Automatic Annotation of Team Actions in Observations of Embodied Agents

Linus J. Luotsinen, Hans Fernlund and Ladislau Bölöni
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, Florida
lluotsin@mail.ucf.edu, fernlund@mail.ucf.edu, lboloni@eecs.ucf.edu

ABSTRACT

Recognizing and annotating the occurrence of team actions in observations of embodied agents has applications in surveillance and in training of military or sport teams. We describe the team actions through a spatio-temporal correlated pattern of movement, which can be modeled by a Hidden Markov Model. The hand-crafting of these models is a difficult task of knowledge engineering, even in application domains where explicit, natural language descriptions of the team actions are available. The main contribution of this paper is an approach through which the library of HMM representations can be acquired from a small number of hand annotated, representative samples of the specific movement patterns. A series of experiments, performed on a dataset describing a real-world terrestrial warfare exercise validates our method and shows good recognition accuracy even in the presence of noisy data. The speed of the recognition engine is sufficiently fast to allow real time annotation of incoming observations.

Categories and Subject Descriptors: I.2.11 [Distributed Artificial Intelligence]: Multi-Agent Systems

General Terms: Algorithms, Performance, Experimentation

Keywords: Teamwork recognition, Multi-agent behavior modeling

1. INTRODUCTION

Humans have the ability to observe, recognize, analyze and occasionally adopt or adapt collaborative behavior. Successful team models, such as the Macedonian phalanx, Cromwell's new model army or the 4-2-4 formation of the 1958 world champion Brazilian soccer team represented a formidable advantage against unprepared opponents. Adversaries countered by studying these models, and either imitated or devised countermeasures against them. In our days, the organizational structures of terrorists groups pose significant challenges to homeland security; we are fighting an enemy which is organized along different patterns than we expected. Our ability to recognize teamwork is not restricted to humans: we can identify collaboration in wolf packs, flocks of birds or airplane formations.

In this paper we are developing methods which can recognize teamwork behavior in the observations of the movement of agents in the field, and annotate the series of observations with the recognized teamwork pattern. There are numerous applications of these approaches, for instance in surveillance or in training of sport or military teams. An annotated recording can help a team identify, in the process of After Action Review (AAR), their strengths and deficiencies. Through the observation of annotated recordings of other teams, one can learn teamwork models from successful teams, or develop countermeasures against adversaries.

Our approach is based on recognizing teamwork patterns based on a library of movements, encoded as Hidden Markov Models (HMM). Our work builds upon previous work by Sukthankar and Sycara [4], and extends it with the ability to dynamically acquire new teamwork patterns based on representative examples. We use a dataset representing a real-world terrestrial warfare exercise, and consider teamwork patterns commonly specified in military doctrines. The approach can be naturally extended to other domains where teamwork is expressed and can be recognized from the coordinated movement patterns of the embodied agents.

2. RELATED WORK

Many automated annotation systems were developed in the context of the RoboCup competition. Han and Veloso [2] employed HMMs to recognize behavior using recordings from RoboCup games. The approach can be used as a soccer commentary system. Rocco is another automatic commentary system for RoboCup that was developed by Voelz et. al. [5].

Recently Sukthankar et. al. developed algorithms for detecting teamwork in Military Operations in Urban Terrain (MOUT) data by considering the spatio-temporal evolution of the teams. In [4] HMMs are employed to create a spatio-temporal classifier that facilitates detection of teamwork activities. Each HMM depicts a specific teamwork behavior, and is created through knowledge engineering techniques. The behavior of the agents is represented by a set of three dimensional vectors (x, y and velocity).

3. ACQUIRING TEAMWORK PATTERNS AND ANNOTATING OBSERVATIONS

3.1 Pre-processing Observations

An observation sequence of length T is denoted by $\mathbf{V}^T$ and consists of observation vectors, or feature vectors, indexed by time,

978-81-904262-7-5 (RPS) ©2007 IFAAMAS

t, as: $\mathbf{V}^T = \{\mathbf{v}_1, \ldots \mathbf{v}_t \ldots, \mathbf{v}_T\}$, where $\mathbf{v}_t$ represents the visible observation $\mathbf{v}$ at time step t. The features selected for the observation vector, $\mathbf{v}_t$, are dependent on the application domain. In our domain of embodied agents, the sources of the observation data are the GPS readings, which we transform into x and y coordinates according to the Universal Transverse Mercator grid. Thus, given a team of four agents at time t we observe 8 features: $\mathbf{v}_t = \{x_{1,t}, y_{1,t}, x_{2,t}, y_{2,t}, x_{3,t}, y_{3,t}, x_{4,t}, y_{4,t}\}$, where x_1 and y_1 depict the x and y coordinate of the first team member. We normalize our observations to improve the ability of our recognition system to generalize over observations taken at different locations and orientations.

