Modeling and Reasoning about Success, Failure, and Intent of Multi-Agent Activities

Adam Sadilek
University of Rochester
sadilek@cs.rochester.edu

Henry Kautz
University of Rochester
kautz@cs.rochester.edu

ABSTRACT

Recent research has shown that surprisingly rich models of human activity can be learned from GPS (positional) data. However, most effort to date has concentrated on modeling single individuals or statistical properties of groups of individuals. We, in contrast, take on the task of understanding human interactions, attempted interactions, and intentions from noisy sensor data in a multi-agent setting. We use a real-world game of capture the flag to illustrate our approach in a well-defined domain. Our evaluation shows that given a model of successfully performed multi-agent activities, along with a set of examples of failed attempts at the same activities, our system can automatically learn an augmented model that is capable of recognizing success, failure, as well as goals of people's actions with high accuracy. Finally, we demonstrate that explicitly modeling unsuccessful attempts boosts performance on other important recognition tasks.

Author Keywords Multi-agent activity recognition, statistical relational learning, structure learning.

ACM Classification Keywords I.2.4 [Knowledge Representation Formalisms and Methods (F.4.1)]: Relation systems.

General Terms Algorithms, Human Factors, Experimentation.

MOTIVATION

Our society is founded on the interplay of human relationships and interactions. Since every person is tightly embedded in our social structure, the vast majority of human behavior can be fully understood only in the context of actions of other—related—people. Being able to recognize people's activities and reason about their behavior is a necessary precondition for having intelligent and helpful machines that are aware of "what is going on" in the human-machine as well as human-human relationships.

Furthermore, reasoning about human *intentions* is an essential element of context-awareness, since if we can recognize what a person (or a group of people) *wants* to do, we can proactively try to help them (or—in adversarial situations—hinder them). Intent is notoriously problematic to quantify (*e.g.*, [1]), but we show that the notion is precisely and naturally captured in the process of learning the structure of unsuccessfully attempted activities. We all know perhaps too well that a successful action is often preceded—and unfortunately sometimes also followed—by multiple failed attempts. Therefore, reasoning about attempts typically entails high practical utility, but not just for their relatively high frequency. Consider, for example, a task of real-time analysis of a security video system. There, detecting that a person or a group of people (again, relations) *intend* to steal something is much more important and useful than recognizing that a theft has taken (or even is taking) place, because then it is certainly too late to entirely *prevent* the incident and it may also be too late or harder to merely stop it. We believe that recognition of attempts in people's activities is a severely underrepresented topic in artificial intelligence that needs to be explored more since it opens a new realm of interesting possibilities.

RELATED WORK

More and more evidence is emerging from social networks research showing that when we want to model behavior of a person, the single best predictor is often the behavior of people in her social network (*e.g.*, [5]). However, previous work heavily focused on nonrelational reasoning and activity recognition [4, 2]. A number of researchers in machine vision have worked on the problem of recognizing events in videos of sporting events, such as impressive recent work on learning models of baseball plays [3]. Most work in that area has focused on recognizing individual actions (*e.g.*, catching and throwing), and the state of the art is just beginning to consider relational actions (*e.g.*, the ball is thrown from player A to player B).

In our previous work on multi-agent activity recognition, we show that while it may be impossible to directly detect an activity due to sensor noise, the occurrence of the activity can still be deduced by its impact on both the past and future behaviors of the individuals it involves [7]. However, that model recognizes only single and statically defined activity. In order to build a truly useful reasoning system, we need to consider not only relationships among people, but also relationships among activities themselves. Here we improve our earlier approach in several directions: We extend the number of activities in the recognition repertoire of the system; we automatically extract people's intentions from their actions; and we show how the system adapts itself and learns to recognize failed attempts at activities and in turn uses the newly acquired knowledge to label both failed and successful events. To our knowledge, there exists no prior work on explicit modeling and recognition of attempted activities or on learning the intended purpose of an activity in a multi-

agent setting.

CAPTURE THE FLAG DOMAIN

Our main test domain is the game of capture the flag (CTF). We collected a CTF dataset by having people play four rounds of the game on a university campus. Each player carried a consumer-grade GPS device that logged its position (plus noise) every second (Fig. 1). There are two teams—each consisting of seven players. The goal is to enter the opponent's flag area. Players can be captured only while on enemy territory by being tagged by the enemy. Upon being captured, they must remain in place until freed (tagged by a teammate) or the game ends. The games involve many competitive and cooperative activities, but here we focus only on capturing and freeing. The visualization of the games is available from the first author's website.

