically generated noise is shown in Table 4. The reason for such high detection rate even with large amount of synthetic noise is that it is unlikely that an anomaly would transform into something regular when perturbed randomly.

6.9 Automatic Event Detection

To move one step closer towards using low-level vision, we wrote a feature-labeling software that a user uses only to label the various objects of interest in the scene such as the doors of the loading dock, the delivery vehicles and its doors, people, packages and carts. We assign each object a unique ID during labeling. The ID numbers and object locations are stored in an XML format on a per-frame basis. We also wrote event detectors that parsed the XML data files to compute the distances between these objects for the 45 test activities.

Noise Model	%age Correct
Insertion Noise	100%
Deletion Noise	99%
Swap Noise	97%
Substitution Noise	100%

Table 4

Anomaly Detection Rate: The average detection rate of the system in the face of noise.

Based on the locations and velocities of these objects, the detectors performed automatic event detection.

The horizontal line in Figure 15 shows the *Regular Classification Rate* of our system over these automatically generated event sequences, *i.e.* 70.8%. The results for *Anomaly Detection Rate* for the automatically generated event sequences is 90.48%.

6.10 Anomalous Activity Explanation

Figure 16 shows the explanation generated by the system for the three anomalous activities (shown in Figure 14). The anomaly shown in Figure 14- (a) was classified to a activity-class where people frequently carry packages through the front door of the building. There was only one person in this anomaly who delivers the package through the side door. This is evident by looking at the extraneous features of the anomaly (Figure 16-b) where the tri-gram `Person Full Handed` → `Person Exits Side Door` → `Person Empty Handed` captures this difference. The second tri-gram of Figure 16-b, `Person Full Handed` → `Person Exits Back Door` → `Person Full Handed` shows the fact that there was another person who went out of the garage to tell the driver of the delivery vehicle that his back door was open.

The membership activity-class of anomaly in Figure 14-b has people frequently carrying packages through the front door of the building. In this anomaly, all of the workers go to the side door of the building. Moreover, majority of events in this anomaly were related to carts that is not one of the general characteristic of its membership activity-class. This is shown in Figure 16-d by tri-grams `Person Enters Back Door of DV` → `Person Empty Handed` → `Person Pushes Cart from Back Door of DV`, and `Person Empty Handed` → `Person Pushes Cart from Back Door of DV` → `Cart Empty`. Similarly Figure 16- (e) and Figure 16-f explain how anomaly in Figure 14-c was different from its membership activity-class.

7 Activity-Class Characterization

So far, we have considered situations where the beginning and end of activities is explicitly known. However, there are many scenarios where such demarcations are not so well-defined. For such situations, it is crucial to find concise

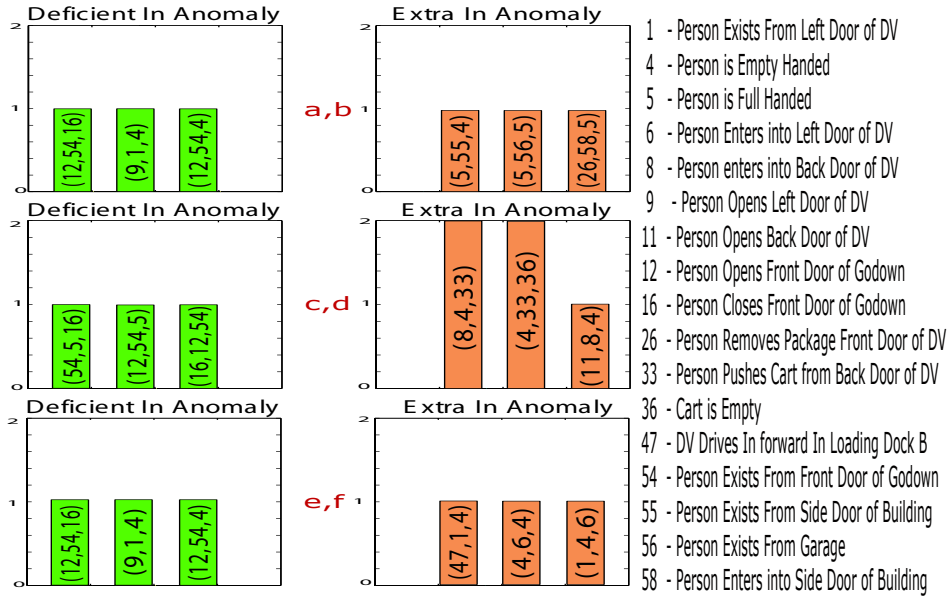


Fig. 16. Anomaly Explanation - Explanations for anomalies in Figure 14. The n-grams with less frequency than expected are shown in green, while those with frequencies greater that their expected frequency are shown in orange.

