Detection of Suspicious Behavior from a Sparse Set of Multiagent Interactions

Boštjan Kaluža
Jozef Stefan Institute
Ljubljana, Slovenia
bostjan.kaluza@ijs.si

Gal A. Kaminka
Bar Ilan University
Ramat Gan, Israel
galk@cs.biu.ac.il

Milind Tambe
University of Southern California
Los Angeles, California
tambe@usc.edu

ABSTRACT

In many multiagent domains, no single observation event is sufficient to determine that the behavior of individuals is suspicious. Instead, suspiciousness must be inferred from a combination of multiple events, where events refer to the individual's interactions with other individuals. Hence, a detection system must employ a detector that combines evidence from multiple events, in contrast to most previous work, which focuses on the detection of a single, clearly suspicious event. This paper proposes a two-step detection system, where it first detects trigger events from multiagent interactions, and then combines the evidence to provide a degree of suspicion. The paper provides three key contributions: (i) proposes a novel detector that generalizes a utility-based plan recognition with arbitrary utility functions, (ii) specifies conditions that any reasonable detector should satisfy, and (iii) analyzes three detectors and compares them with the proposed approach. The results on a simulated airport domain and a dangerous-driver domain show that our new algorithm outperforms other approaches in several settings.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: intelligent agents, multiagent systems

General Terms

Algorithms, Security, Experimentation

Keywords

suspicious behavior, multiagent interactions, scoring functions

1. INTRODUCTION

There is a significant amount of research in suspicious activity detection, given its importance in many domains [1, 5, 9, 16]. The goal is to augment the traditional security measures by scrutinizing the behavior of all the subjects in the environment. We target a large class of applications where no single event is sufficient to make a decision about whether behavior is suspicious or not. Instead, we face a sparse set of *trigger events* that identify interesting parts characterizing the behavior trace. Examples include a potentially suspicious passenger who appears to turn away in the presence of security personnel, but not blatantly so, hence no single such event is enough to raise suspicion. The main question we address is how to combine multiple events to decide whether an event trace corresponds to the behavior of a normal or a suspicious person.

There are four challenges that need to be addressed. First, there is no one significant event or incident that would help us to immediately reach a decision; a series of trigger events allows us to reach a decision. Second, we have no knowledge about the exact plans devised by a suspicious person. Third, trigger events include the interactions of multiple agents making recognition in the presence of noise difficult. Fourth, the degree of suspiciousness contributed by a suspicious event depends on the agent's behavior in the past. For example, the third suspicious event is evaluated differently than the first, since the agent's previous behavior indicates a tendency to behave suspiciously. Hence, the simple counting of suspicious trigger events cannot be applied, since it accumulates all the events linearly. Furthermore, most of the plan-recognition methods, which rely on a plan library, are insufficient, since plans are not known in advance.

This paper presents a two-step approach to suspicious behavior detection from a sequence of an agent's actions. The first step detects trigger events, i.e., interesting parts of the sequence that serve as evidence, and estimates the probability that an event is suspicious. For this task we present an approach using coupled hidden Markov models [4] that are able to model interactive behavior. The second step combines evidence from multiple events in order to determine suspiciousness.

The key contributions of this paper are in the second step, which is defined as a decision problem: Is the behavior of an agent suspicious given a sequence of trigger events? First, we formally describe the detection problem and specify the conditions that any reasonable detector should satisfy. Second, we analyze three detectors, namely the naive Bayes detector, the hidden Markov models and the utility-based plan recognition (UPR). These detectors, however, either simplify the problem or evaluate the events linearly. Finally, we present a novel detector that is a generalization of UPR and denoted as Function-UPR (F-UPR): (i) we define utilities as a set of functions over state transitions and observations; and (ii) we introduce an observation utility function that is especially suitable for suspicious behavior detection, since it is able to evaluate events non-linearly. The experimental evaluation on a simulated airport domain first compares the three detectors with our proposed approach. The best two approaches are additionally compared on the dangerous-driver domain.

2. MOTIVATING DOMAIN AND RELATED WORK

Airports require numerous security solutions, including the identification of suspicious activities among passengers and staff in sur-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

rounding areas. Our goal is to monitor passengers during the time they spend at the airport and to detect those that indicate a high level of stress, fear or deception. It is reasonable to assume that there is a camera network to track a passenger throughout the airport. We focus on a task where no single event is sufficient to identify a suspicious passenger, but a series of events establishes the decision over time. The detection of events might be limited due to noise or an inability to extract some features (e.g., using a ceiling-mounted camera one can extract the trajectory of a passenger, but not facial expressions), hence a normal person may appear suspicious (and vice versa). Also, a precise plan of the suspicious passenger is not known in advance. Other domains of interest may include catching a reckless driver executing dangerous (but still legal) maneuvers [2], detecting a pirate vessel that plans to capture a transport vessel and therefore avoids security patrols, etc.

There are two approaches to detecting deviant behavior [2]: *suspicious* and *anomalous* behavior detection. The first approach assumes a behavior library that encodes *negative behavior*, and thus recognizing observed behavior corresponds to identifying a match in the library. The second approach uses the behavior library in an inverse fashion, meaning that the library encodes only *positive behavior*. When an observed behavior cannot be matched against the library it is considered as anomalous. Several approaches have been proposed to tackle the problem either way. In the airport scenario various systems were introduced to automatically detect some of the threats, such as leaving objects behind [10], suspicious trajectory paths [16], thefts [10], and vandalism acts and fights [12]. There is also a commercially available system [7] that is able to detect events such as running passengers, climbing over a fence, etc. However, these approaches mainly deal with the detection of single incidents, which are clearly suspicious. They do not address accumulating suspicion as we do.

Another area of related work includes hidden Markov models (HMMs) [13] that are widely used in traditional activity recognition for modeling a sequence of actions. Brand et al. [4] introduced coupled HMMs as an extension with multiple hidden interacting chains that are able to model interactive behavior. Duong et al. [5] focused on the duration of activities and introduced switching hidden semi-Markov models that provide probabilistic constraints over the duration of plans, and applied them to the detection of anomalies in the activities of daily living. Although widely used, HMMs may become inadequate when actions are more complex or have long-term temporal dependencies [11].

