Unsupervised Role Discovery Using Temporal Observations of Agents

Andrew Silva
Georgia Institute of Technology
Atlanta, GA
andrew.silva@gatech.edu

Sonia Chernova
Georgia Institute of Technology
Atlanta, GA
chernova@gatech.edu

ABSTRACT

Agent-based modeling of multi-agent systems has enormous potential with applications in modeling social, economic, medical and other application domains containing temporal data. We propose an unsupervised approach to discovering common roles by observing agents over time, allowing us to construct a role-based representation of multi-agent systems that aids in understanding and interpreting the state of the system. We validate our approach on both a soccer and a StarCraft dataset, and show that unsupervised role discovery through observation can provide meaningful insight into the state of a multi-agent system, aiding or even replacing game state data for interpretation or understanding of the system.

KEYWORDS

Multi-agent systems; Unsupervised learning; Temporal learning; Interpretability

ACM Reference Format:
Andrew Silva and Sonia Chernova. 2019. Unsupervised Role Discovery Using Temporal Observations of Agents. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 8 pages.

1 INTRODUCTION

Agent-based models (ABM) represent phenomena as a multitude of different agents, each with their own goals, memory, tasks, and actions. By working independently or together, agents move toward a common goal, and these models often produce interesting emergent behaviors or insights that analytical and algorithmic models fail to reproduce [5]. Much of the research on ABM has been dedicated to accurately modeling agents themselves, and there has been less focus on studying multi-agent system (MAS) behaviors at a higher level. Analysis of MAS behavior can be challenging, as there may not be ground truth data about the underlying actions or motivations of individual agents.

Recently, role discovery has become a popular approach to understanding large quantities of unlabeled data. Role discovery provides insights into large amounts of data by autonomously identifying subsets of the data that are in some way related, often by similar graphical structures in social networks, such as in [20]. However, current approaches to role discovery are limited in that they require data to either be represented as a graph, or in some other way encode how agents are related to each other [11].

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

We introduce a new approach to understanding MAS through role discovery without the need for human input or supervision. By taking advantage of recent advances in unsupervised deep learning, we are able to use observations of individual agents over time in order to identify common subsets of the MAS data by finding similar behaviors over time, rather than by similar network connections or static agent histories. Our approach allows for interpretation of MAS strictly through observation, without the need to construct a graph to model relationships between agents, or the need to label agent actions, behaviors, or histories.

Our work makes three primary contributions. First, we introduce a novel way to interpret MAS by observing agents over time and identifying common roles, rather than by constructing graphs or labeling data. Second, we present multiple deep clustering approaches which employ autoencoders to reduce the dimensionality of temporal data for more interpretable and useful clusters. Finally, we validate these techniques using a soccer game dataset and a StarCraft replay dataset, and demonstrate how unsupervised role discovery can provide labels which improve human interpretability of the data and predictive model accuracy. Our results show that our role-based interpretation of MAS allows for greater semantic understanding of the system, and that our roles can meaningfully and usefully aid in the understanding of the state of a MAS.

2 RELATED WORK

Our work is related to several active areas of research, which we summarize briefly below.

2.1 Role Discovery

Role discovery has become a greater area of interest as social networks have gained popularity and there is a growing amount of unlabeled data from known individual agents. The goal of role discovery is to identify and assign roles to agents within some unlabeled dataset. Role discovery has been explored in relation to networks [11, 22, 29], Wikipedia editors [1, 28, 31], community structure [12, 15], email interactions [19], and more [9, 24, 32].

Traditionally, much of the work within role discovery approaches the problem by representing the data in a graph, as in [13, 22, 29]. Role discovery is then the challenge of finding nodes that have similar structural patterns, i.e. the same number of edges, or the same connectivity pattern, without paying any attention to the content of the node. Recently, however, researchers have begun to approach role discovery with feature-based representations, rather than graph-based representations of the data, as in [1, 19, 20, 31]. That is, research is beginning to examine the *semantics* of agent actions and behavior, and not simply the *syntactics* of how they are connected in a graph. Similarly, our work approaches role discovery as a feature-based problem, ignoring edges to neighbors or connections between agents. Our work is unique in that we discover roles using only brief temporal observations of agents, with no consideration for edges between agents.

2.2 Temporal Clustering

As deep networks continue to make great strides in supervised learning for images and videos, more attention is being turned towards unsupervised learning to attempt to leverage massive amounts of unlabeled data. In particular, Long Short-Term Memory (LSTM) networks have been used as autoencoders to condense high-dimensional temporal data into much smaller and more manageable vectors for clustering [2]. Clustering loss functions have even been developed for use by deep networks [6, 14, 30], where the input is an image and the output is a cluster label. We make use of an unsupervised deep network in our work, though our approach differs in that we seek to cluster temporal sequences, not images.

Our work is also related to activity recognition, which a recent survey [34] reviews various approaches for. Our work differs from activity recognition in that we seek to identify behaviors or roles, which are a higher level abstraction of activities. The same activity may be common in several different roles, where the only differentiation is the context in which the activity is found.