If we are to reliably recognize events that happen in these games in the presence of severe noise, we need to consider not only each player individually but also the relationships among them over extended periods of time (possibly the whole length of the game). Consider the task of inferring the individual and joint activities and intentions of the CTF players from their GPS traces. For example, suppose the GPS data shows Player A running toward a stationary teammate Player B, then moving away. What occurred? Possibly Player A has just "freed" Player B, but GPS error has hidden the fact that Player A actually *reached* B. Another possibility is that Player A had the *intention* of freeing Player B, but was scared off by an opponent at the last second. Yet another possibility is that no freeing occurred or was even intended, because Player B had not been previously captured.

Understanding a game thus consists of inferring a complex set of interactions among the various players as well as players' intentions. The conclusions drawn about what occurs at one point in time affect and are affected by inferences about past and future events. In the example just given, recognizing that Player B is moving in the future reinforces the conclusion that Player A is freeing Player B, while failing to recognize a past event of Player B being captured decreases confidence in that conclusion. The game of CTF also illustrates that understanding a situation is as much or more about recognizing attempts and intentions as about recognizing successfully executed actions. For example, in course of a 15 minute game, only a handful of capture or freeing events occur. However, there are dozens of cases where one player unsuccessfully tries to capture an opponent or to free a teammate. A description of a game that was restricted to what actually occurred would be only a pale reflection of the original.

Although the CTF domain doesn't capture all the intricacies of life, it contains many complex, interesting, and yet well-defined (multi-agent) activities. Moreover, it is based on extensive real-world GPS data (total of 40,000 data points). Thus most of the problems that we are addressing here clearly have direct analogs in everyday-life situations that ubiquitous computing needs to address—imagine people going about their daily lives in a city instead of CTF players, and their own smart phones instead of GPS loggers.

Figure 1. A snapshot of a game of capture the flag prior data denoising.

BACKGROUND: MARKOV LOGIC

We cast our model of CTF in Markov logic (ML), a statistical-relational language that provides an elegant combination of probabilistic reasoning with expressive, relatively natural, and compact but strictly true or false formulas of first-order logic (FOL) [6]. The basic elements of Markov logic are (a) FOL formulas (*e.g.*, all dogs are animals: $\forall x\ \mathrm{dog}(x) \Rightarrow \mathrm{animal}(x)$); and (b) weights assigned to each formula. The relative importance of a formula is proportional to its weight. For example, $[\forall x\ \mathrm{animal}(x) \Rightarrow\ \mathrm{cat}(x)]\cdot 8.3$ means that if we know that x is an animal, with some well-defined probability (captured in the weight of 8.3) x is also a cat. Thus, the formulas allow us to express complex attributes, structure, and relationships while the weights present a way to model any such finite domain in a probabilistic way.

METHODOLOGY

We investigate the following research questions in a multi-agent setting:

Q1. Can models of attempted activities be automatically learned (induced) from models of successfully performed actions?

Q2. Does modeling attempts of activities improve performance of recognizing the activities themselves?

In this section, we describe three major components of our approach that, when put together, allow us to answer Q1 and Q2. In short, we first manually augment the model of capturing activities in CTF presented in [7] so that it recognizes freeing events as well. This constitutes our "seed" theory that our system then automatically extends in order to learn the structure of failed attempts at captures and freeings and the relationships among them and their successful counterparts. Finally, we use the augmented theory to recognize a richer set of multi-agent activities and extract their respec-

tive intended goals.

Adding Freeing Recognition

We augment the original model described in [7] by simply adding two additional rules:

1. If players *a* and *b* are allies, both are on enemy territory, *b* is currently captured, *a* is not, and they are close to each other, then *a probably* frees *b*.
2. At any given time, a player is either captured or free.

In the actual implementation, these rules are written in Markov logic. For example rule 2 becomes

$$\forall a, t : \left[\text{isFree}(a, t) \text{ XOR } \text{isCaptured}(a, t)\right] \cdot \infty.$$

Learning a Model of Failed Attempts

Next, the system executes the following algorithm in order to induce new ML formulas and learn a new set of weights. The algorithm produces a new model (ML theory) that recognizes both successfully performed capturing and freeing events as well as instances where the players attempted those activities but failed. Intuitively, the algorithm implements top-down learning to generate a seed hypothesis and then applies bottom-up (data-driven) learning to prune the seed theory, whereby the training data (user-provided examples of failed attempts) guides our search for formulas to remove.