Plan recognition algorithms may use a hybrid approach for suspicious activity recognition. A symbolic plan recognizer is used to filter consistent hypotheses, passing them to an evaluation engine, which focuses on ranking. Geib and Goldman presented PHATT [8], a probabilistic approach based on tree grammars able to cope with interleaved goals, partially ordered plans, and failed observed actions. Sukthankar and Sycara [14] addressed plan recognition for multiagent teams, where plans were ordered by linear accumulation of observed actions consistent with the plan. Another approach is presented by Avrahami-Zilberbrand and Kaminka [2, 3]. Utility-based Plan Recognition (UPR) introduces utility to the observer in selecting the recognition hypotheses. The main strength of UPR is that it can incorporate an observer's bias to events with a low likelihood, for example, the a-priori probability for planting a bomb is very low, but detecting it has a high expected utility. We further discuss this approach in Section 5.3.

Furthermore, intrusion detection systems analyze a variety of user activities to identify suspicious computer activities. Helman and Liepins [9] proposed an intrusion detection system that provides a rating for computer activities, demonstrating frequency estimator and matching rules. Esponda et al. [6] analyzed tradeoffs between positive and negative activity patterns in the library and presented an approach based on partially matching rules. These approaches similarly address the problem of how to decide whether a user's activity is suspicious, but differ significantly in using a different approach to match and assess behavior.

3. DEFINITIONS AND ASSUMPTIONS

Our methods are general, but for illustrative purposes we will make use of the airport domain to provide examples. We treat subjects as agents in a multi-agent environment. At this point we assume that we can perfectly observe their actions.

Definition 1. Action a_t is a tuple of observed feature values $\langle f_1, ..., f_n \rangle$ that describe state of an agent at a given time stamp t.

Definition 2. Action trace $\mathbf{a}^{(l)}$ is a totally-ordered sequence of l actions $\mathbf{a}^{(l)} = (a_1, a_2, ..., a_l)$.

Definition 3. Trigger event $x^{i,j} = (a_i, ..., a_j)$ is a subsequence of action trace $\mathbf{a}^{(k)}$ (s.t. $1 \leq i < j \leq k$). A trigger event x is described by probabilities that the corresponding subsequence is suspicious $s(x)$ and normal $n(x)$.

Definition 4. Event trace $\mathbf{x}^{(k)}$ is a totally-ordered sequence of k trigger events $\mathbf{x}^{(k)} = (x_1, x_2, ..., x_k)$.

We address the problem of suspicious behavior detection in two steps, as shown in Figure 1. The first step analyzes an action trace and the surrounding environment to detect trigger events that characterize its interesting parts. The event trace then enters the second step, where it is evaluated. If the evaluation result exceeds a threshold value or is large relative to other evaluations of the event traces, then it is considered as suspicious.

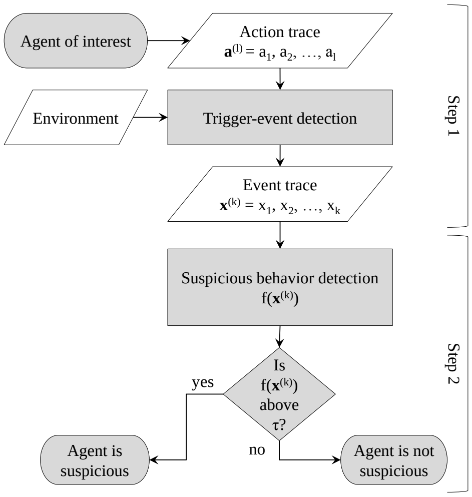


Figure 1: Two-step detection of suspicious behavior: (1) detection of trigger events and (2) detection of suspicious behavior.

Trigger events can be any kind of partial observations we are able to extract from the domain. In the airport domain, one can focus on people exhibiting indications of suspicious behavior, such as taking photos of critical infrastructure, revisiting the same location, evading the area when noticed, standing in customer service but not requesting the service, etc. We focus on a well-known detector obtained from conversations with domain experts. We observe the interactions between agents at the airport, more precisely, we are interested in how a passenger behaves in the presence of a uniformed authority figure. A person exposed to a high level of stress produces behavior that indicates fear, anxiety, pressure, tension, deception, etc. Hence, it is rational for the suspicious agent to minimize contacts with the authorities. Note, that no single avoidance is enough to raise a flag, but many such events put together cause the person to be treated as suspicious.

A trigger-event detection able to identify interactive behavior may rely on coupled hidden Markov models (CHMMs), which are briefly described below. The reader is referred to [4] for details; the CHMMs are not the main contribution of the paper. The observations consist of two action traces, namely the action trace of the agent of interest and the action trace of an authority agent when they are within some predefined radius. The CHMMs are able to model the complex, interactive behavior by two HMM chains, where the hidden states from one chain directly impact on the hidden states from the other chain. Figure 2 illustrates the CHMM for a pair of action traces with length $l = 3$. The current state Q_t^A of agent A is affected by both its previous state Q_{t-1}^A and previous state Q_{t-1}^B of the agent B (similarly Q_t^B is affected by Q_{t-1}^B and Q_{t-1}^A). Each state Q_i also impacts the corresponding observation state Y_t. For example, if the authority agent moves toward the suspicious agent, the next state of the latter takes this into account and produces an action for an avoidance maneuver.

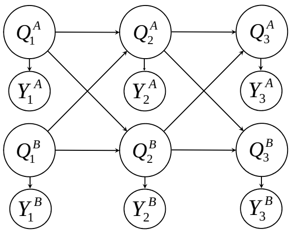


Figure 2: An example of CHMM for a pair of action traces with length $l = 3$.

A regular passenger may not turn or do anything different in the presence of authorities, while a suspicious person will (although as described below, an observer may not have perfect observability). Therefore, we create and train two CHMMs: $\hat{N}_I$ models the interactions produced by authorities and regular passengers, while $\hat{S}_I$ models the interactions produced by authorities and suspicious passengers. For a new event (interaction) x we compute the posterior probability that the event is generated with both models yielding $\hat{n}_I(x) = Pr\{x|\hat{N}\}$ and $\hat{s}_I(x) = Pr\{x|\hat{S}\}$, respectively.

