Classifying Organizational Structures on Targets in the Cooperative Target Observation

Thayanne F. da Silva[1], Matheus S. Araújo[1], Raimundo J. F. Junior[1], Leonardo F. da Costa[1], João Pedro B. Andrade[1], Gustavo Augusto L. de Campos[1]

[1]Universidade Estadual do Ceará (UECE)
Caixa Postal 703 – 60.740.903 – Fortaleza – CE – Brazil

Abstract. *This paper proposes an approach to classify the organizational structure of a group of moving target agents that are continuously monitored by a smaller group of moving observer agents. The approach considers that the group of target agents can organize themselves according to eight different paradigms. The agents communicate through the exchange of messages whose contents are speech-act performative. We evaluate the approach considering seven techniques to solve the problem of classifying the group of target agents' organizational structure. The results show that the approach is promising, as it obtained a good performance, measured during experiments using agent-based simulations.*

1. Introduction

The Cooperative Target Observation (CTO) is a reformulation of the Cooperative Multi-Robot Observation of Multiple Moving Targets Problem (CMOMMT), the core of surveillance problems in environments where there are more targets than observers [Parker 1999], initially discussed in [Luke et al. 2005]. In these works, targets are random agents. However, in a globalized society, we must consider that the targets can be as strategic and organized in the same manner as their observers. For example, military on espionage missions, in which they use strategies based on organizational paradigms, to obtain information about the environment or their enemies avoiding being captured.

The observers in the CTO problem can collect data on the group of target agents' communication processes, whose content is performative of speech-act [Wooldridge 2009]. Thus, the group of observers in this paper aims to classify the organizational structure of the target agent group, possibly generated in the context of eight organizational paradigms [Abbas et al. 2015]: hierarchy, holarchy, team, coalition, congregation, society, federation, and matrix organization.

In experiments carried out using agent-based simulation, we considered that the target agents could be structured according to eight organizational paradigms. We evaluated seven supervised machine learning techniques applied to solve the classification problem in various CTO scenarios. We compare these results with the results obtained by an approach that classifies the structural organization based on the value of the frequency with which some organizational structures occur in the simulations. The results showed that machine learning classifiers are better than the frequency classifier in a considerably tricky scenario for the observer group: fast targets, low range of the sensor, and low range of the camera.

This paper is organized in more four sections. Section II presents the literature review. Section III describes the approach employed to solve the classification problem. Section IV shows the experiment and the results. Finally, Section V concludes the paper with the final remark and future researches.

2. Literature review

In the CTO problem, as in The Cooperative Multi-Robot Observation of Multiple Moving Targets Problem (also known by the acronym CMOMMT), the environment is a continuous 2D non-toroidal rectangular field, obstacle-free, containing N observer agents and M target agents, such that $N < M$. The speed of the target agents is slower than the speed of the observing agents. The observer agents try collectively to move and remain within a range of observation of as many target agents as possible. In most approaches, the task of cooperative observation of target agents, the next position of each observing agent, is computed by a coordinator agent that computes and sends the new destination position to each observer agent every γ time step, based on the k-means clustering algorithm. Each observing agent moves towards the calculated destination and continuously waits until a new destination point is received. The observer agents' objective is to maximize the average number of target agents observed (ANOT) during the observation time interval.

The first approach to the CTO problem compared the Hill-Climbing search and the K-means clustering strategy, to locate the targets such that observers can observe and maintain under their range of observation the maximum number of moving targets [Luke et al. 2005]. We recently improved the approach based on k-means. We introduced the notion of an organization in the team to model its functional, structural, especially the hierarchy structure, and behavioral dimensions, which must be present in a rational team of observers [Andrade et al. 2018].

In these and other approaches, the focus was on the rationality of the observer agents' movement. The targets move randomly by the environment or, consider targets only with basic movements, like the straight-line movement and one kind of controlled randomization movement [Aswani et al. 2017]. In a recent approach, we tried to improve the strategies of the target agents in two manners: we compared two organizational structure paradigms, the hierarchy, and the holarchy, incorporated in the group of target agents, with a strategy that incorporates an elementary neural network in each independent target to predict and avoid the observers' movements [da Silva et al. 2019].