characterizations of the discovered activity-classes that could be used for online activity classification and detection of anomalous activities. We formalize this problem as finding predictably recurrent activity subsequences (called event motifs) using variable-memory Markov chains (*VMMC*). Note that our usage of (*VMMC's*) in § 3.1 to empirically analyze the competence of n-grams, was for purely generative purposes. However, here we describe a novel method to learn the *VMMC* model of an activity-class in a data-driven manner.

7.1 Defining Event Motifs

We are interested in frequently occurring event subsequences that are useful in predicting future events in activities. Following [45], we assume that a class of activity sequences can be modeled as a variable-memory Markov chain (*VMMC*). We define an ***event-motif*** for an activity-class as one of the variable-memory elements of its *VMMC*. We cast the problem of finding the optimal length of the memory element of a *VMMC* as a function optimization problem and propose our objective function in the following.

7.2 Formulation of Objective Function

Let Y be the set of events, A be the set of activity-instances, and C be the set of discovered activity-classes. Let function $\mathcal{U}(a)$ map an activity $a \in A$ to its membership class $c \in C$. Let the set of activities belonging to a particular class $c \in C$ be defined as $A_c = \{a \in A : \mathcal{U}(a) = c\}$. For $a = (y_1, y_2, ..., y_n) \in A$

where $y_1, y_2, ...y_n \in Y$, let $p(c|a)$ denote the probability that activity a belongs to class c. Then,

$$p(c|a) = \frac{p(a|c)p(c)}{p(a)} \propto \prod_{i=1}^{n} p(y_i|y_{i-1}, y_{i-2}, ..., y_1, c) \tag{21}$$

where we have assumed that all activities and classes are equally likely. We approximate Eq 21 by a *VMMC*, M_c as:

$$\prod_{i=1}^{n} p(y_i|y_{i-1}, ..., y_1, c) = \prod_{i=1}^{n} p(y_i|y_{i-1}, ..., y_{i-m_i}, c) \tag{22}$$

where $m_i \leq i-1 \ \forall \ i$. For any $1 \leq i \leq n$, the sequence $(y_{i-1}, y_{i-2}, ..., y_{i-m_i})$ is called the *context* of y_i in M_c([45]), denoted by $\mathcal{S}_{M_c}(y_i)$. We want to find the sub-sequences which can effeciently characterize a particular class, while having minimal representation in other classes. We therefore define our objective function as:

$$\mathcal{Q}(M_c|A_c) = \gamma - \lambda \tag{23}$$

where

$$\gamma = \prod_{a \in A_c} p(c|a) \quad \text{and} \quad \lambda = \sum_{c' \in C \setminus \{c\}} \prod_{a \in A_{c'}} p(c'|a) \tag{24}$$

Intuitively, γ represents how well a set of event-motifs can characterize a class in terms of correctly classifying the activities belonging to that class. On the other hand, λ denotes to what extent a set of motifs of a class represent activities belonging to other classes. It is clear that maximizing γ while minimizing λ would result in the optimization of $\mathcal{Q}(M_c|A_c)$. Note that our motif finding algorithm leverages our activity-class discovery framework by using the availability of the discovered activity-classes to find the maximally mutually exclusive motifs.