4. PROBLEM DEFINITION

This section formally analyzes how to evaluate a sequence of trigger events. We leverage the Bayesian framework for intrusion detection [9] for the problem definition. At each time step t we observe an event x_t, generated by a hidden stochastic process H. Now suppose that H is a mixture of two auxiliary stochastic processes, namely the normal process N and the suspicious process S that correspond to a normal and a suspicious passenger. The random variable $y_t = 0$ if x_t is generated by N and $y_t = 1$ if x_t is generated by S. Since a suspicious passenger always emits a suspicious event (and a normal person a normal event), y for a specific agent does not change over time. In reality, there can be many subprocesses contributing to each of N and S, i.e., many normal users with different behavior patterns; however, here we assume only a single N and a single S that capture all the variability.

To this point we assumed that an observer is able to perfectly observe whether an event is generated by S or N. In practice, however, it may appear that a normal person emits suspicious events (or vice-versa). An observer might be limited for various reasons, such as an inability to detect characterizing features and noisy trigger-event detectors. Therefore, we relax this assumption as follows. An event x_t is observed as generated by N with the probability $n(x_t) = Pr\{H(t) = x_t|y_t = 0\}$ and as generated by S with the probability $s(x_t) = Pr\{H(t) = x_t|y_t = 1\} = 1 - n(x_t)$. The mixture distribution of an event x_t and a prior probability λ is

$$Pr\{H(t) = x_t\} = \lambda s(x_t) + (1-\lambda)n(x_t). \tag{1}$$

The objective of suspicious behavior detection is to identify those traces $\mathbf{x}^{(k)} = (x_1, x_2, ..., x_k)$ that are likely to be suspicious activities, i.e., traces $\mathbf{x}$ for which

$$Pr\{y = 1|H(t) = x_t, t = 1, ..., k\} > \tau, \tag{2}$$

is above some threshold τ or is large relative to the probability for other traces.

The reason why this problem is difficult is because of the nonlinear effect. Consider the following example. Suppose we observe a person do a U-turn in front of a police officer, so that the likelihood that this was a suspicious person becomes high. Later we see the same person doing a half-turn in front of a police officer. This trigger event if seen on its own, would not contribute much to the overall suspicion. However, following the initial turn we had observed, this new turn is a much stronger evidence to be attributed to the overall suspicion, because we bias the new event with our previous observation.

Theoretically, it might be possible to optimally detect suspicious behavior using Eq. (2). Unfortunately, this is usually not the case in practice. To see this, let us assume a prior probability $\lambda = Pr\{y_t = 1, t = 1, ..., k\}$. In most cases λ is close to 0, since in real-world applications suspicious activities are rare. Let the stochastic processes N, S and H denote $n(\mathbf{x}^{(k)}) = Pr\{H(t) = x_t, t = 1, ..., k|y = 0\}$, $s(\mathbf{x}^{(k)}) = Pr\{H(t) = x_t, t = 1, ..., k|y = 1\}$, and $h(\mathbf{x}^{(k)}) = Pr\{H(t) = x_t, t = 1, ..., k\}$, respectively. Using Bayes theorem we can derive from Eq. (2)

$$Pr\{y = 1|H(t) = x_t, t = 1, ..., k\} = \frac{\lambda \cdot s(\mathbf{x}^{(k)})}{h(\mathbf{x}^{(k)})} = \tag{3}$$

$$= \frac{\lambda \cdot \prod_{t=1}^{k} s(x_t|x_{i,i=t-1,...,1})}{\lambda \prod_{t=1}^{k} s(x_t|x_{i,i=t-1,...,1}) + (1-\lambda)\prod_{t=1}^{k} n(x_t|x_{i,i=t-1,...,1})}$$

To this point we implicitly assumed that the distributions λ, n and s are reliably estimable. The degree to which this assumption is valid depends on our detection capability. Suppose we have a sufficiently large dataset D_l of labeled event traces, we can estimate the prior probability λ from the D_l using the relative frequency, presenting the number of traces generated by a suspicious agent divided by the total number of traces (since traces can be of different lengths, the quotient is normalized by the traces' length). Note that in order to compute $Pr\{H(t) = x_t, t = 1, ..., k|y = 1\}$ we have to evaluate

$$s(x_1) \cdot s(x_2|x_1) \cdot ... \cdot s(x_k|x_{k-1}, ..., x_1) \tag{4}$$

While some first terms, i.e., $s(x_t), s(x_t|x_{t-1})$, can still be estimated, the estimation of latter terms including increasingly more history becomes less and less reliable. In real-world applications we have no direct knowledge of the values of the conditional probabilities, i.e., we are unable to specify the probability of an event given all the possible combinations of history. For this reason we must approximate the Bayes optimality in general. In particular, we will be concerned with estimating $Pr\{y = 1|H(t) = x_t, t = 1, ..., k\}$ using approximate approaches.

Given an event trace, some events may appear suspicious and some not. Hence, detection systems must have a scoring function that combines the evidence. The output of a function is interpreted as the degree of suspicion attributed to the event trace. Although any two scoring functions need not be exactly the same, we can specify the conditions that any reasonable scoring function must satisfy. The class defined below appears to be both natural and general.

The detection system can employ a *scoring function* f that interprets events to produce a score characterizing the overall suspicion of the trace. Given a threshold value τ and an event trace $\mathbf{x}^{(k)}$ we can classify $\mathbf{x}^{(k)}$ as suspicious if $f(\mathbf{x}^{(k)}) \geq \tau$.

Definition 5. A scoring function f over a trace of events $\mathbf{x}^{(k)}$ is a function

$$f : \bigcup_{k=1}^{K} \mathbf{x}^{(k)} \to \mathbb{R}$$

The function f assigns a real value to any trace $\mathbf{x}^{(k)}$ of length $k = 1, ..., K$.