To contribute to improving future research in the context of the CTO, in the more recent previous work, we seek to improve the performance of the target team by improving the rationality incorporated in the decision-making system of each target agent of the team. In this paper, we are returning to the context of the approaches that try to improve the observer agents performing, mainly motivated by the approaches described in [da Silva et al. 2019] and [Costa et al. 2019]. The following sections show the approach we developed to improve the performance of the observer team.

3. Model Description

3.1. Scenario

The CTO problem considers that the environment is fully observable and that targets cooperate with observers by reporting their locations. We modified the targets' objectives so that it only smears the environment when there were no observers around. Thus the targets seek to move away from the observers so that they can achieve their goal of dirtying the environment. In this paper, the targets were separated into two groups with the same number of members. However, one group walks randomly on the scene while the other seeks to dirty the environment when there are no nearby observers.

We also modified the scenario. There are discreet cameras that see the targets that are soiling the environment and transmit this information to the observers must decide where to make their next moves.

In this new CTO scenario, containing N agents observers and M agents target, so that $N < M$, we modify the observers' decision-making system. Initially, the observer group in the environment follows the same strategy proposed by [Luke et al. 2005], i.e., clustering the M agents into the target agent group in N subgroups so that each of the N observers decides to move towards the centroid of their nearest target group. When a malicious target is detected, one observer moves towards it, while the others decide to move towards their clusters. When the observer group identifies that the number of malicious targets K in the environment is greater than or equal to N, i.e., $N \leq K \leq M$, the clustering algorithm runs to find the N clusters containing only the malicious targets.

3.2. Modeling of Organizational Paradigms

The main organizational structure paradigms used by the targets in our approach are described in [Abbas et al. 2015]: hierarchy, holarchy, coalition, team, congregation, society, federation, and matrix organization. These structures are informally described bellow.

3.2.1. Hierarchy

The hierarchy has two levels. The lowest level is the targets, who have a local view based on the range of their sensor, and at the highest level, its general coordinator, who has a global view of the scenario, Algorithm 1.

3.2.2. Holarchy

In this approach, the targets were divided into two hierarchical groups, with the same number of members fixed, with a coordinator subordinate to a general coordinator, Algorithm 2. The coordinator of one group has no relationship with the members of another group.

3.2.3. Team

In this approach, Algorithm 3, targets work cooperatively to dirty the scenery. When a target has one or more observers within the range of the sensor, it communicates to the other targets next to them to avoid it with the straight-line movement in the opposite direction.

Algorithm 1: Hierarchy

```
while $t < \Delta t$ do
    if general coordinator then
        if $t\ mod\ \gamma = 0$ then
            receives a message from targets about their positions;
            execute K-means;
            sends a message to targets requiring action;
        end
    else
        if $t\ mod\ \gamma = 0$ then
            sends a message to the general coordinator informing its position;
            receives a message from general coordinator about the required action;
            straight-line movement of the nearest centroid;
        else
            walk randomly;
        end
        if count observers in range-sensor = 0 then
            mess the environment;
        else
            walk randomly;
        end
    end
end
```

Algorithm 2: Holarchy

```
while $t < \Delta t$ do
    if coordinator then
        if $t\ mod\ \gamma = 0$ then
            if general coordinator then
                receives a message from targets about their positions;
                sends a message propagating the information to Kmeans and Cmeans coordinators;
            else
                if Kmeans coordinator then
                    receives a message from general coordinator;
                    execute K-means;
                    send a message to Kmeans targets requiring action;
                else
                    receives a message from general coordinator;
                    execute Fuzzy C-means;
                    send a message to Cmeans targets requiring action;
                end
            end
        end
    else
        if $t\ mod\ \gamma = 0$ then
            if Kmeans targets then
                sends a message to the general coordinator informing its position;
                receives a message from its coordinator about the required action;
                straight-line movement of the nearest observer;
            else
                sends a message to the general coordinator informing its position;
                receives a message from its coordinator about the required action;
                straight-line movement of the nearest observer;
            end
        else
            walk randomly;
        end
        if count observers in range-sensor = 0 then
            mess the environment;
        end
    end
end
```

Algorithm 3: Team

```
while $t < \Delta t$ do
    if targets in range-sensor then
        sends a message to nearest targets informing its status;
        receives a message from nearest targets about their status;
        if status = watched then
            straight-line movement of the watched target;
        end
    else
        walk randomly;
    end
    if count observers in range-sensor = 0 then
        mess the environment;
    end
end
```

3.2.4. Coalition

In the coalition, there are two targets: those on the right and those on the left, Algorithm 4. This division is not previously decided but occurs according to the real situations the agents are in the environment. The targets only interact with the members of their group, informing their situation.