7.3 Objective Function Optimization

We now explain how we optimize our proposed objective function. [45] describe a technique to compare different *VMMC* models that balances the predictive power of a model with its complexity. Let s be a context in M_c, where $s = y_{n-1}, y_{n-2}, ..., y_1$, and $y_{n-1}, y_{n-2}, ..., y_1 \in Y$. Let us define the suffix of s as $\mathit{suffix}(s) = y_{n-1}, y_{n-1}, ...y_2$. For each $y \in Y$, let $N_{A'}(y, s)$ be the number of occurrences of event y in activity-sequences contained in $A' \subseteq A$ where s precedes y, and let $N_{A'}(s)$ be the number of occurances of s in activity-sequences in A'. We define the function $\Delta_{A'}(s)$ as

$$\Delta_{A'}(s) = \sum_{y \in Y} N(s, y) log \left(\frac{\hat{p}(y|s)}{\hat{p}(y|_{\mathit{suffix}}(s))} \right) \tag{25}$$

where $\hat{p}(y|s) = N_{A'}(s,y)/N_{A'}(s)$ is the maximum likelihood estimator of $p(y|s)$. Intuitively, $\Delta_{A'}(s)$ represents the number of bits that would be saved if the events following s in A', were encoded using s as a context, versus having *suffix*(s) as a context. In other words, it represents how much better the model could predict the events following s by including the last event in s as part of context of these events.

We now define the function $\Psi_c(s)$ (bit gain of s) as

$$\Psi_c(s) = \Delta_{A_c}(s) - \sum_{c' \in C \setminus \{c\}} \Delta_{A_{c'}}(s) \tag{26}$$

Note that higher values of $\Delta_{A_c}(s)$ imply greater probability that an activity in A_c is assigned to c, given that s is used as a motif. In particular, higher the value of $\Delta_{A_c}(s)$, higher will be the value of γ. Similarly, higher the value of $\sum_{c' \in C \setminus \{c\}} \Delta_{A_{c'}}(s)$, higher the value of λ.

We include a sequence s as a context in the model M_c iff

$$\Psi_c(s) > K \times log(\ell) \tag{27}$$

where ℓ is the total length of all the activities in A, while K is a user defined parameter. The term $K \times log(\ell)$ represents added complexity of the model M_c, by using s as opposed to *suffix*(s) as a context, which is shorter in length and occurs at least as often as s. The higher the value of K the more parsimonious the model will be.

Equation 27 selects sequences that both appear regularly and have good classification and predictive power - and hence can be thought of as event-motifs. Work in [34] shows how the motifs in a *VMMC* can be represented as a tree. Work done in [1] presents a linear time algorithm that constructs such a tree by first constructing a data structure called a Suffix Tree to represent all subsequences in the training data A, and then by pruning this tree to leave only the sequences representing motifs in the *VMMC* for some activity-class. We follow this approach by using Equation 27 as our pruning criterion.

8 Results: Discovered Event Motifs

We now present the results of motifs we obtained using our method for the previously discovered activity-classes in Loading Dock and House environments.

8.1 Analysis of Discovered Event Motifs

The highest big-gain event-motifs found for the 7 discovered activity-classes in the Loading Dock domain are given in Table 5. The discovered motifs of activity-classes seem to characterize these classes efficiently. Note that the

discovered motifs for activity-classes where package delivery occurred, have events like Person Places Package In The Back Door Of Delivery Vehicle and Person Pushes Cart In The Front Door of Building→ Cart is Full. On the other hand event-motifs for activity-classes where package pick-up occurred, have events such as Person Removes Package From Back-Door Of Delivery Vehicle and Person Places Package Into Cart.

The highest big-gain event-motifs found for the 5 discovered activity-classes in the House scenario are given Table 6. The motifs for the House environment capture the position where the person spends most of her time, and the order in which she visits the different places.

8.2 Subjective Assessment of Discovered Motifs

To subjectively assess the interpretability of the motifs discovered by our proposed method, we performed an anecdotal user test involving 7 participants. For each participant, 2 of the 7 discovered activity-classes were selected from the Loading Dock environment. Each participant was shown 6 example activities, 3 from each of the 2 selected activity-classes. The participants were then shown 6 motifs, 3 for each of the 2 classes, and were asked to associate