Let $\Delta(x_t)$ decide whether a single event x_t is suspicious or not

$$\Delta(x_t) = \begin{cases} 1; & \text{if } s'(x_t) \geq \tau' \\ 0; & \text{else} \end{cases}, \tag{5}$$

$$s'(x_t) = \frac{\lambda \cdot s(x_t)}{\lambda \cdot s(x_t) + (1 - \lambda) \cdot n(x_t)}. \tag{6}$$

Definition 6. A class of *well-behaved* functions consist of scoring functions s.t. $\forall \mathbf{x}^{(k)}, x_{k+1}$:

$$\begin{aligned} f(\mathbf{x}^{(k)}, x_{k+1}) &\geq f(\mathbf{x}^{(k)}) \qquad \text{if } \Delta(x_{k+1}) = 1, \\ f(\mathbf{x}^{(k)}, x_{k+1}) &\leq f(\mathbf{x}^{(k)}) \qquad \text{if } \Delta(x_{k+1}) = 0. \end{aligned}$$

The conditions imply that: (i) the scoring function f's evaluation increases when a new suspicious event is added to the trace and (ii) decreases when a normal event is added to the trace. The well-behaved scoring functions are motivated by the key observation that a suspicious event x_{k+1} (i.e., $\Delta(x_{k+1}) = 1$) is more likely to be generated by a suspicious process S than a normal process N, regardless of the history $\mathbf{x}^{(k)}$, i.e.,

$$\begin{aligned} s(x_{k+1}|\mathbf{x}^{(k)}) &\geq n(x_{k+1}|\mathbf{x}^{(k)}) \qquad \text{if } \Delta(x_{k+1}) = 1 \text{ and} \\ s(x_{k+1}|\mathbf{x}^{(k)}) &\leq n(x_{k+1}|\mathbf{x}^{(k)}) \qquad \text{if } \Delta(x_{k+1}) = 0. \end{aligned}$$

5. DETECTORS

In this section we analyze the approaches that decide whether an event trace is suspicious. First, we discuss the naive Bayes detector that relaxes the initial assumptions. Next, we discuss an approach that directly tackles the problem of estimating the likelihood that a trace was generated by a suspicious process using HMMs. Finally, we analyze an approach based on plan recognition and present two extensions: (1) we define utilities as a potential function; and (2) we present an observation utility function able to address non-linear accumulation.

5.1 Naive Bayes Detector

A naive approach assumes that events are independent, which means that the current event depends only on the current time step t and not on the time steps prior to t. The evaluation of Eq. (3) is simplified using the naive assumption:

$$Pr\{y = 1|H(t) = x_t, t = 1, ..., k\} = \frac{\lambda \cdot \prod_{t=1}^{k} \hat{s}(x_t)}{\lambda \cdot \prod_{i=1}^{k} \hat{s}(x_t) + (1 - \lambda) \cdot \prod_{i=1}^{k} \hat{n}(x_t)} \tag{7}$$

We have to evaluate the probability $Pr\{H(t) = x_t|y_t\}$ that an event is generated by a normal process $\hat{n}(x_t)$ and a suspicious process $\hat{s}(x_t)$, which is tractable in terms of evaluation. The approaches for estimating $\hat{n}$ and $\hat{s}$ may include a frequentist estimator, hidden Markov models, k-nearest neighbors, neural networks, etc. We showed an approach using CHMM in Section 3. An evaluation of the event trace is also well behaved when $\tau' = \lambda$.

In practice, the assumptions may oversimplify the model; however, we will use it as a baseline in our experiments.

5.2 Hidden Markov Models

An estimation of the conditional probabilities including the history can be encoded with hidden Markov models (HMMs) [13]. A HMM is a temporal probabilistic model with two embedded stochastic processes: an unobservable (hidden) process Q, which can be observed only through another (visible) stochastic process O. Each state in Q has state-transition probabilities (which are visible) and a probability distribution over the possible values of O. The key assumption is that the current hidden state of the agent is affected only by its previous state.

Now suppose we create a HMM to estimate $Pr\{H(t) = x_t|y = 1, t = 1, ..., k\}$, more precisely, it models the probability that a trace of events is generated by a suspicious agent. The hidden states of the process Q may be referred to as internal states presenting the intentions of the suspicious agent. For the sake of clarity, let us assume only two hidden states: a normal intention and a suspicious intention, emitting normal and suspicious events, respectively. The transitions between the hidden states can be explained as probabilities that the agent will either follow or change its current intention. Informally, this switching of intentions may be interpreted as follows: from an observer's perspective, sometimes suggesting that the observed agent is switching intentions appears to provide a better explanation of the behaviors.

We construct two HMM models: a normal model $\bar{N}$ and a suspicious model $\bar{S}$. We split all the labeled traces $\mathbf{x} \in D_l$ to traces generated by normal and suspicious agents, and use them to learn the parameters of the models $\bar{N}$ and $\bar{S}$, respectively. The model parameters can be locally optimized using an iterative procedure such as Baum-Welch method [13]. Given a new event trace $\mathbf{x}^{(k)} = (x_1, x_2, ..., x_k)$ we compute the probability that the trace was generated by each model $Pr\{\mathbf{x}^{(x)}|\bar{N}\}$ and $Pr\{\mathbf{x}^{(x)}|\bar{S}\}$ using a forward-backward procedure [13]. Given the prior probability $\bar{\lambda}$ we compute an estimate the trace $\mathbf{x}^{(k)}$ was generated by the suspicious process S:

$$Pr\{y = 1|H(t) = x_t, t = 1, ..., k\} = \frac{\bar{\lambda} \cdot Pr\{\mathbf{x}^{(k)}|\bar{S}\}}{\bar{\lambda} \cdot Pr\{\mathbf{x}^{(k)}|\bar{S}\} + (1 - \bar{\lambda}) \cdot Pr\{\mathbf{x}^{(k)}|\bar{N}\}}. \tag{8}$$

Although the information about previous behavior is now partially encoded in the transition probabilities (i.e., given the agent's intention at time step t is suspicious it is more likely that the intention at $t + 1$ will be suspicious as well), the model still uses the Markov assumption, i.e., the next agent's intention depends only on it's current intention. It is possible to introduce more complex HMM structures with long-term dependencies, but learning and inference in such models become computationally intractable [11].

5.3 Utility-Based Plan Recognition

We exploit UPR, an *Utility-based Plan Recognition*, briefly described below. The reader is referred to [3] for details. UPR consists of a plan library, which encodes behaviors of the observed agents in a form of directed graph, and a matching algorithm. It follows the footsteps of the hierarchical HMM in representing probabilistic information in the plan library. A plan step can be atomic, or non-atomic, i.e., broken down into atomic sub-steps, each a plan step in itself. Plan steps are linked via sequential edges, describing the execution order of a given plan and its sub-steps. UPR introduces three types of utilities on the edges: (a) the sequential utility from the current step to the next; (b) the interruption utility from the current step to the end of the plan; and (c) the decomposition utility from the current step at current level to its first substep at the sub-level. A corresponding probability is maintained for each type of utility. The observation sequence o is matched against the library using a *Symbolic Plan Recognizer* [2], which filters hypotheses that are consistent with o. Finally, the hypotheses are ranked by their expected utility.