Algorithm 1 : Extend a ML theory to model successful as well as failed activities.

Input: A: set of activities
$\mathcal{T}$: ML theory that models successful instances of activities in A
S: set of examples of successful activities
F: set of examples of failed activities

Output: $\mathcal{T}''$: augmented ML theory with learned weights that models both successful and attempted activities in A
$\mathcal{I}$: intended goals of the activities

1: $\mathcal{T}^2 \Leftarrow$ liftToSecondOrderML($\mathcal{T}$, A)
2: $\mathcal{T}' \Leftarrow$ instantiate($\mathcal{T}^2$, A)
3: $(\mathcal{T}'', \mathcal{I}) \Leftarrow$ findIncompatibleFormulas(F, $\mathcal{T}'$)
4: learnWeights(S, F, $\mathcal{T}''$)
5: **return** $\mathcal{T}'', \mathcal{I}$

In step 1 of the algorithm, we first variabilize the predicate names of the activities (capturing and freeing). This process is called lifting to second-order ML and allows our system to automatically introduce new activities, instead of working with a fixed set of pre-defined ones. In step 2, the system instantiates these predicate variables thereby introducing a new set of ML formulas, some of which model successful capturing and freeing, some model attempted capturing and freeing, and the rest do not capture anything of interest and will be pruned out in the following step. We use a standard Boolean satisfiability solver (miniSAT) to find the right formulas to remove from our model. Our system achieves that by testing which subsets of the formulas are compatible with the examples ($S \cup F$) in step 3. This step also extracts the intended goal for each activity in A by resolving the incompatibilities between the theory and the training examples. In step 4, we learn new weights within the augmented model $\mathcal{T}''$ as described in [7]. Our system also displays the consequences of each refinement of the model as a set of annotated animations so that the user can easily follow the learning procedure.

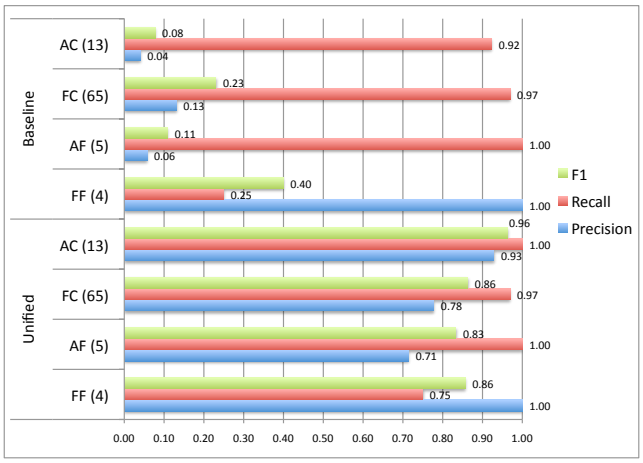

Figure 2. Performance of the baseline and unified models. In parentheses are the numbers of respective events in the dataset.

EXPERIMENTS AND RESULTS

To answer our research question Q1, we apply the theory augmentation process (Algorithm 1) on the CTF seed theory described in [7], augmented with freeing recognition as explained above. This induces a new set of formulas that capture the structure of attempted activities and ties them together with the existing formulas in the theory.

In the process of pruning formulas, Algorithm 1 correctly discovers that whether a player remains stationary or not is the key distinction between a successful and failed capturing (since players who were not actually captured can still move). This discovery is not hidden in the internal structure of the model, but is explicitly output in the language of Markov logic. Analogous process yields a fitting separation between failed and successful freeings. Namely, our model learns that an unsuccessfully freed player remains stationary. Note that in practice, the difference between success and failure in performing an activity directly constitutes the purpose of the activity and consequently the intent of the actors.

Next we use this newly learned *unified* model to recognize *all four* activities of interest: actual and failed capturing (AC and FC respectively), and actual and failed freeing (AF, FF). We compare the performance of our unified model to an alternative (baseline) method that labels all four activities in the following way. There are two separate stages. First we denoise (snap) the GPS data by creating an occupancy map of the game area as described in [7], and afterward we label the instances of the four activities. The following labeling rule is applied. We loop over the whole data set and look for instances where a pair of players *a* and *b* were snapped (in the first step) to either the same cell or to two adjacent cells at time *t*, they are enemies, *b* is not captured already, and *a* is on its home territory while *b* is not. If *b* moves (is snapped to a different cell at a later time) *without* having an ally nearby, we output *failedCapturing(a,b,t)*, otherwise we output *capturing(a,b,t)*. The labeling rule for freeing is defined analogously and all four events are tied together. The unified model is executed as described in section "Methods" above and is evaluated using four-fold cross-validation (al-