Finally, research within motif discovery seeks to discover similar sequences within temporal data. Motfis are "short approximately repeated patterns" [33] commonly used in medicine or genomics [25], and can be considered building blocks of activities. Unsupervised motif discovery [26] has made progress in finding multi-dimensional sequences with high similarity, which aided in unsupervised action discovery. As we intend to discover roles, our work seeks to find higher-level behavioral commonalities.

2.3 Multi-Agent Systems

Existing research in MAS has evaluated the impact of organizations of agents, and the ways that viewing MAS as organized subsets of agents rather than by attempting to model each individual agent [10] can improve overall performance of the system. Similarly, [7] have examined the ways that roles can improve interpretation of a MAS, though their work assumes roles are known in advance, and represents agents and roles as a graph for simplification of the MAS.

Research in sports analysis has taken a similar approach, labeling agents by their position in order to make simplifying assumptions about where they should be and how they should impact the game. Using this information, researchers are able to identify scoring opportunities [18], team formations [4], or even identify teams themselves based on the team compositions that they tend to use [3]. Recent work has even found that, using roles as labels for training different imitation learning models on defensive movements, it is possible to generate an accurate prediction of where defenders will move in response to incoming attacks [16]. Player positions can be identified by examining heat maps of player movement throughout a game, and establishing various mean (x,y) positions on the field [4]. We examine a soccer game as a MAS in our work, though our work is distinct in that we seek to identify a high level abstraction of player positions, namely roles that different players fulfill, rather than positions they are in. Additionally, we do so by using player behavior without considering their precise location on the field or how they move the ball. This removes to possibility of discovering roles using location-based heat maps, and makes the problem focus more on agent behavior and movement rather than position on the field.

Real-time strategy (RTS) games are popular within robotics and computer science, as they represent challenging MAS environments, often with massive state-action spaces, partial observability, adversarial agents, and non-deterministic, loosely-structured environments. RTS games have been used as a testbed for a variety of challenges, from plan recognition [23] to reinforcement learning [27]. Our work approaches RTS games as a MAS, where units represent individual agents that may conform to some latent hierarchy that can be used to aid in human understanding of the game, or in a model's interpretation of the game state.

3 DATASETS

In order to evaluate our approach and explore role discovery through observation, we make use of two datasets.

3.1 Soccer Dataset

The first dataset we use is a soccer dataset from the DEBS 2013 Soccer Grand Challenge[1]. This dataset includes location data for all players and the ball in a thirty-minute[2] game with eight players on each side. In order to avoid clustering by (x,y) position on the field, we construct and employ the following features:

- *Distance From Home:* distance a player is from their own team's goal
- *Distance From Enemy:* distance a player is from their opponent's goal
- *Distance From Ball:* distance a player is from the ball
- *Allies Nearby:* number of teammates within 10 meters
- *Enemies Nearby:* number of opponents within 10 meters

Distance features are normalized by the length of the field, and teammate or opponent features are normalized by the number of agents on each team. If information is ever missing, such as from a sensor malfunction, it is replaced with -1. We sample the soccer data at approximately 16fps and construct 16-sample windows, giving us 5700 1-second clips.

3.2 StarCraft Dataset

The second dataset we explore is a large dataset of StarCraft: Brood War replays [17]. As we are exploring the game as a MAS, we randomly sample 100 games with at least one Terran player for training, and we randomly sample 225 Terran vs. Terran games for evaluation. In both our training and evaluation datasets, we only consider mobile Terran units. The data is extracted using the Brood Wars API[3], which yields a wealth of information for every agent in each frame. We use the following features for every unit:

- *Distance From Home:* distance an agent is from their nearest base

[1] https://www.iis.fraunhofer.de/de/ff/lv/dataanalytics/tech/ek/download.html
[2] We only use about twenty five minutes of this data, as the ball sensor stopped functioning near the end of the first half
[3] https://github.com/bwapi/bwapi

- *Energy:* amount of energy an agent has (a resource for special abilities)
- *Visible:* binary flag for whether an agent is visible to enemies or not
- *Distance Moved:* distance an agent has moved since the last sampled frame
- *Nearby Allies:* number of allied agents within 10 units[4]
- *Nearby Enemies:* number of visible enemy agents within 10 units
- *Percent Health:* percent of health an agent has remaining
- *Current Cooldown:* percent of an agent's cooldown remaining on their weapon

Note that the Brood Wars API contains additional data that we leave out, either because it is extraneous (e.g. "burrowed", "irradiated", "blind", etc.), or because it makes the role assignment problem trivial (e.g. "attacking", unit type, "harvesting minerals", etc.). As in our soccer dataset, distances are normalized by the size of the map, and neighbors are normalized by the size of the army. If information is ever missing, such as if an agent dies during a window of observation, attributes are replaced with -1. We sample the StarCraft data at approximately 2fps and construct 16-sample windows, giving us 538,000 clips for training, and 1,125,000 clips for evaluation, where each clip is just over 8 seconds long.