Class Index	**Class Description**
Class 1	Person places package into back door of delivery vehicle → Person enters into side door of building → Person is empty handed → Person exists from side door of building → Person is full handed → Person places package into back door of delivery vehicle.
Class 2	Cart is full → Person opens front door of building → Person pushes cart into front door of building → Cart is full → Person closes front door of building → Person opens front door of building → Person exists from front door of building → Person is empty handed → Person closes front door of building.
Class 3	DV drives in forward into LDA → Person opens left door of DV → Person exists from left door of DV → Person is empty handed → Person closes the left door of delivery vehicle.
Class 4	Person opens back door of DV → Person removes package from back door of DV → Person removes package from back door of DV → Person removes package from back door of DV → Person removes package from back door of DV → Person removes package from back door of DV.
Class 5	Person closes front door of building → Person removes package from cart → Person places package into back door of DV → Person removes package from cart → Person places package into back door of DV → Person removes package from cart → Person places package into back door of DV.
Class 6	Person Removes Cart From Back Door of DV → Person Removes Package From Back Door of DV → Person Places Package Into Cart → Person Places Package Into Cart → Person Removes Package From Back Door of DV → Person Places Package Into Cart → Person Removes Package From Back Door of DV → Person Places Package Into Cart.
Class 7	Person closes back door of DV → Person opens left door of DV → Person enters left door of DV → Person is empty handed → Person closes left door of DV.

Table 5

Description for the Discovered Event Motifs in Loading Dock.: A brief description is given for the various discovered event motifs for the 7 discovered activity-classes in the Loading Dock Environment.

Class Index	Class Description
Class 1	Alarm → Kitchen entrance → Fridge → Sink → Garage door (inside).
Class 2	Stairway → Fridge → Sink → Cupboard → Sink.
Class 3	Stairway → Dining Table → Den → Living-room Door → Sun-room → Living-room door → Den.
Class 4	Den → Living-room door → Den → Kitchen Entrance → Stairway.
Class 5	Fridge → Dining Table → Kitchen Entrance → Fridge → Sink.

Table 6

Description for the Discovered Event Motifs in Residential House Domain.: A brief description is given for the various discovered event motifs for the 5 discovered activity-classes in the Residential House Environment.

each motif to the class that it best belonged to. Their answers agreed with our systems 83% of the time, *i.e.*, on average a participant agreed with our system on 5 out of 6 motifs. The probability of obtaining this agreement by random guessing [4] is only 0.093.

9 Conclusions & Future Work

This paper explores the problem of learning to discover and recognize human activities in everyday environments. Traditional approaches to this end assume that the structure of activities being modeled is known *a priori*. However, for a majority of everyday environments, the structure of such activities is generally not available. The main contribution of this work is an investigation of knowledge representations and manipulation techniques that can facilitate learning of everyday human activities in a minimally supervised manner.

We posit that if we choose to describe everyday activities in term of an appropriate set of events, then the structural information of these activities can be encoded using their local event subsequences, and that this encoding is sufficient for activity-class discovery and classification. With this perspective at hand, we particularly investigate representation of event n-grams that characterize activities in terms of their fixed length event subsequences.

Exploiting this representation, we propose a computational framework to discover the various activity-classes taking place in an environment. We model these activity-classes as maximally similar activity-cliques in a completely connected graph of activities, and describe how to discover them efficiently. Moreover, we propose methods for finding concise characterizations of these discovered activity-classes, both from a holistic as well as a by-parts perspective. Using such characterizations, we present an incremental method to classify a new activity instance to any one of the discovered activity-classes, and to automatically detect whether it is anomalous with respect to the general

[4] According to the binomial probability function the chance of randomly agreeing on 5 out of 6 motifs is $C_5^6(0.5)^1(0.5)^5$.

characteristics of its membership class. Our results show the efficacy of our framework in a variety of everyday environments, including a Loading Dock area, a Household Kitchen, and a Residential House environment.

9.1 Main Conclusions

In the following we describe the main conclusions of our work.

9.1.1 Learning Global Activity Structure Using Local Event Statistics

The key conclusion of this work is that if we describe everyday activities in terms of an appropriate set of events, the structural information of these activities can be uniquely encoded using statistics of their local event subsequences. At the heart of this idea of learning activity structure using event statistics is the question whether we can have such an appropriately expressive yet robustly detectable event vocabulary to describe human activities in a variety of everyday environments. There exists an inherent tradeoff between the expressiveness of events and the robustness with which they can be detected using low-level perceptual information. The way one strikes a balance between these two opposing factors will impact the kinds of analysis we can perform on the activities taking place in an environment.