Algorithm 4: Coalition

```
while $t < \Delta t$ do
    if target position between $-1$ and $75$ then
        receive message from group 1;
    else
        receive message from group 2;
    end
    if group1 target then
        if group1 target in range-sensor then
            sends a message to nearest group1 targets informing its status;
            receives a message from nearest group1 targets about their status;
            if status = watched then
                straight-line movement of the watched group1 target;
            else
                walk randomly;
            end
        else
            walk randomly;
        end
    else
        if group2 target in range-sensor then
            sends a message to nearest group2 targets informing its status;
            receives a message from nearest group2 targets about their status;
            if status = watched then
                straight-line movement of the watched group2 target;
            else
                walk randomly;
            end
        else
            walk randomly;
        end
    end
    if count observers in range-sensor = 0 then
        mess the environment;
    end
end
```

3.2.5. Congregation

The congregation is similar to the coalition approach. There are also two groups of target agents but the same number of members. The groups also are previously defined, and they do not change according to the situations the agents are in the environment, Algorithm 5.

Each target communicates only with the agents of its group about its situation in the environment.

Algorithm 5: Congregation

```
while $t < \Delta t$ do
    if group1 target then
        if group1 target in range-sensor then
            sends a message to nearest group1 targets informing its status;
            receives a message from nearest group1 targets about their status;
            if status = watched then
                straight-line movement of the watched group1 target;
            else
                walk randomly;
            end
        else
            walk randomly;
        end
    else
        if group2 target in range-sensor then
            sends a message to nearest group2 targets informing its status;
            receives a message from nearest group2 targets about their status;
            if status = watched then
                straight-line movement of the watched group2 target;
            else
                walk randomly;
            end
        else
            walk randomly;
        end
    end
    if count observers in range-sensor = 0 then
        mess the environment;
    end
end
```

3.2.6. Society

In society, there are two types of malicious target agents with their respective tasks, targets that dirty the environment and the targets that look for the cameras, Algorithm 6.

Algorithm 6: Society

```
while $t < \Delta t$ do
    if lookouts then
        if found camera then
            stop and sends a message to targets requiring action;
            sends a message to lookout informing about the camera;
        else
            search for cameras;
        end
    else
        if lookouts in range-sensor then
            receives a message from lookouts about the required action;
            if action is to move away then
                straight-line movement;
            else
                mess the environment;
            end
        else
            if targets in range-sensor then
                sends a message to nearest targets informing its status;
                receives a message from nearest targets about their status;
                if status = watched then
                    straight movement of the watched target;
                else
                    walk randomly;
                end
            end
            if count observers in range-sensor = 0 then
                mess the environment;
            else
                walk randomly;
            end
        end
    end
end
```

3.2.7. Federation

In this approach, Algorithm 7, there are two types of target agents: targets that dirty the environment and intermediate targets. There are also two groups of malicious agents with the same number of members. One of whom will be the intermediary. The group targets cede their autonomy to the intermediary to achieve the goal of their group. The intermediary explores the environment, without getting dirty, communicating with other intermediaries to obtain information to pass the information on to his subordinates, where they may or may not dirty the environment.

Algorithm 7: Federation

```
while $t < \Delta t$ do
    if intermediary then
        if observers in range-sensor then
            if their subordinates in range-sensor then
                sends a message to your subordinates requiring action of not soiling;
            end
        else
            search for observers;
            sends a message to your subordinates requiring action of soiling;
        end
        if other intermediary in range-sensor then
            sends a message to other intermediary informing about the state of the environment;
            if state = not allowed then
                sends a message to your subordinates requiring action of not soiling;
            else
                sends a message to your subordinates requiring action of soiling;
            end
        end
    else
        if intermediary in range-sensor then
            receives a message from intermediary about the required action;
            if permission = allowed then
                mess the environment;
            else
                walk randomly;
            end
        else
            walk randomly;
        end
    end
end
```

3.2.8. Matrix Organizations

In this approach, we have a coordinator (high level), intermediaries (middle level), and the targets (lowest level), Algorithm 8. If the coordinator does not send any information in the matrix organizations, it communicates with an intermediary within its communication range to obtain information about the environment. If there is also no intermediary nearby, it communicates with other targets.