4 APPROACH

As discussed in Sec. 2.2, there are several approaches to unsupervised learning for temporal data. To address the role discovery problem, we extend recent deep embedded clustering (DEC) work [30] and perform dimensionality reduction and clustering with one network, trained in two parts. DEC works by first pretraining a network as an autoencoder, and then exchanging the decoding layers for a cluster layer, which learns cluster centroids as weights. The clustering layer is initialized with k-means over embeddings from the encoding portion of the network, and then optimized over an unsupervised loss function. The unsupervised loss function is the KL divergence between the current cluster assignments and a target distribution P, where P is defined for every point in the dataset by squaring the probability of that point belonging to each cluster, and normalizing by the frequency of each cluster. In other words, for every point i in the data and for every cluster k

$$p_{i,k} = \frac{q_{i,k}^2/f_j}{\sum_{k'} q_{i,k'}^2/f_{k'}} \tag{1}$$

where $q_{i,k}$ is the probability of data point i belonging to cluster k, and f_k is the frequency for cluster k, normalized across all clusters $k' \in K$. Training in two parts allows the network to first learn a useful encoder for the data, and then improve the encoding while learning a function for clustering the embeddings.

Throughout our approach, we experimented with a variety of clustering methods for our clustering layer initialization and for running an abalation study. Methods we explored include: k-means, Gaussian-mixture models (GMMs), hidden Markov models (HMMs), spectral clustering, and agglomerative clustering. We found that spectral and agglomerative clustering consistently degraded into only two clusters, and GMMs performed on par with k-means. For simplicity, we only report results with k-means.

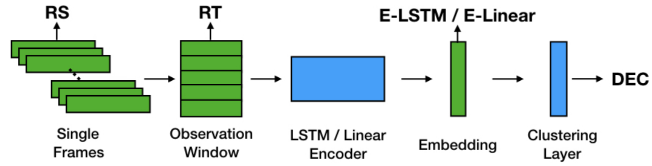

Figure 1: Visualization of our entire role discovery pipeline with our different baselines. RS: Raw Single frame data. RT: Raw Temporal data. E-LSTM/E-Linear: Embeddings + k-means. DEC: Full DEC network output.

An overview of the approaches explored in this paper is presented in Fig. 1, and discussed in detail below. Our approach begins with single frames of data, each of which represents a static observation of an agent. These single frames are then stacked into a matrix which represents a short temporal window of observation (1-8 seconds) for that particular agent. Our autoencoders then receive these windows as input data.

Prior work on DEC has utilized a convolutional autoencoder for clustering image data. In this work, we consider two alternate approaches in order to encode sequential observations of invidual agents. The first autoencoder we consider is an LSTM, which samples each timestep in series. The LSTM autoencoder uses a single LSTM for dimensionality reduction, downsampling the input into a lower dimension embedding of size Z. This embedding is then replicated to reproduce the window size of the input, before being passed through a separate decoding LSTM, which produces the output for comparison to the input.

The second autoencoder we consider is a linear autoencoder, which flattens all incoming timesteps into one long vector. This single vector is then downsampled by a linear layer into a lower dimension embedding of size Z, and finally upsampled back to the original dimensionality by a decoding linear layer. The decoded vector is then broken back into separate timesteps and compared with the input.

As in the original DEC work, we then exchange the decoding portions of our autoencoders for a cluster layer, and continue training. Once training is complete, the networks take in temporal observations of agents and output cluster labels. We refer to these models as LSTM DEC and Linear DEC, respectively.

We evaluate the contributions of each component of the DEC models by performing a thorough ablation study on the network. First, we compare to clustering on the raw single-frame data (RS) without considering a window of activity. Second, we compare to clustering on raw temporal data (RT), simply stacking frames over a window into a single vector. Finally, we compare to clustering on the Z-dimensional embeddings generated by the encoding portions of our DEC models (E-LSTM and E-Linear). By comparing each of these to the output from the DEC networks we train, we can validate the efficacy of temporal data over static data, and the efficacy of the embedding network and clustering layer independently.

The final challenge we face in our approach is selecting an appropriate number of clusters. Related role discovery work, such as [31], often involves searching through a range of possible clusters

[4]The average sight radius for StarCraft units we consider is 10

that a human has pre-specified, and manually inspecting the results of different numbers. As our roles represent patterns of behavior in different situations, it is difficult for us to determine the role that a cluster represents without manually watching many samples of that cluster. In order to approach this problem, we employ the Davies-Bouldin index [8], as it provides an unsupervised, compact, and fast metric for measuring overall cluster density and separation, and the results from the Davies-Bouldin index were comparable to results from slower or more memory intensive methods, such as the Silhouette Index [21]. We provide a brief overview of the metric below.