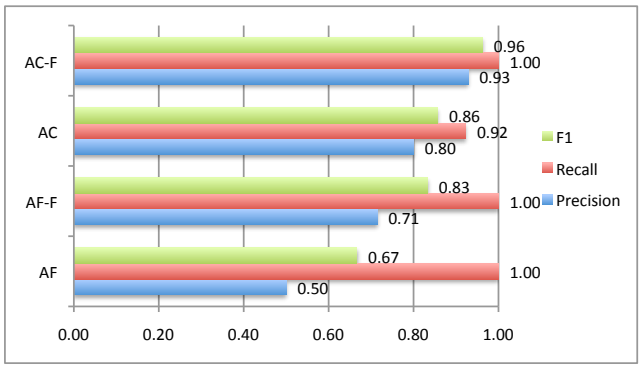

Figure 3. Considering unsuccessfully attempted activities boosts performance on standard activity recognition.

ways training on three games and testing against the fourth).

Fig. 2 compares both models in terms of precision, recall, and F1 score. We see that the baseline model has, in general, a respectable recall but it produces a large number of false positives for all activities besides failed freeing, which has the opposite problem of having low recall. The false positives stem from the fact that the algorithm is "greedy" in that it typically labels a situation where several players appear close to each other for certain period of time as a sequence of many captures and subsequent frees even though none of them actually occurred. The unified model gives significantly better results because it takes full advantage of the structure of the game in a probabilistic fashion.

Note that we did not put in any knowledge about the unsuccessfully attempted activities into the model. We only provided examples of game situations where those attempts occur and the system augmented itself and subsequently labeled all four activities. Thus, we see that we can indeed extend preexisting models in an automated fashion so that the unified model is capable of recognizing not only individual activities, but also both success and failure in people's behavior.

To address question Q2, we want to see how much does the recognition of attempted activities help in modeling the successful actions (the latter being the standard activity recognition problem). Therefore, we remove the module responsible for learning the failed attempts from our model and compare this stripped-down version, which considers only successful capturing and freeing, to the full-fledged unified model. Fig. 3 summarizes the results. We see that actual capturing with failure recognition model (AC-F) performs significantly better than the plain model of actual capturing (AC), and similarly for actual freeing with failure detection (AF-F) versus just actual freeing (AF).

CONCLUSIONS

Understanding activities that involve multiple interacting individuals is an important component of context-aware systems. This applies even in domains that are not primarily human-centered, because people are the "final judges" of the performance of any system. In this work, we make first steps toward explicit modeling of success and failure exhibited in complex activities. We show that given a model of successfully performed activities, along with a set of examples of failed attempts at the same activities, our system can automatically learn a generalized and augmented model that is capable of recognizing both the successful actions as well as mere attempts within a multi-agent domain. Additionally, we show that success, failure, and the intended goal of an activity are intimately tied together and having a model for successful events allows us to naturally learn models of the other two important aspects of life.

We evaluate our model and show that it achieves high recognition accuracy on real-world data. Furthermore, we demonstrate that explicitly modeling unsuccessful attempts boosts performance on other recognition tasks. Our system can function in both off-line and on-line (processing streaming data) modes and scales well both in terms of the number of data points considered and the extent of the geographic area. Additionally, our system makes it easy even for nonscientists to analyze the internal structure of the models.

FUTURE WORK

We are currently extending our model to handle not only explicit GPS traces, but also be able to *infer* the location of people who do not broadcast their GPS coordinates. The basic idea is, again, to leverage the structure of relationships among people. The vast majority of us participate in on-line social networks and typically some of our friends there do publish their location. We thus view the GPS-enabled people as noisy location sensors and use the network interactions and dynamics to estimate the location of the rest of the users. At present, we are testing this approach on public tweets.

REFERENCES

1. D. A. Baldwin and J. A. Baird. Discerning intentions in dynamic human action. *Trends in Cognitive Sciences*, 5(4):171 – 178, 2001.

2. N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4), 2006.

3. A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots: Learning a visually grounded storyline model from annotated videos. 2009.

4. L. Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, 2004.

5. A. S. Pentland. *Honest Signals: How They Shape Our World*. The MIT Press, 2008.

6. M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006.

7. A. Sadilek and H. Kautz. *Recognizing Multi-Agent Activities from GPS Data*. Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010.