Discovering Strategic Multi-Agent Behavior in a Robotic Soccer Domain

Andraz Bezek
Jozef Stefan Institute
Department of Intelligent Systems
andraz.bezek@ijs.si

ABSTRACT

This paper presents a method for multi-agent strategic modeling (MASM) applied in a robotic soccer domain. The method transforms multi-agent action sequences into a visual graph-based diagram, called action graph. Graph nodes are further augmented with additional domain knowledge. Using hierarchical clustering, action graph nodes are merged by utilizing domain-specific distance function. This step results in an abstract graphical model of agent behavior. Then, sub-graphs describing relevant agent behavior are used as input for association rule mining algorithm. The final output of MASM are strategic action concepts in the form of abstract action graph and association rules.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning - *Concept learning*

General Terms

Algorithms, Experimentation.

Keywords

Agents, multi-agent systems, agent behavior, strategy, RoboCup.

1. INTRODUCTION

Analysis of multi-agent systems must capture complex world state representation and asynchronous agent activities. Instead of studying pure numerical data researchers often use knowledge-level structures, such as rules and decision trees. Although being extremely useful when characterizing state space, they lack the ability to clearly represent temporal state changes. To capture such qualitative information, most often a graphical representation performs better in terms of human understanding.

This paper addresses the problem of qualitatively and quantitatively representing strategic concepts of multi-agent behavior. Our goal was to create a MASM algorithm that would enable humans to understand and study the strategic action concepts of the observed multi-agent systems. The presented approach translates annotated multi-agent action sequence into visual and rule-based representation of abstract agent behavior by using supplied domain knowledge.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

2. THE MASM ALGORITHM

An input to the MASM algorithm is a time-annotated multi-agent action sequence. The action sequence is then transformed into an action graph. An *action graph* is a directed graph, where nodes represent agent actions. Nodes a and b are connected, $a \rightarrow b$, if action b occurs directly after action a. The position of each node corresponds to the agent position when performing action represented by the node. The process of constructing an action graph from a multi-agent action sequence is presented in work from Bezek [2.].

We further augment action graph nodes with domain knowledge that adequately describes the represented agent action. This additional knowledge allows us to define distance function which measures distance between two nodes, i.e. two action descriptions.

The main difficulty with the approach is the following: what kind of knowledge is required to sufficiently represent agent action, and what is the corresponding distance function. We define action-related knowledge and the corresponding distance function in a way that facilitates us to achieve our main goals:

- **Find actions that occur near in the domain space.**

Distance in the domain space is a distance between domain space attributes. In general, distance between attributes, $dist_A$, is a heterogeneous distance function.

- **Generate abstract action descriptions.**

We introduce knowledge represented as *taxonomy*, which is a hierarchical representation of domain concepts. A concept a in taxonomy is an ancestor of a concept b, $a \leftarrow b$, if it exhibits more general concept than the concept b. Such representation allows us to travel up in a hierarchy to find more abstract domain concepts. Distance between two concepts in taxonomy is labeled as $dist_T$.

- **Find rules that describe abstract action descriptions.**

Action rules can be represented by a set of truth domain features. Each action is thus represented in a feature space as a binary *feature vector*, where i-th vector component corresponds to truth value of the i-th feature. The distance between two feature vectors is labeled as $dist_F$.

The node distance is as a weighted sum of each distance function:

$$dist(a,b) = w_A\, dist_A(a,b) + w_T\, dist_T(a,b) + w_F\, dist_F(a,b)$$

Distance function allows us to merge nearby action graph nodes using hierarchical clustering. We repeatedly merge the nearest nodes until the distance is grater than the specified *distance threshold*. The rationale behind the merge process is that similar actions, frequently occurring near in domain-space, reveal some important strategic concepts. Consequently, it is expected that the clustering process results in an *abstract action graph* that describes

agent behavior in a more abstract way than the original graph. The merged nodes thus represent similar action concepts. If we merge graph nodes with different hierarchical concepts then we appropriately update concept of the resulting node by choosing the nearest common ancestor for each concept pair. We also update node's position by averaging agent positions for all time instances, represented by the node.

Abstract action graph preserves the original action sequences and is thus a powerful behavioral representation. Another nice property is that abstract action graph has fewer nodes and connections and is thus easier to comprehend by humans. Node power is visually represented with node size and connection width is used as a visual measure of number of action instances between nodes.

The temporal information stored in nodes is then used to generate a dataset of truth features. These are taken as an input for the association rule mining algorithm [1.] that generates associative rules that describe abstract actions.

3. EVALUATION AND DISCUSSION

Our approach is applied on a RoboCup Simulated League (RoboCup) domain [3.], a multi-agent domain where two teams of 11 agents play simulated 2D soccer games.