4.1 Davies-Bouldin Index

A cluster's dispersion, S, is measured as the average distance between a point in the cluster and the cluster centroid.

$$S_k = \frac{1}{N_k} \sum_{i=1}^{N_k} dist(x_i, C_k) \tag{2}$$

Where C_k refers to the centroid of cluster k, N_k is the size of cluster k, x_i is the i^{th} element of cluster k, and $dist(x_i, C_k)$ is any function that returns a non-negative symmetric distance between two points. Also, let $M_{k,j}$ be the distance between the centroids of clusters k and j. The Davies-Bouldin index (DBI) defines the similarity of two clusters as:

$$R_{k,j} = \frac{S_k + S_j}{M_{k,j}} \tag{3}$$

Which is to say, two clusters are more similar if their points are scattered and their centroids are nearby, and they are less similar if they are densely packed and their centroids are distant. Then, the DBI is simply the average of the maximum similarity between all pairwise clusters. In other words:

$$score = \frac{1}{K} \sum_{k=1}^{K} max(R_{k,j}) \forall j \in K \neq k \tag{4}$$

Where K is the set of all cluster centroids. For the DBI, lower scores are better, as they indicate greater distance and less similarity between clusters. Note that because the DBI is dependent on a distance metric, it is not comparable between different sets of embeddings, as we would have no way of normalizing for values that encoders generate in their embedding spaces. Therefore, we cannot say that one clustering model is superior to another strictly from their DBI scores.

4.2 Cluster Selection

Using the DBI scores for quantitative evaluation, we are able to search for values of K_m that will yield well-separated clusters for each model, m, independently. We begin by finding $K_{LSTMDEC}$ and $K_{LinearDEC}$ for our LSTM DEC and Linear DEC models, respectively, by searching for values between 2 and 10, fine-tuning the LSTM and Linear DEC models with each value, and computing DBI scores for each model. Once we have found the values of $K_{LSTMDEC}$ and $K_{LinearDEC}$ that yield the lowest DBI scores, we save the LSTM DEC model trained on $K_{LSTMDEC}$ clusters, and the Linear DEC model trained on $K_{LinearDEC}$ clusters.

We then use these trained models to generate the embeddings that will be passed to the E-LSTM and E-Linear models. We again try a range of different values for K_{E-LSTM} and $K_{E-Linear}$ for the E-LSTM and E-Linear methods, respectively, and ultimately save the k-means model that yields the lowest DBI score.

We also use the DBI to evaluate hyperparameter settings. If the lowest DBI score for a certain set of hyperparemeters is 2, it is likely that the model has not learned anything particularly interesting or insightful about our MAS data, as more than 2 roles are fulfilled throughout a soccer or StarCraft game. Similarly, if the lowest DBI score for set of hyperparameters is 10, it is likely that model has overfit to different actions in the games, and is no longer capturing interesting archetypal roles.

We restrict our search to few clusters, as we expect that there are few archetypal roles in our MAS data. Though our approach extends easily to any arbitrary number of clusters with little additional computational overhead, we note that higher numbers of clusters end up closer to motif discovery or action discovery rather than role discovery.

As noted above, the raw DBI is not comparable across different sets of embeddings or models. We can, however, normalize the scores for each model in order to concisely visualize the relative DBI scores from different models for various values of K. Fig. 2 shows normalized plots of DBI scores for top performing models on each dataset. The DBI scores are normalized with simple min-max normalization for each model so that all values are between 0-1. Given this information, we can see which models tend to underfit or overfit the data, and which values of K are most common.

5 EXPERIMENTS

We evaluate our role discovery approaches on the soccer and StarCraft datasets detailed in Sec. 3, using the number of clusters for each model resulting from the analysis above. We define the role of an agent as a function of its behavior through time, as opposed to the work of [31], where it is a function of topic modeling, or [20], where it is a function of graphical structure or relations. As such, our methods do not have an existing baseline for comparison. For all evaluations, we report results of our ablation study using the RS, RT, LSTM DEC[5], Linear DEC[6], E-LSTM, and E-Linear models. Most parameter values were achieved by performing a hyperparameter search for each model independently, though certain hyperparameters were shared by all models. We used a batch size of 64 for all experiments, and all models were trained with the Adam optimizer with a learning rate of 0.001. Our soccer encoders were pretrained for 64 epochs, while the StarCraft encoders were pretrained for 12 epochs, and all DEC models were fine-tuned until fewer than 0.1% of samples changed labels between updates.

5.1 Soccer Experiments

The soccer dataset that we use is considerably smaller than the StarCraft dataset, and as such we can easily visualize the entire game, as individual agents move throughout the field and change roles

[5] For both datasets, best performance was achieved with Z=8.

[6] The best performance was achieved with Z=16 for the soccer dataset, and Z=8 for the StarCraft dataset.

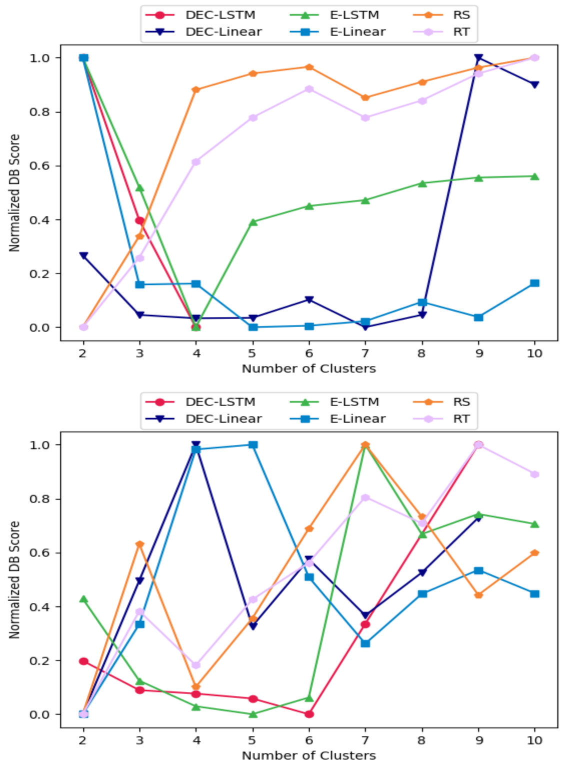

Figure 2: DBI scores for top performing models for soccer data (Top) and StarCraft data (Bottom). Scores are normalized between 0-1 for visual comparison.