9.1.2 Specificity versus Sensitivity of Sequential Representations

Another tradeoff we came across over the course of this work is between the specificity to which a representation captures the structure of a sequential process, and its sensitivity to sensor noise. While n-grams for smaller values of n only encode activity structure up to a fixed temporal scale, and are therefore less exact and more lossy, they are at the same time more robust to various perturbations brought about by the sensor noise. On the other hand n-grams with larger values of n are able to capture sequence structure over longer temporal scale, therefore encoding the structural information more exactly. However, their greater specificity results in their higher sensitivity to sensor noise. We saw this trend in the simulation experiments of Section 3.1.

9.1.3 Behavior Discovery Using Feature Based View of Activity-Classes

In this work, we have taken on the problem of unsupervised discovery of human behaviors with a feature-based view of activity-classes. This view posits that members of an activity-class generally share a set of common properties that make them perceptually similar to each other. For instance, activities of frying omelets look similar to each other as they mostly require events such as beating eggs followed by frying them. We believe that our representation of modeling activities as conjunctions of their sequential features supports a notion of their perceptual similarity that can be used for the unsupervised discovery and characterization of various human behaviors. We have shown that posing this question as a graph partitioning problem by modeling activity-classes as maximal cliques of nodes in activity-graphs is a plausible way of solving this

problem. Moreover, we showed that using the framework of Dominant Sets is an efficient means to this end.

9.1.4 A Detection Based Approach To Finding Anomalous Behaviors

One application of our proposed computational framework is automatically finding activities that are in some sense irregular or anomalous. As anomalies are rare occurrences with large variation amongst them, traditional approaches that attempt to learn explicitly defined models of anomalies do not generalize well. In this work, we have approached the problem of finding anomalous behaviors from a detection rather than a recognition based perspective. As the notion of anomaly is closely related to what is meant by regular, we have modeled anomalies as activities that deviate from behaviors perceived as regular in an environment. Using discovered activity-classes to learn the notion of regularity in an environment, we have tried to detect anomalies that deviate from regular behaviors.

Our research findings demonstrate that since all deviations from regular behaviors are not necessarily interesting, taking a purely detection-based perspective towards finding anomalies can in fact be too general. One way of striking a balance between the brittleness of an exclusively recognition-based perspective and the generality of a purely detection-based view towards anomalies, is to learn certain domain specific constraints on what makes various irregularities truly alarming. How these constraints should be modeled and learned for different environments remains an open question.

9.2 Current Limitations & Future Research Directions

At present, there are several limitations of our framework that might be used as avenues for future research. Some of these are given in the following.

9.2.1 Incorporating Temporal Information of Events

In everyday environments, any particular event may take variable time to finish. In a household kitchen for instance, the event of taking something out of the refrigerator may take longer or shorter time depending on how many items are being taken out. This duration over which an event takes place can be an important discriminating factor to distinguish amongst various activity-classes. Furthermore, the event duration can be an important indicator about whether the event was performed correctly or not. At present, we are not incorporating any information regarding the duration that the various events take to be executed. A potential future direction of our work might be to investigate the extent to which considering such temporal information of events is useful for activity analysis.

9.2.2 Automatic Parsing of Activities in a Stream of Detected Events

Currently, our framework assumes that the start and end of each activity is know *a priori*. However, there are many environments where such explicit demarcations of the start and end of activities are not available. One way of inferring these demarcations is to use the occurrence of event motifs in the event stream that are maximally mutually exclusive amongst the various activity-classes. We leave the exploration of such an approach to temporally segment activities from event streams as a part of our future work.

9.2.3 Analyzing Group Activities

Currently, we are only focusing on activities where one agent performs one event at a time. However, there is a large class of activities where multiple agents simultaneously perform multiple events. An important question to explore in the future would be how can our proposed framework scale up to efficiently analyzing such simultaneous streams of events.

9.3 Concluding Remarks & Discussion

We conclude with discussion regarding choice of an appropriate event vocabulary, and the general applicability of our proposed framework.