We use a heuristic version of UPR as follows. Let $\hat{s}(x_t) = 1 - \hat{n}(x_t)$ be the probability that the trigger event x_t was generated by a suspicious person. Let $c_s > 0$ be the cost of the damage caused by a suspicious person if we do not stop him, and similarly, let $d_n = 0$ be the cost of the damage caused by a normal person. The expected cost of letting this person go (marking him as normal) is $c_{go} = c_s\hat{s}(x_t) + d_n\hat{n}(x_t) = c_s\hat{s}(x_t)$. Now suppose $c_n > 0$ is the cost of arresting an innocent person and $d_s = 0$ is the cost of the damage caused by a suspicious person when arrested. The expected cost of stopping this person (marking him as suspicious) is $c_{stop} = c_n\hat{n}(x_t) + d_s\hat{s}(x_t) = c_n\hat{n}(x_t)$. If there was only one event, we would compare both hypotheses and choose the one with the lowest expected cost. Supposing in this case $c_n\hat{n}(x_t)$ is lower, we would call this person suspicious.

One possible approach, based on the above expected-cost calculation, would be to determine whether a trigger event is to be categorized as suspicious or normal, and then to accumulate the total number of suspicious events, and subtract the total number of normal events; unfortunately, this simple strategy performs poorly. Therefore, not only do we count whether an event is suspicious or normal, but we give it a weight, proportional to the benefit or cost accrued. The function U_{UPR} hence evaluates an event trace $\mathbf{x}^{(k)}$ of a person by accumulating the weighted benefit of stopping this person and subtracting the weighted cost of arresting a normal person:

$$
\begin{aligned}
U_{UPR}(\mathbf{x}^{(k)}) &= \sum_{t=1}^{k} b(x_t), && (9)\\
b(x_t) &= \begin{cases} c_s\hat{s}(x_t); & \text{if } c_n\hat{n}(x_t) \leq c_s\hat{s}(x_t) \\ -c_n\hat{n}(x_t); & \text{if } c_n\hat{n}(x_t) > c_s\hat{s}(x_t) \end{cases}. && (10)
\end{aligned}
$$

If the accumulated cost exceeds a threshold value τ', the person (i.e., trace $\mathbf{x}^{(k)}$) is marked as suspicious.

This remains a heuristic approach and further investigations could be a topic for future work; however, given that our next approach performs significantly superior, we chose to investigate that in more detail rather than providing more heuristics for the current approach.

5.3.1 Utilities as Potential Functions

Although the evaluation function U_{UPR} is well behaved, the utilities are constant and hence do not allow a dynamic adjustment to the behavior of the agent in the past. Thus, for instance, the first time we note a suspicious event, and the second time we note the same agent making a suspicious event, count equally. These utilities, however, are unable to express the characteristics of the empirical observations. Therefore, we extend the notion of utility and define the utility U as follows.

Definition 7. The utility function U over a plan step q_a, a plan step q_b, and the entire observation sequence $\mathbf{x}^{(t)}$ until current time step t is a function

$$U : \langle q_a, q_b, \mathbf{x}^{(t)} \rangle^n \rightarrow \mathbb{R}.$$

Utility function can be written as

$$U(q_a, q_b, \mathbf{x}^{(t)}) = \sum_{j=1}^{n} \lambda_j u_j(q_a, q_b, \mathbf{x}^{(t)}),$$

where each utility function u_j can be sequential, interruption, decomposition or any other utility, and λ_j are parameters to be defined. This allows us to introduce a set of auxiliary utility functions u_j describing not only the plan-step transitions but also the additional characteristics of the observation sequence. For example, the sequential utility from step q_i to q_{i+1} can be written as $u_t(q_i, q_{i+1}, \mathbf{x}^{(t)}) = c$, but in general, the constant c can be replaced with any function over q_i, q_{i+1} and $\mathbf{x}^{(t)}$.

Lemma 1. U is a well behaved function iff

$$\forall u_j, j = 1...k : u_j \text{ is well a behaved function.}$$

PROOF. Consider two well behaved functions f and g, and two scalar constants λ_f and λ_g. Let $f' = \lambda_f f$. Since multiplication with scalar preserves well-behaved property, f' is also a well behaved function. Let function u denote $u = f' + g'$. Then, $u(\mathbf{x}^{(t)}, x_{t+1}) = f'(\mathbf{x}^{(t)}, x_{t+1}) + g'(\mathbf{x}^{(t)}, x_{t+1}) \geq u(\mathbf{x}^{(t)}) = f'(\mathbf{x}^{(t)}) + g'(\mathbf{x}^{(t)})$ if $\Delta(x_{t+1} = 1)$, since f and g are well behaved and therefore $f'(\mathbf{x}^{(t)}, x_{t+1})$ and $g'(\mathbf{x}^{(t)}), x_{t+1})$ are non-negative. Similarly, f' and g' are non-positive when $\Delta(x_{t+1}) = 0$. □

5.3.2 Observation Utility for Suspicious Behavior Detection

In order to include the past behavior of an agent in an evaluation of the evidence, the utility function must be defined over the observation sequence. We propose an observation utility function that assigns cost using the number of normal and suspicious events in the past. Consider the example from Section 4. Suppose we see a person do a full U-turn in front of a police officer and we give this event a cost of 1. Later we see the same person doing a half-turn in front of a police officer. This event if seen on its own, would be given cost 0.5. However, following this initial turn where we had given a cost of 1, this new turn, becomes a 1 instead of 0.5. So, a linear accumulation would have given us a cost of 1.5, whereas because we bias the new event to register higher on our scale, our cost is 2 instead of 1.5.