depending on the context they are in. Visualization aids us in qualitatively making sense of the results from our different approaches, and in naming the roles for human readability, while evaluating which approaches have provided the most general abstraction of different roles on the field. We use the soccer dataset as a testbed for our approach to unsupervised role discovery, largely through qualitative observation of the results. Additionally, we introduce a quantitative metric of *continuity*, which is a measure of how stable or consistent role labels are. If a model produces roles with low continuity, it means that the agents are constantly flickering between different roles over time. This makes interpretation of the data more challenging, and likely indicates that the model has fit too closely to activities. We define continuity as:

$$changes = \sum_{i}^{I} \sum_{t=1}^{T} \mathbb{1}(K_{i,t}! = K_{i,t-1}) \tag{5}$$

$$continuity = 1 - \frac{changes}{|I| * |T|} \tag{6}$$

Where I is the set of all agents, and T is the set of all timesteps. After running our different quantitative cluster metrics, we seek to select a model that maximizes the continuity across all agents, without selecting the minimum number of clusters (i.e. 2 in soccer). As discussed above, the minimum number of clusters indicates underfit for our particular datasets. The LSTM autoencoder provides the best embeddings for discovering archetypal roles according to this metric. As we can see in Fig. 3 the Linear DEC and E-Linear models find roles with less continuity (56.83% and 69.29%, respectively), and upon visualization the Linear DEC model in particular appears to overfit to particular player positions or activities, as roles flicker in response to small changes in context. The RS and RT models, on the other hand, are too general, and simply identify which side of the half-way line players are on (continuity of 87.29% and 88.58%, respectively, but only 2 clusters each). These underfit roles do not provide us with anything useful or interesting to say about the *behavior* of the players, only about their location on the field. The LSTM DEC and E-LSTM models each have higher average continuity (71% and 77.08%, respectively) than the two Linear models, though they find comparable numbers of clusters.

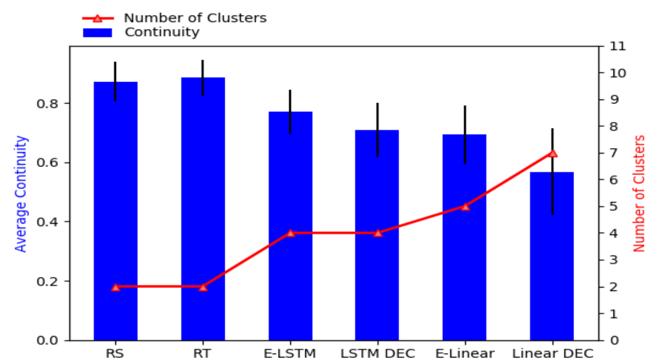

Figure 3: Average continuity across all agents and number of clusters discovered for each model

The difference between the success of our different approaches becomes much clearer upon visualization of the roles on the game data itself. The LSTM approaches discover the same four key roles, which could be best described as: “Attacker”, “Defender”, and “Defensive Midfielder”, and “Midfielder”. The Linear DEC and E-Linear approaches, on the other hand, discover seven and five different roles, respectively. The roles these methods discovered can be difficult to distinguish from each other, as a player will often flicker between different roles despite appearing to exhibit the same behavior. We attribute this to an overfitting of the models to different contexts or actions, and a failure to capture general player roles. The RS and RT methods each discover only two roles, which are easily described as “in my half” and “in my opponent’s half”. In situations where agents behave erratically, we observe that the linear methods create new clusters or roles for erratic or unseen behavior, while the LSTM methods will instead assign agents to the nearest existing role if a behavioral pattern is not common enough to warrant the creation of a new role.

It is interesting to note that there is some agreement between our approaches on key roles. There is substantial overlap between the roles that the LSTM and Linear approaches discover, for example, and the key difference between the models is in the granularity of the roles that are discovered. As we seek to discover behavioral roles that are independent of small changes in context, the LSTM models offer the best solution. For an example of the visualization