3.2 Modeling Team Actions with HMMs

In this study we employ HMMs to model the spatial and temporal relationships of a specific teamwork pattern. A HMM consists of n hidden states, where the hidden states are denoted by $\boldsymbol{\omega} = \{\omega_1, \ldots, \omega_n\}$. Let us denote with $P(\omega_j(t+1) \mid \omega_i(t))$ the transition probability, that is, the probability that the state of the HMM at time $t+1$ will be ω_j if at time t it was ω_i. The parameters determining the HMM are:

1. Transition probabilities,
   $\mathbf{A} = \{\alpha_{ij}\}$, where $\alpha_{ij} = P(\omega_j(t+1) \mid \omega_i(t))$
2. Emission probabilities,
   $\mathbf{B} = \{\beta_j(\mathbf{v})\}$, where $\beta_j(\mathbf{v}) = P(\mathbf{v} \mid \omega_j(t))$
3. Initial probabilities,
   $\boldsymbol{\pi} = \{\pi_i\}$, where $\pi_i = P(\omega_i(t))$

Thus, we can model a team behavior as $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$. One way of relating $\mathbf{A}$ and $\mathbf{B}$ to a team behavior is to think of it as a model of the team's mental and physical state respectively [2]. We model the emission probabilities using the general multi-variate normal density.

The model of a team behavior, $\boldsymbol{\theta}$, can be created through knowledge engineering or learning. Knowledge engineering is the manual creation of the model based on interviews with experts, natural language descriptions or direct observation of recordings. This is the approach taken in [4]. Unfortunately, the hidden nodes of the HMM do not necessarily correspond to states for which we have a good intuition, making the manual assigning of probabilities a difficult task. The approach taken by the work presented in this paper is to learn models from a set of representative examples of teamwork actions. This requires us to first manually annotate the occurrence of specific team actions in our observations[1]. We've employed two separate learning algorithms: Baum-Welch [1]; and Segmental K-Means [3].

To apply these algorithms we need to identify and extract representative examples from relatively large observation databases. Identifying and extracting representative examples from such large database can be a tedious and time consuming knowledge engineering task. For instance, one has to identify which team members contribute to the team behavior as well as where the representative example of interest starts and ends in time.

We've developed an interactive application for the editing and manipulation of recorded observations which simplifies the identification and extraction of representative examples. The workflow of the application is inspired from video editing applications, but instead of a single stream of video information, it allows the editing of the observation streams coming from a large number of agents. Using the application we can find and isolate the representative examples, which can be exported to the teamwork annotation framework for direct input to the learning algorithms that generate behavior models.

[1]This step is necessary even in the case of knowledge engineered models for verification purposes.

Figure 1 shows an overview of the full workflow of teamwork annotation. An external database of observations is imported into the observation editor where visualization and identification of teamwork activities result in a set of representative examples which will be used as training and validation data. The training and validation data is exported the teamwork annotation framework where teamwork behavior models are generated through the automatic learning process. Once the library of the team behaviors was built, we can use them to continuously check whether the currently observed behavior of the agents matches one of the patterns. This process, called behavior classification, will annotate the observations with the recognized team action. Given a behavior model $\boldsymbol{\theta}$ we can calculate $P(\mathbf{V}^T \mid \boldsymbol{\theta})$ recursively using the forward evaluation algorithm.

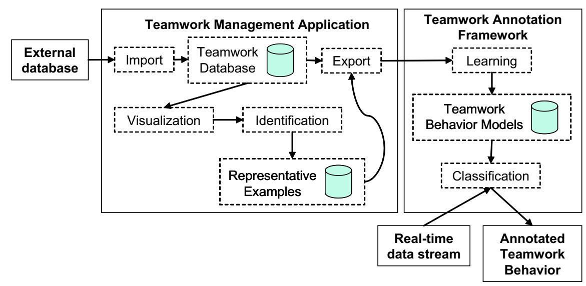

Figure 1. An overview of the teamwork annotation workflow.

4. EXPERIMENTAL RESULTS

4.1 Dataset Acquisition

A dataset to test the teamwork annotation framework was acquired from a real-world recording of a warfare exercise. To extract the representative examples necessary to train the behavior models, we had chosen data from eight tanks forming two competitive platoons. Using the teamwork editor application, we extracted representative examples for the following teamwork behaviors: 1) Column formation traveling; 2) Line formation traveling; 3) Box (combined line and column) traveling; 4) Team split; 5) Team merge; and 6) Bounding overwatch. The bounding overwatch movement was executed as follows. After all four tanks come to a stop, the team splits so that two tanks advance while the other two tanks keep their position. When advancing tanks stop, the previously halted tanks start their advance.