Let $\eta_s(\mathbf{x}^{(k)})$ define the number of suspicious events in an event trace $\mathbf{x}^{(k)}$:

$$\eta_s(\mathbf{x}^{(k)}) = \sum_{t=1}^{k} \Delta(x_t), \qquad (11)$$

Similarly, let $\eta_n(\mathbf{x}^{(k)}) = k - \eta_s(\mathbf{x}^{(k)})$ represent the number of normal events. Suppose we observed a trace $\mathbf{x}^{(k)}$ of all the suspicious events, i.e., $\forall t, t = 1, ..., k : \Delta(x_t) = 1$. Intuitively, the likelihood that an event x_t was indeed generated by a suspicious process increases exponentially according to the number of suspicious events in the past. On the other hand, if the events in $\mathbf{x}$ were normal, i.e., $\forall t, t = 1, ..., k : \Delta(x_t) = 0$, the likelihood exponentially decreases as the number of normal events increases. We define an observation utility function u_o over the current event x_t and trace $\mathbf{x}^{(t-1)}$ recursively as follows:

$$
\begin{aligned}
u_o(x_t, \mathbf{x}^{(t-1)}) &= \psi(\mathbf{x}^{(t)}) \cdot (u_o(\mathbf{x}^{(t-1)}) + \omega(\mathbf{x}^{(t)})), && (12)\\
u_o(\mathbf{x}^{(0)}) &= 0, &&\\
\omega(\mathbf{x}^{(t)}) &= \alpha \cdot \eta_s(\mathbf{x}^{(t)})^{s(x_t)/\beta}, && (13)\\
\psi(\mathbf{x}^{(t)}) &= \gamma \cdot \rho^{-\eta_n^*(\mathbf{x}^{(t)})/\eta_s(\mathbf{x}^{(t)})}. && (14)
\end{aligned}
$$

The term $\omega(\mathbf{x}^{(t)})$ uses an exponential function to assign a cost to the likelihood $s(x_t)$ that an event is suspicious. The parameter $\alpha > 0$ is the initial cost, η_s corresponds to the growth factor, and the parameter $0 < \beta < 1$ is the likelihood required for the cost to increase by the growth factor. The parameters α and β are estimated from the data. Suppose we observe two full U-turns, the second U-turn attributes higher cost to the overall suspicion, since the exponent base is increased due to the first U-turn.

Additionally, the term $\psi(\mathbf{x}^{(t)})$ employs an exponential time decay function that discounts the accumulated cost at time t according to the number of consecutive normal events η_n^*. The modified η_n^* represents *the time elapsed* since the last event $\Delta(x_i) = 1$, i.e., the number of normal events since the last suspicious event. The higher the number of consecutive normal events, the faster the cost decay. The parameter $0 < \gamma \leq 1$ is the initial decay, the parameter $0 < \rho < 1$ is the decay factor, and η_s is used to specify the number of events required for the decay to decrease by the decay factor. The parameters γ and ρ are also estimated from the data. Suppose we observe two agents, one already having made two U-turns and the other with only one U-turn. Suppose we observe both agents do a clearly normal event. The overall suspicion of the first agent is reduced less than the overall suspicion of the second agent. Hence the higher the number of suspicious events, the slower the suspicion decay.

The function u_o is a well-behaved function by definition. Eq. (12) can be rewritten, which gives us the utility function U_{F-UPR}:

$$
\begin{aligned}
U_{F-UPR}(\mathbf{x}^{(k)}) &= \sum_{t=1}^{k} \sum_{j=1}^{n} \lambda_j f_j(\mathbf{x}^{(t)}, q(t-i), q(t))\\
&= \sum_{t=1}^{k} (\omega(\mathbf{x}^{(t)}) \prod_{i=t}^{k} \psi(\mathbf{x}^{(i)})). \qquad (15)
\end{aligned}
$$

6. EXPERIMENTAL EVALUATION

We conducted empirical tests in a simulated airport domain to evaluate the performance of suspicious-passenger detection generated by four candidate algorithms. In addition, we compared the best two algorithms on the dangerous-driver domain [2].

To run proof-of-concept tests we considered a simulated environment, mainly to avoid difficulties due to privacy and confidentiality issues, and as well as due to the absence of real-world annotated data of suspicious behavior. A simulator also made it possible to control the amount of noise otherwise introduced by various vision systems (occlusions, false detections, etc.), and provided controllable and repeatable situations.

6.1 Airport domain

The experiments in this paper use the ESCAPES [15], a state-of-the-art, multiagent simulator for airport evacuations with several types of agents exhibiting behaviors of regular travelers, authorities, and families. The agents' behavior incorporates emotional, informational and behavioral interactions, such as emotional contagion, the spread of knowledge/fear, social comparison, etc. Therefore, an agent is affected by the behavior of other agents and their emotional states, and faced with uncertainty as to what happened and where the nearest exits are. We assume that the behavior of the agents corresponds to the behavior of real passengers at the airport.

In cooperation with security officials we defined a scenario where a suspicious passenger goes from point A to point B while trying to avoid security personnel at the airport. One may argue that an adversary that plans to do something malicious would behave normally in the presence of authorities, and this might be true for a highly trained individual. As discussed previously, an average person exposed to a high level of stress produces behavior that indicates fear, anxiety, tension, etc., and hence tries to cover it by minimizing close-range interactions by making u-turns, avoidance maneuvers, hiding in nearby shops, etc. Implementation details are provided on a supplemental web page[1].

A simulation in ESCAPES is run with a given airport map, authority agents, regular passengers and a suspicious agent going from point A to B, outputting traces with 2D coordinates for all agents. We initialized the simulator with 100 agents including $K_a \in \{5, 10, 15, 20, 25\}$ authorities and a suspicious person with randomly chosen initial and final points. For each K_a setting we ran 30 simulations, each consisting of $1500 - 3000$ time steps and 100 traces. On average, there were 215 interactions between the authorities and the passengers per run. To avoid issues that arise with highly unbalanced datasets we used random re-sampling without replacement to balance the data to the ratio *suspicious : normal* $= 20 : 80$.

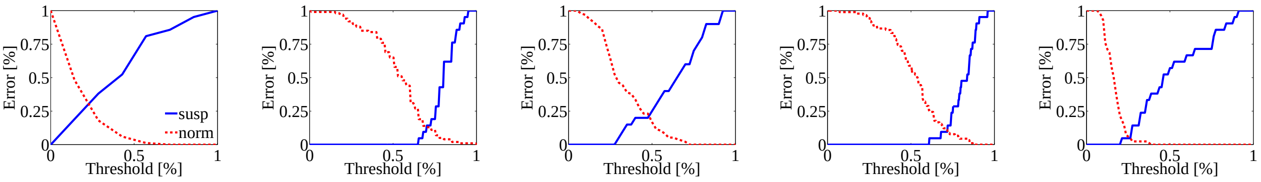
The trace of the coordinates was preprocessed to the action trace as follows. A change in position from the previous to the current state was described as taking the action of moving North, South, East and West, and their combinations (nine in total). This transformation describes the shape of a trajectory but discards the location information, which leads to better generalization. We also experimented with other transformations, for example, a more general one that also discards the orientation (forward, backward, left, right), and a less general one that divides the airport map with a square-based grid with numbered squares [2]. Preliminary tests showed the best performance when using the first transformation.