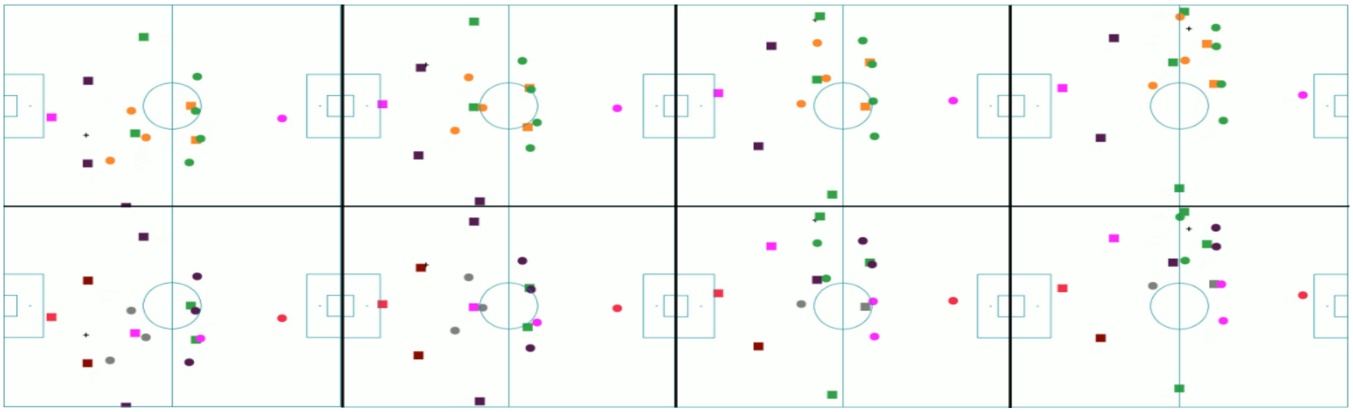

Figure 4: A sample of our soccer game visualization, where roles are symbolized by different colors, and teams are represented by different shapes. Roles from the LSTM (Top) are much more consistent than the Linear (Bottom) models, as is evident in this example by looking at the five left-most squares in each frame.

we use to see which roles are discovered in the soccer dataset, refer to Fig. 4.

5.2 StarCraft Experiments

Our StarCraft dataset is significantly larger than the soccer dataset we use, and therefore affords more quantitative evaluation. We use this data to evaluate how effective our roles are for predictive models to infer the state of the game. We compare the performance of three k-Nearest Neighbors classifiers trained on different datasets for two different tasks. The three datasets we compare are:

Raw data: Raw game state data for all units, as in Sec. 3.2

Raw data + roles: As above, with each unit's predicted role appended to the raw data

Role composition : A K-dimensional vector, where each element represents what percent of the army is currently fulfilling role k. For example, if an army contains 5 workers, 10 defenders, and 10 attackers, the vector would be [0.2, 0.4, 0.4]

We designed two tasks to measure the value of the roles discovered by our models. First, if roles capture meaningful information about the behavior and state of an agent, then we expect that they will be useful in predicting the state of the overall system, i.e. who will win the game. Thus, the first task is *success prediction*: given information on both teams, the model is tasked with predicting which team will win the game.

Second, we expect that meaningful roles will also be useful for predicting ablated attributes of an individual agent. Thus, the second task is *attribute prediction*: we ablate the "Distance From Home" feature from the datasets above, and task the model with predicting its value. We chose this feature because it is the most difficult to predict using only the game state, and it produces the highest average mean squared error across all of our models. Note that for this test, instead of role composition, which does not apply to single units, we use only a single feature: the unit's predicted role.

5.2.1 Success Prediction. For success prediction, we train and evaluate our models using 10-fold cross validation at 10% increments of time throughout the game (which allows us to normalize

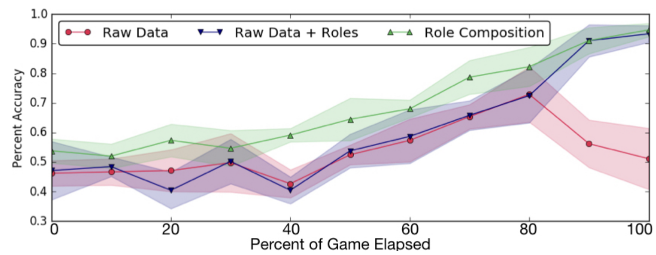

Figure 5: Accuracy for predicting the outcome of a game of StarCraft using KNN on 3 different datasets. Role compositions are significantly ($p < .001$) stronger predictors than raw game states. Role Composition is generated with the E-LSTM method, while the Raw Data and Raw Data + Roles are generated with the LSTM DEC method.

for varying game lengths). Note that for each data type, we report the results of the best performing model, as follows: LSTM DEC for *raw*, LSTM DEC for *raw data + roles*, and E-LSTM for *role composition*. Fig. 5 summarizes the results. We performed a repeated measures ANOVA across the three datasets at each interval and a Tukey's post-hoc test. Our results show significant difference in the predictive power of *role composition* over *raw* game states ($p < 0.001$), demonstrating the success of our approach in discovering semantically meaningful roles.