9.3.1 Choosing An Appropriate Event Vocabulary

The choice of a particular set of events to describe the *in situ* activities determines how strictly or loosely defined the structure of these activities is. In the following we enlist some of the criteria for selecting an event vocabulary suitable for a computational system such as ours.

(1) Events in an event vocabulary should be of finite duration and temporally local, *i.e.* events should have a finite duration between their start and end points, and this range should in general be reasonably smaller than the duration of an entire activity.
(2) Events should not be temporally overlapping. In other words, each event must end before another event starts. In situations where multiple events are being simultaneously performed by different agents, there must exist a mapping between which event is being performed by which agent.
(3) Events in an event vocabulary should not occur spuriously and there needs to be a strict correlation between the action of an agent which causes the occurrence of a particular event.
(4) Events should have temporal atomicity, *i.e.*, if there are two events in a vocabulary that cannot happen without being temporally adjacent to each other, then they should be merged to one, provided that the merged event can still be robustly detected. This condition supports a minimal sized event vocabulary, resulting in reduced computational complexity.
(5) An event vocabulary should be complete with respect to spanning the various occurrences of interest that can transpire in an environment.

A key factor to consider while choosing event vocabulary for an environment is the inherent tradeoff that exists between how well does a set of events capture the underlying structure of activities, versus how robustly these events can be detected using some low-level perceptual information. There does not exist a set of hard and fast rule according to which the granularity of constituent events should be selected, however in general this choice should be made based on the dynamics of an environment, and the available sensor modalities.

9.3.2 General Applicability of the Proposed Framework

The usefulness of a computational framework for activity analysis depends on the general characteristics of various activities that take place in an environment. Everyday environments can have a wide range of activity dynamics. On one end of this spectrum are the environments where activities with strict and well defined structure take place. Examples of such environments include assembly lines on factory floors, missile installation sites, or runways of aircraft carriers, where a very strict regimen is followed. For such environments, our proposed framework is overkill, and more grammar driven-approaches would work better. On the other end of this spectrum are the environments where activities show a very loosely defined structure. Examples of these include kindergarten playgrounds, scenarios of loitering at a subway station, or simply hanging out in the living room. Since there is not enough repetitive activity-structure, our system would have difficulty in finding it. Somewhere between these two ends is the set of environments where the activity structure is neither too strict, nor too loosely defined. Our proposed framework is geared towards this class of environments. Some of the general characteristics of such environments are listed in the following:

(1) Many different types of activities can take place in the environment, and the number of possible *in situ* activities is not necessarily known *a priori*.
(2) There exists enough variance amongst the instances of different types of activities so that it is not feasible to write an explicit grammar-based model for the different activity classes.
(3) Instances belonging to each type of activity require execution of multiple intermediate tasks (*i.e.* events) for their successful completion.
(4) All instances belonging to the various activity types taking place in an environment can be described in terms of a shared set of events.
(5) There exists a mostly common set of partially ordered constraints amongst the constituent events of any particular activity type. Constituent events of different activity types mostly adhere to sets of different partially ordered constraints.

Some of the example environments that generally have the aforementioned properties include car repair shops, surgical operation theaters, or building construction sites.

References

[1] A. Apostolico and J. Bejerano. Optimal amnesic probabilistic automata. *Journal of Computational Biology*, 7:381–393, 2000.

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

[3] T. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proc International Conference of Intelligent Systsystems in Molecular Biology, pp. 28-36*, 1994.

[4] G. Bejerano and G. Yona. Modeling protein families using probabilistic suffix trees. In *In the Proc. of International Conference of Research in Computational Molecular Biology*, 1999.

[5] A. Bobick. Movement, activity and action: the role of knowledge in the perception of motion. In *Movement, Activity and Action: the Role of Knowledge in the Perception of Motion, Royal Society Workshop on Knowledge-based Vision in Man and Machine.*, 1997.

[6] A.F. Bobick, S.S. Intille, J.W. Davis, F. Baird, C.S. Pinhanez, L.W. Campbell, Y.A. Ivanov, A. Schuette, and A. Wilson. The kidsroom: A perceptually-based interactive and immersive story environment. In *Vismod*, 1996.