For the evaluation we used *precision*, *recall*, *specificity* and *F-measure*. Precision is defined as the number of true positives (all suspicious cases correctly classified as suspicious) divided by the number of all cases marked as suspicious (true and false positives): $pr = TP/(TP+FP)$. A perfect score 1 means that all cases marked as suspicious were indeed suspicious. Hence, the score $1 - pr$ represents the rate of *false alarms*. Recall is defined as the number of true positives divided by the number of all the suspicious cases: $re = TP/(TP+FN)$. A perfect score 1 means that all the suspicious cases were detected (but says nothing about falsely marked normal cases). Similarly, the specificity is defined for normal cases $sp = TN/(TN+FP)$. There are two points of interest, depending on our objective. The first one is when both scores are minimized, i.e., the trade-off point between false alarms and non-detected suspicious passengers, which can be detected with the F-measure $FM = 2 \cdot pr \cdot re/(pr + re)$. The other case is when a high false-alarm rate is acceptable and non-detected cases are extremely costly. In this case we are interested in precision when recall $re = 1$, i.e., all the suspicious passengers are found. In the worst-case scenario, all the passengers are marked as suspicious. We evaluate the statistical significance of our results using the two-sample t-test.

6.1.1 Results

In the first experiment we fixed the number of authority figures

[1] http://dis.ijs.si/bostjan/aamas2012



(a) $\exists k$ rule (b) Naive Bayes (c) HMMs (d) UPR (e) F-UPR

Figure 3: Confusion error rates for different threshold values.

$K_a = 10$. We instantiated the naive Bayes, HMMs, UPR, and F-UPR detectors. Additionally, we considered another baseline detector using a simple rule over the threshold k and the event trace $\mathbf{x}^{(t)}$, saying that if the number of suspicious events exceeds k (i.e., $\exists k : \eta_s(\mathbf{x}^{(t)}) > k$), then mark trace $\mathbf{x}^{(t)}$ as suspicious. All the detectors used the event-trace probabilities $s'(\mathbf{x}^{(t)})$ and $n' = 1 - s'(\mathbf{x}^{(t)})$ as returned by the event-detection step. For the HMM approach we considered two ergodic HMMs as described is Section 6.1.2. We used two observations, the normal $\Delta(x_t) = 0$ and the suspicious $\Delta(x_t) = 1$ event, and varied the number of hidden states. The best results were achieved with three hidden states. Note that the HMMs detector applied on top of the CHMMs detector basically presents a version of the mixed layered HMM structure. All the models (including UPR and F-UPR detectors) were evaluated with 10-fold-cross validation.

Figures 3(a)–3(e) show the confusion error rates for suspicious (1-recall) and normal (1-specificity) passengers as a function of the normalized threshold value for all the five algorithms. For example, if the threshold is zero, then all the passengers are marked as suspicious. In this case: (i) all the suspicious passengers are correctly identified as suspicious, hence the error rate is also zero; and (ii) all the normal passengers are incorrectly identified as suspicious, hence the error rate is 1. As the threshold value increases, the error rate for correctly identifying the suspicious passengers increases, while the error rate for correctly identifying the normal passengers decreases.

There are two points of interest: (i) when the error rates cross each other, i.e., the F-measure is maximized; and (ii) the rightmost point when the error rate for suspicious passengers is zero (i.e., $re = 1$) and the other one is minimized. These cases are tabulated in Table 1. The first case is summarized in columns 2-4 showing the recall, precision and F-measure. F-UPR outperforms the $\exists k$ rule ($p < 0.01$), naive Bayes ($p < 0.01$), HMMs ($p < 0.01$), and UPR ($p < 0.01$). The second case, where the threshold value is such that all the suspicious passengers are discovered, is shown in columns 5-6. Column five shows the confusion error for normal passengers (i.e., 1-specificity), while the column six shows the ratio of correctly raised alarms (i.e., precision). The $\exists k$ rule, for instance, marks all the passengers as suspicious (FP rate is 100%) and consequentially almost 80% of alarms are false. HMMs achieve better performance, but still mark more than 50% of normal passengers as suspicious. Other methods mark between 1/5 and 1/4 of normal passengers as suspicious, but precision is around 50%, which means that every second passenger marked as suspicious is indeed suspicious (and all suspicious passengers are discovered!). Overall, F-UPR in this setting also outperforms the $\exists k$ rule ($p < 0.01$), naive Bayes ($p < 0.05$), HMMs ($p < 0.01$), and UPR ($p < 0.05$). Finally, Figure 4 depicts the ROC curves showing that F-UPR performs the same or better in all the threshold settings.

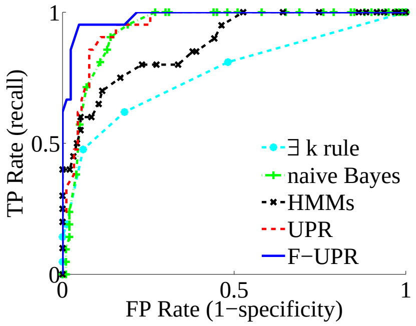
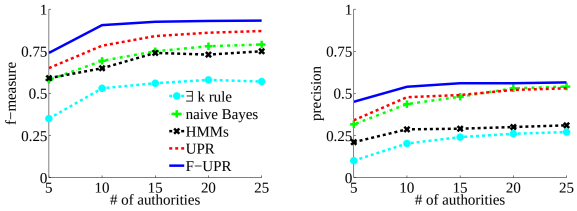


Figure 4: ROC curves comparing all the detectors.

Table 1: Evaluation results when the F-measure is maximized (columns 2-4) and all the suspicious cases are discovered (last two columns).

	max FM			re=1	
Algorithm	re	pr	FM	1-spec	pr
$\exists k$ rule	0.619	0.464	0.530	1.000	0.202
Naive Bayes	0.857	0.581	0.693	0.270	0.436
HMMs	0.600	0.706	0.649	0.526	0.286
UPR	0.857	0.720	0.783	0.256	0.477
F-UPR	0.905	0.905	0.905	0.217	0.539



(a) F-measure is maximized. (b) All suspicious passengers are discovered.

Figure 5: Evaluation results for varying the number of authority figures in the simulation and two different threshold values.