5.2.2 Attribute Prediction. Comparison on mean squared error for "Distance From Home" prediction reveals that the role data alone is not sufficient for perfectly predicting unit position. This result is to be expected, because the aim of our work is to capture high level roles that may take place at various locations in space (i.e. patrolling at different distances from the base). However, a one-way ANOVA with a Tukey's post-hoc shows that the LSTM models again significantly outperform all other models ($p < 0.001$) using only a single role feature. This result demonstrates that the LSTM models have identified roles that are fulfilled in different contexts throughout the game, as different roles are fulfilled in different contexts and positions around the map in StarCraft. A comparison across the top performing model in each category can be seen in Fig. 6

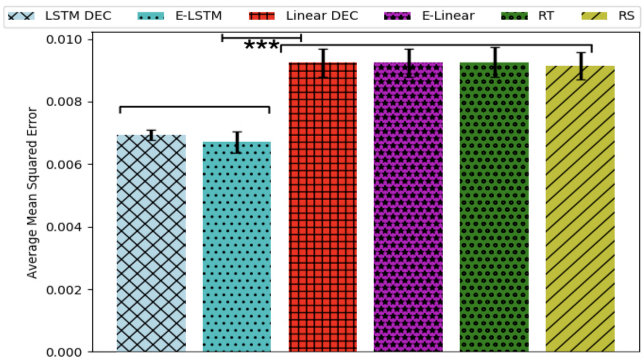

Figure 6: Predictive accuracy for each model using only roles to estimate a unit's distance from home. LSTM models significantly (p < 0.001) outperform all other models

When examining the *raw*, *raw data + roles*, and *role* datasets, we again see that roles provide semantically meaningful data for predictive models. A one-way ANOVA and Tukey's post-hoc reveals that the *raw data + roles* significantly outperforms all other datasets ($p < 0.001$) for predicting the ablated feature. This result demonstrates that, when combined with game state information, roles can be powerful signals to aid in understanding what an agent is doing at any given time. A comparison across datasets can be seen in Fig. 7.

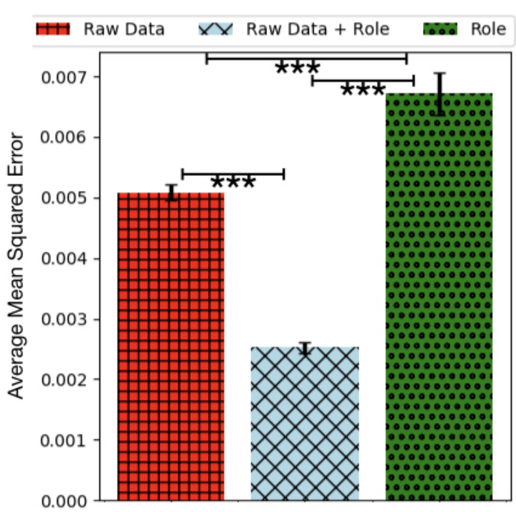

Figure 7: Predictive accuracy for each dataset estimating a unit's withheld attribute

6 DISCUSSION

Our results show that it is possible to identify behavior-based roles in different MAS from observations of unlabeled temporal data, and that these roles can be used to make sense of the state of the system. Our soccer experiments allow us to qualitatively confirm that our role discovery approach finds meaningful and interesting roles, while also confirming that deep LSTM-based methods are superior to raw k-means or deep linear-based methods for temporal analysis. Likewise, our StarCraft results validate that LSTM-based approaches provide the most useful roles for predicting game state information.

Interestingly, we observe that, while roles generally provide useful information in conjunction with raw data for predicting the outcome of a game of StarCraft, the role composition on its own is often a better predictor than the game state in its entirety (Fig. 5). We speculate that poor performance from the raw data is explained by the game state containing too much redundant or noisy information, preventing predictive models from clearly identifying a signal with only 200 games of data. Role composition alone, however, is able to concisely convey all of that information while removing much of the noise.

Finally, we find that in both datasets, classical techniques fail to find useful or interesting behavioral roles. Static k-means models cannot meaningfully find behaviors without seeing a full window of observation, and the temporal k-means models cannot handle all of the noise or interactions between static observations. In the case of the StarCraft data, both RS and RT identify whether or not units are alive or dead, and in soccer data they identify which half of the field players are in. Interestingly, we find that running k-means over DEC embeddings often yields results comparable to the DEC networks themselves. Comparable performance between E-LSTM / E-Linear and DEC indicates that, while the unsupervised loss function is essential to learning a separable embedding of the input data, the final clustering layer itself is not much better than simply performing k-means clustering on the embeddings.

7 CONCLUSION AND FUTURE WORK

We have presented a new definition for roles based on behavior and context rather than network connectivity or graphical structure, which can be easily applied to MAS to improve interpretability or semantic structure for predictive models. We have compared several different methods for identifying the higher level role information throughout temporal observations of soccer players and StarCraft units, and found that the LSTM DEC and E-LSTM methods are the most useful models for discovering roles that aid in interpretation of game state.

Future work includes taking advantage of this state abstraction for simplification of game state prediction in reinforcement learning, exploring other approaches for integrating behavior-based roles into MAS analysis and interpretation, and exploring role discovery on MAS without hand-engineered features.

8 ACKNOWLEDGMENTS

We would like to thank Matthew Gombolay for helpful discussions and suggestions. This work was supported in part by the Office of Naval Research grant N000141612835 and the Army Research Lab grant W911NF-17-2-0181.

REFERENCES

[1] Ofer Arazy, Hila Lifshitz-Assaf, Oded Nov, Johannes Daxenberger, Martina Balestra, and Coye Cheshire. 2017. On the" How" and" Why" of Emergent Role Behaviors in Wikipedia.. In *CSCW*. 2039–2051.