4.2 Annotation Framework Accuracy

The accuracy, represented by the fraction of successful classifications, of the annotation framework is calculated using the cross-validation procedure. Table 1 depicts accuracy (in a range from 0 to 1) as a function of the number of hidden states for both learning algorithms. The best overall result was obtained by the segmental K-Means algorithm using 4 hidden states, yielding a 0.8244 rate of successful classifications. The Baum-Welch algorithm peaks when 3 hidden states are used for each HMM, with a performance number of 0.7865.

Table 3. Error distribution (confusion) matrix showing misclassification rates for all behaviors.

	Bounding overwatch	Team split	Traveling line	Traveling box	Team merge	Traveling column	Unknown
Bounding overwatch	-	0.1	0	0	0	0	0.9
Team split	0	-	0	1.0	0	0	0
Traveling line	0	0.79	-	0.17	0	0	0.034
Traveling box	0	0.56	0.33	-	0.11	0	0
Team merge	0	1.0	0	0	-	0	0
Traveling column	0	0	0	0	0	-	0

Table 1. The ratio of correctly classified samples for HMM patterns with the number of hidden states ranging from 1 to 5, and trained in using the Baum-Welch and the Segmental K-Means learning algorithms.

Hidden states	Baum-Welch	K-Means
1	0.6787	0.6787
2	0.7294	0.7419
3	**0.7865**	0.7817
4	0.7588	**0.8244**
5	0.7188	0.766

Table 2. Classification accuracy over all teamwork behaviors. The table show correctly classified sequences (Matches), misclassified sequences (Error), correct classification rate and misclassification rate.

Behavior	Match	Error	Match rate	Error rate
Bounding overwatch	30	10	0.75	0.25
Team split	98	2	0.98	0.02
Traveling line	31	29	0.52	0.48
Traveling box	51	9	0.85	0.15
Team merge	51	9	0.85	0.15
Traveling column	80	0	1.0	0.0

Another statistics of interest is the classification accuracy distribution over all teamwork behaviors. The teamwork behavior library used was the one generated by the Segmental K-Means learning algorithm with 4 hidden states. In Table 2 we notice that the recognition accuracy is not distributed evenly across the behaviors. The behavior which was recognized most consistently was the traveling column, with a 1.0 recognition rate. Interestingly, the relatively simple behavior of traveling line fared the worst, being recognized only 31 times out of the 60 test sequences, yielding a match rate of 0.52.

We are interested not only in the number of behaviors which were classified incorrectly, but also which behaviors were the misclassified behaviors mistaken for. The confusion matrix in Table 3 contains this information. We find that errors in the classification of the traveling line behavior model occurred by the behavior being confused with the team split (79%) and traveling box (17%) behaviors. This is not a surprising result since the trace appearance of the traveling box behavior is similar to the traveling line behavior. Also, team split can occur in many ways. One particularly interesting observation sequence, that is included in our training dataset, is when the team splits into two team in the traveling line mode. The similarity of these training examples might have led to the high confusion rate in this case.

5. CONCLUSIONS AND FUTURE WORK

In this paper we described an approach for detecting and annotating teamwork behavior in observations of embodied agents. The main contribution is an approach through which the HMMs are learned from a small number of representative examples extracted from observation data. We tested our approach on a dataset representing a real-world military exercise. We find that the framework provides a recognition accuracy of approximately 82% on our dataset.

Despite these results, significant future work is necessary. The accuracy and performance numbers obtained refer to the somewhat idealized world of a military exercise, where embodied agents could be isolated from the other participating agents and embedded GPS devices allowed the acquisition of high quality data. In many practical applications, the annotation of teamwork needs to be performed in the conditions of higher environmental noise, less precise data and the presence of many additional agents. Our future work concerns improving our approach and develop new techniques to meet these challenges.

Acknowledgements: Research was sponsored in part by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-06-2-0041. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation heron.

References

[1] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Annals Mathematical Statistics*, 41(1):164–171, 1970.

[2] K. Han and M. Veloso. Automated robot behavior recognition applied to robotic soccer. In *Proceedings of International Joint Conference on Artificial Intelligence Workshop on Team Behaviors and Plan Recognition (IJCAI-99)*, 1999.

[3] B. H. Juang and L. R. Rabiner. The segmental k-means algorithm for estimating parameters of hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(9):1639–1641, 1990.

[4] G. Sukthankar and K. Sycara. Robust recognition of physical team behaviors using spatio-temporal models. In *Proceedings of Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2006.

[5] D. Voelz, E. André, G. Herzog, and T. Rist. Rocco: A RoboCup soccer commentator system. In *Proceedings of RoboCup-98: Robot Soccer World Cup II*, pages 50–60. Springer-Verlag, 1999.