[7] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *IEEE Conference of Computer Vision and Pattern Recognition*, 1997.

[8] S. Calderara, R. Cucchiara, and A. Prati. Detection of abnormal behaviors using a mixture of von mises distributions. *IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, 2007.

[9] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *SIGKDD*, 2003.

[10] T. Choudhury, M. Philipose, D. Wyatt, and J. Lester. Towards activity databases: Using sensors and statistical models to summarize people's lives. In *IEEE Data Engineering Bulletin*, 2006.

[11] A. Dey, R. Hamid, C. Beckmann, I. Li, and D. Hsu. a cappella: programming by demonstration of context-aware applications. In *SIGCHI*, pages 33–40, 2004.

[12] R. Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, 2000.

[13] F. Duchne, C. Garbay, and V. Rialle. Similarity measure for heterogeneous multivariate time-series. *Proc. of the 12th European Signal Processing Conference*, pages 7–10, 2004.

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

[15] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[16] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, and C. Isbell. Discovery and characterization of activities from event-streams. In *International Conference of UAI*, 2005.

[17] I. Heller and C. Tompkins. An extension of a theorem of dantzig's. In *Linear Inequalities and Related Systems*. Princeton University Press, 1956.

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

[19] K. Ilgun, R. Kemmerer, and P. Porras. State transition analysis: A rule-based intrusion detection approach. *IEEE Transaction on software engineering*, pages 188–199, 1995.

[20] Y. Ivanov and A. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE PAMI*, 22(8):852–872, 2000.

[21] A. Johnson and A. Bobick. Relationship between identification metrics: Expected confusion and area under a roc curve. In *IEEE CVPR*, 2002.

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

[23] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, 1999.

[24] L. Liao, D.J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence. J.*, 2007.

[25] D. Minnen, I. Essa, and T. Starner. Expectation grammars: Leveraging high-level expectations for activity recognition. In *IEEE Conference on CVPR. Madison, WI.*, 2003.

[26] D. Moore, I. Essa, and M. Hayes. Context management for human activity recognition. In *Proc. of Audio & Vision-based Person Authentication*, 1999.

[27] T. Oates. Peruse: An unsupervised algorithm for finding recurring patterns in time series. In *IEEE ICDM, Japan.*, 2002.

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

[29] J. Patino, E. Corvee, F. Bremond, and M. Thonnat. Management of large video recordings. *In the AmI.d 2007 Ambient Intelligence Developments*, 2007.

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

[31] C. Piciarelli, G. Foresti, and L. Snidaro. Trajectory clustering and its applications for video surveillance. *In Proc. of Advanced Video and Signal Based Surveillance*, 2005.

[32] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Alex Weibel and Kay-Fu Lee (eds.), Readings in Speech Recognition*, pages 267–296, 1990.

[33] V. Raghavan and C. Yu. A comparison of the stability characteristics of some graph theoretic clustering methods. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 3:393–402, 1981.

[34] D. Ron, Y. Singer, and N. Tishby. The power of amnesia: learning probabilistic automata with variable memory length. *Machine Learning*, 25:117–149, 1996.

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

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

[37] R. Schank. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press, 1983.

[38] 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 Computer Vision and Pattern Recognition*, 2004.

[39] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[40] G. Sukthankar and K. Sycara. Robust recognition of physical team behaviors using spatio-temporal models. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 638–645, 2006.

[41] J. Sullivan, A. Blake, M. Isard, and J. MacCormick. Bayesian object localisation in images. *IJCV*, 4, 2:111–135, September 2001.

[42] A. Toshev, F. Brmond, and M. Thonnat. An a priori-based method for frequent composite event discovery in videos. *The Proceedings of 2006 IEEE International Conference on Computer Vision Systems*, 2006.

[43] P. Tse, J. Intriligator, J. Rivest, and P. Cavanagh. Attention and the subjective expansion of time. *Perception and Psychophysics*, 66:1171–1189, 2004.

[44] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, 1979.

[45] 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.

[46] A. Yuille and N. Grzywacz. A computational theory for the perception of coherent visual motion. *Nature*, 333:71–74, may 1988.

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