[2] Bharat Lal Bhatnagar, Suriya Singh, Chetan Arora, CV Jawahar, and KCIS CVIT. 2017. Unsupervised learning of deep feature representation for clustering egocentric actions. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 1447–1453.

[3] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, and Iain Matthews. 2014. Identifying team style in soccer using formations learned from spatiotemporal tracking data. In *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*. IEEE, 9–14.

[4] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, and Iain Matthews. 2014. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 725–730.

[5] Eric Bonabeau. 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences* 99, suppl 3 (2002), 7280–7287.

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep Clustering for Unsupervised Learning of Visual Features. *arXiv preprint arXiv:1807.05520* (2018).

[7] Rajib Kumar Chatterjee and Anirban Sarkar. 2014. Analysis of Multi Agent Systems: A formal approach. In *TENCON 2014-2014 IEEE Region 10 Conference*. IEEE, 1–6.

[8] David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* 2 (1979), 224–227.

[9] Anders Drachen, Rafet Sifa, Christian Bauckhage, and Christian Thurau. 2012. Guns, swords and data: Clustering of player behavior in computer games in the wild. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*. IEEE, 163–170.

[10] Jacques Ferber, Olivier Gutknecht, and Fabien Michel. 2003. From agents to organizations: an organizational view of multi-agent systems. In *International Workshop on Agent-Oriented Software Engineering*. Springer, 214–230.

[11] Danyel Fisher, Marc Smith, and Howard T Welser. 2006. You are who you talk to: Detecting roles in usenet newsgroups. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, Vol. 3. IEEE, 59b–59b.

[12] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174.

[13] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. Rolx: structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1231–1239.

[14] Steven Hickson, Anelia Angelova, Irfan Essa, and Rahul Sukthankar. 2018. Object category learning and retrieval with weak supervision. *arXiv preprint arXiv:1801.08985* (2018).

[15] Andrea Lancichinetti and Santo Fortunato. 2009. Community detection algorithms: a comparative analysis. *Physical review E* 80, 5 (2009), 056117.

[16] Hoang M Le, Peter Carr, Yisong Yue, and Patrick Lucey. 2017. Data-driven ghosting using deep imitation learning. (2017).

[17] Zeming Lin, Jonas Gehring, Vasil Khalidov, and Gabriel Synnaeve. 2017. STARDATA: A starcraft AI research dataset. *arXiv preprint arXiv:1708.02139* (2017).

[18] Patrick Lucey, Alina Bialkowski, Peter Carr, Yisong Yue, and Iain Matthews. 2014. How to get an open shot: Analyzing team movement in basketball using tracking data. In *Proceedings of the 8th annual MIT SLOAN sports analytics conference*.

[19] Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. 2007. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research* 30 (2007), 249–272.

[20] Ryan A Rossi and Nesreen K Ahmed. 2015. Role discovery in networks. *IEEE Transactions on Knowledge and Data Engineering* 27, 4 (2015), 1112–1131.

[21] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[22] Jerry Scripps, Pang-Ning Tan, and Abdol-Hossein Esfahanian. 2007. Node roles and community structure in networks. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. ACM, 26–35.

[23] Gabriel Synnaeve, Pierre Bessiere, et al. 2011. A Bayesian Model for Plan Recognition in RTS Games Applied to StarCraft.. In *AIIDE*.

[24] Christian Thurau and Christian Bauckhage. 2010. Analyzing the Evolution of Social Groups in World of Warcraft®. *CIG* 10 (2010), 170–177.

[25] Sahar Torkamani and Volker Lohweg. 2017. Survey on time series motif discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7, 2 (2017), e1199.

[26] Alireza Vahdatpour, Navid Amini, and Majid Sarrafzadeh. 2009. Toward Unsupervised Activity Discovery Using Multi-Dimensional Motif Detection in Time Series.. In *IJCAI*, Vol. 9. 1261–1266.

[27] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. 2017. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782* (2017).

[28] Howard T Welser, Dan Cosley, Gueorgi Kossinets, Austin Lin, Fedor Dokshin, Geri Gay, and Marc Smith. 2011. Finding social roles in Wikipedia. In *Proceedings of the 2011 iConference*. ACM, 122–129.

[29] Howard T Welser, Eric Gleave, Danyel Fisher, and Marc Smith. 2007. Visualizing the signatures of social roles in online discussion groups. *Journal of social structure* 8, 2 (2007), 1–32.

[30] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*. 478–487.

[31] Diyi Yang, Aaron Halfaker, Robert E Kraut, and Eduard H Hovy. 2016. Who Did What: Editor Role Identification in Wikipedia.. In *ICWSM*. 446–455.

[32] Diyi Yang, Miaomiao Wen, and Carolyn Rose. 2015. Weakly supervised role identification in teamwork interactions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 1671–1680.

[33] Yan Zhu, Abdullah Mueen, and Eamonn Keogh. 2018. Admissible Time Series Motif Discovery with Missing Data. *arXiv preprint arXiv:1802.05472* (2018).

[34] Maryam Ziaeefard and Robert Bergevin. 2015. Semantic human activity recognition: a literature review. *Pattern Recognition* 48, 8 (2015), 2329–2345.