In the last experiment we varied the number of authorities in the simulation. We expect that an increased number of authority figures will result in more interactions between the suspicious passengers and the authorities, which will make detection easier. Figure 5 shows the results for the $K_a \in \{5, 10, 15, 20, 25\}$ authority figures in a simulation: Fig. 5(a) shows the F-measure for a threshold such that the F-measure is maximized, while Fig. 5(b) shows the precision when $re = 1$. An increased number of authority figures first significantly increases the detection capabilities. For example, the F-measure for F-UPR increases by 15% when the security resources are doubled from five to ten, but as the number increases, the impact is smaller. We can also see that F-UPR achieves the same performance as other methods using significantly less security resources.

6.1.2 *Detection Based on the Action Trace*

We also applied a sanity check and tested the suspicious behavior detection from a sequence of agent's actions (i.e., action trace **a**) instead of a sequence of trigger events (i.e., event trace **x**). We used HMMs, since they are considered as a baseline for modeling a sequence of actions. The goal is to differentiate between a sequence of actions produced by a suspicious and a regular passenger. We expect this approach not to perform well, since it is too general and unable to precisely model the interactive behavior present in a multiagent environment.

The suspicious behavior detector consists of two ergodic HMMs: S' trained on the suspicious and N' trained on the regular action traces. A new trace is first transformed to the action trace $\mathbf{a}^{(k)}$ as described previously and then matched against both HMMs, yielding the likelihood that it produced the given $\mathbf{a}^{(k)}$. If the likelihood is greater than a threshold the action trace is marked as suspicious. We tested this approach for $K_a = 10$. At the threshold value s.t. the highest F-measure of 18.01 was achieved this approach achieved an acceptable discovery rate ($re = 66.23$) and an extremely low precision ($pr = 10.42$). Such a performance positions this approach under the $\exists k$ rule. The overall performance was consistent with our expectations. Modeling single-agent actions in a multiagent environment is not able to capture the interactive behavior.

6.2 Catching a Dangerous Driver

In addition to the airport domain we applied UPR and F-UPR to the dangerous-driver domain, as introduced in [2]. This domain also includes behavior that becomes increasingly costly if repeated; a driver switching a lane once or twice is not necessarily acting suspiciously, but a driver zigzagging across two lanes is dangerous. Our goal was to detect such drivers as soon as possible.

We generated 100 observation sequences (each of N observations) of a zigzagging driver, and 1000 sequences of a safe driver. The observations were sampled with 10% noise from the trajectories. If the driver stayed on the same lane as in the previous sample, the event was considered as normal, otherwise it was considered as dangerous. For each sequence of trigger events we accumulated the associated cost using both UPR and F-UPR.

Table 2 reports the performance at the peak F-measure for different lengths of the observation sequence. The results confirm the experiments on the airport domain for two points. First, F-UPR performs better than UPR for any selected sequence length. Second, the performance of both methods increases as the number of observations increases, where F-UPR requires fewer observations than UPR to achieve the same performance.

Table 2: Performance at the peak F-measure in dangerous driver domain.

Sequence length N	F-UPR	UPR
25	0.632	0.540
50	0.720	0.667
75	0.900	0.800
100	0.952	0.857
125	1.000	0.947

7. CONCLUSION

This paper successfully addressed the problem of suspicious behavior detection from a set of observations, where no single observation suffices to make the decision. The paper addresses the problem in two steps, i.e., the detection of trigger events and a combination of evidence to reach the final decision. To that end, the main contributions of this paper are: (i) the conditions that a reasonable detector should satisfy; (ii) an analysis of three detectors; (iii) a novel F-UPR approach that extends the notion of utilities; and (iv) comprehensive experiments on two simulated domains. By providing a new algorithm that outperforms other approaches, this paper has advanced the state of the art.

8. REFERENCES

[1] D. Arsić, B. Schuller, and G. Rigoll. Suspicious behavior detection in public transport by fusion of low-level video descriptors. In *Proc. of the 8th ICME*, pages 218–221, 2007.

[2] D. Avrahami-Zilberbrand. *Efficient Hybrid Algorithms for Plan Recognition and Detection of Suspicious and Anomalous Behavior*. PhD thesis, Bar-Ilan University, 2009.

[3] D. Avrahami-Zilberbrand and G. A. Kaminka. Incorporating observer biases in keyhole plan recognition (efficiently!). In *AAAI*, 2007.

[4] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *CVPR*, pages 994 – 999, 1997.

[5] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-Markov model. In *CVPR*, pages 838–845, 2005.

[6] F. Esponda, S. Forrest, and P. Helman. A formal framework for positive and negative detection schemes. *IEEE Systems Man and Cybernetics Society*, 34(1):357–373, 2004.

[7] R. S. Feris, A. Hampapur, Y. Zhai, R. Bobbitt, L. Brown, D. A. Vaquero, Y. li Tian, H. Liu, and M.-T. Sun. *IBM Smart Surveillance System*. Taylor & Francis Group, 2009.

[8] C. W. Geib and R. P. Goldman. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11):1101–1132, 2009.

[9] P. Helman and G. Liepins. Statistical foundations of audit trail analysis for the detection of computer misuse. *IEEE Transactions on Software Engineering*, 19(9):886–901, 1993.

[10] S. Hongeng and R. Nevatia. Large-scale event detection using semi-hidden Markov models. In *IEEE Int. Conf. on Computer Vision*, pages 1455–1462, Aug. 2003.

[11] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[12] M. Naylor and C. I. Attwood. Advisor: Annotated digital video for intelligent surveillance and optimised retrieval, 2003. Final report.

[13] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *IEEE*, 1989.

[14] G. Sukthankar and K. Sycara. Hypothesis pruning and ranking for large plan recognition problems. In *AAAI*, 2008.

[15] J. Tsai, G. Kaminka, S. Epstein, A. Zilka, I. Rika, X. Wang, A. Ogden, M. Brown, N. Fridman, M. Taylor, E. Bowring, S. Marsella, M. Tambe, and A. Sheel. ESCAPES - Evacuation simulation with children, authorities, parents, emotions, and social comparison. In *AAMAS*, 2011.

[16] N. Vaswani, A. R. Chowdhury, and R. Chellappa. Shape activity: A continuous state HMM for moving/deforming shapes with application to abnormal activity detection. In *IEEE Trans. on Image Processing*, pages 1603 – 1616, 2005.