For our purpose, we choose domain attributes to be agent's x and y coordinate. Therefore we define $dist_A$ as the Euclidean distance:

$$dist_A(a, b) = \sqrt{((a.x - b.x)^2 + (a.y - b.y)^2)}$$

We introduce hierarchical soccer knowledge through taxonomy of soccer roles (e.g. *field player → midfielder → center midfielder → left center midfielder*) and a taxonomy of soccer actions (e.g. *movement → offensive movement → movement on the ball → speed dribble*). We adopt the definition of hierarchical distance as presented by Giunchiglia et. al. [3.]. They define taxonomy distance as:

$$dist_T(a,b) = \begin{cases} 0; \text{ if } |shortest\_path(a,b)| <= threshold_T \\ \infty; \text{ else} \end{cases}$$

Features describe relations to players, ball, and both teams. We defined the following feature types: position-, speed-, ball-relative, and team-play- related (e.g. *left-wing, speed-slow, ball-near, team-play-attack*). Feature distance is a normalized Hamming distance:

$$dist_F(a,b) = |different\_features(a,b)| / |all\ features|$$

Rules in the form: *agent_role.agent_action ← true domain_features* are generated as follows. For each node, generate dataset of true domain features. Association rule mining will discover significant rules that describe the provided action concepts. We argue that the obtained rules and an abstract action graph represent strategic multi-agent behavior.

We applied our approach on a RoboCup game. Results are presented in Figure 1, where a) represents the original action graph, b) abstract action graph and c) the sub-graph describing actions that led to a successful goal**.** Table 1 presents some of the obtained rules that describe Figure 1 c).

Table 1: Rules describing a successful attack.

LTeam.Midfielder->Dribble	LTeam->Teamplay.Counterattack RTeam->Teamplay.Defensive ball->speed.Fast LTeam.Forward->Center-of-the-field
↓	
LTeam.Forward->Pass	ball->Penalty-box LTeam.Goalkeeper->Goal-box LTeam.Midfielder->speed.Fast LTeam.Fullback->speed.Stationary
↓	
LTeam.Forward->Shoot	ball->Danger-zone ball->Center-of-the-field LTeam.Forward->Center-of-the-field LTeam.Forward->Danger-zone

Although MASM was evaluated only on the RoboCup domain, we believe that there is a wide range of possible domains that can be exploited by the MASM since it's essence is stepwise abstraction in the domain-space.

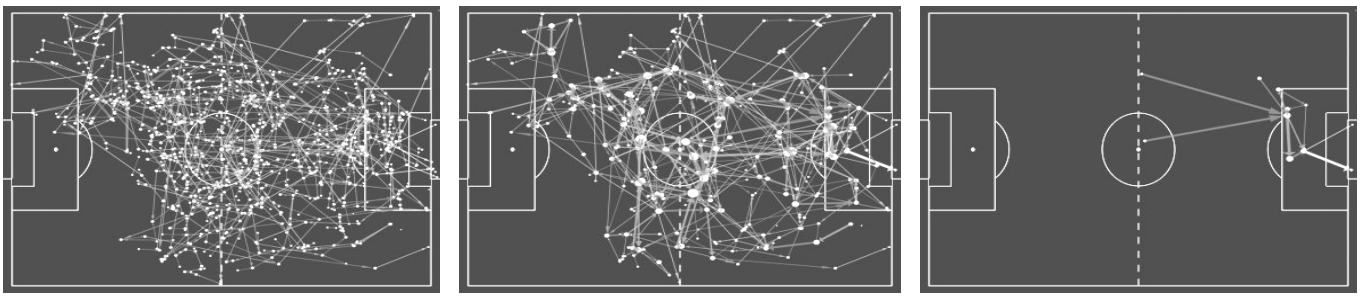
References

[1.] Rakesh Agrawal and Ramakrishnan Srikant: *Fast Algorithms for Mining Association Rules*. Proceedings of the VLDB 94, 1994.

[2.] Andraz Bezek: *Modeling Multiagent Games Using Action Graphs*. Proceedings of Modeling Other Agents from Observations (MOO 2004), 2004.

[3.] F. Giunchiglia and M..Yatskevich: *Element Level Semantic Matching*. Technical Report #DIT-04-035, 2004.

[4.] Itsuki Noda et al.: *Overview of RoboCup-97*. In Hiroaki Kitano, editor, RoboCup-97: Robot Soccer World Cup I, pp. 20-41. Springer Verlag, 1997.



a) Original action graph. b) Abstract action graph c) Relevant abstract action sub-graph

Figure 1: Various action graph representations of the same RoboCup game.