3.3. Communication

The organizational approach applied to the group of targets that dirty the environment considers that an agent playing the observer role must be able to monitor the exchange of information, through the proximity between the observer and the targets. If the observer is in range of the communication range, it can collect the communication from the targets.

The target agents, to achieve objectives, can employ a language of communication in which the speech-act performative are visualized as actions (in the same way as the actions performed by the agents' actuators), whose effects occur mainly in the models that the speakers (S) and hearers (H) keep of each other. Regarding the communication

Algorithm 8: Matrix Organizations

```
while $t < \Delta t$ do
  if coordinator then
    if $t \; mod \; \gamma = 0$ then
      receives message from targets about their positions;
      execute K-means;
      sends a message to targets requiring action;
    end
  else
    if intermediary then
      if observers in range-sensor then
        if their subordinates in range-sensor then
          sends a message to targets requiring action of not soiling;
        end
      else
        search for observers;
        sends a message to targets requiring action of soiling;
      end
      if other intermediary in range-sensor then
        sends a message to other intermediary informing about the state of the environment;
        if state = not allowed then
          sends a message to targets requiring action of not soiling;
        else
          sends a message to targets requiring action of soiling;
        end
      end
    else
      if $t \; mod \; \gamma = 0$ then
        sends a message to the coordinator informing its position;
        receives a message from coordinator about the required action;
        straight-line movement of the nearest observer;
      else
        if intermediary in range-sensor then
          receives a message from intermediary about the required action;
          if permission = not allowed then
            straight-line movement of the intermediary;
          end
        else
          if targets in range-sensor then
            sends a message to nearest targets informing its status;
            receives a message from nearest targets about their status;
            if status = watched then
              straight-line movement of the watched target;
            else
              walk randomly;
            end
          else
            walk randomly;
          end
        end
      end
      if count observers in range-sensor = 0 then
        mess the environment;
      end
    end
  end
end
```

language, the approach considers a subset of the speech-act language generated from two key speech-act messages: request and inform. Figure 1 illustrates the proposed format for these messages and two examples of compositions of these messages aiming at generating speech-acts of questions (requests to inform) and requests involving at least three agents (requests to request).

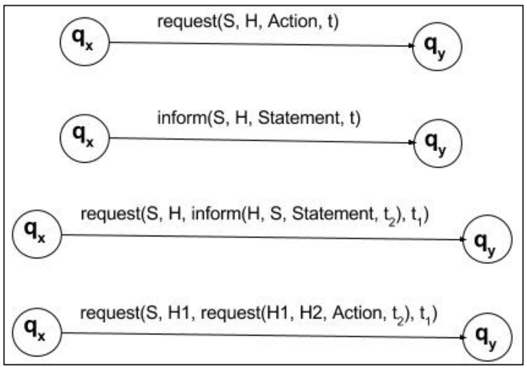
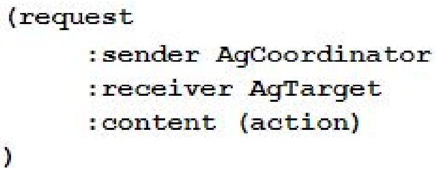

q_x
request(S, H, Action, t)
q_y
q_x
inform(S, H, Statement, t)
q_y
q_x
request(S, H, inform(H, S, Statement, t_2), t_1)
q_y
q_x
request(S, H1, request(H1, H2, Action, t_2), t_1)
q_y

Figure 1. Exchange of messages between target agents in an organization.

The formalism for describing the interaction protocol between the target agents can be synthesized from the works on finite state machines. In the Fig. 1, states represented different stages of conversation and directed arcs connecting states are labeled with messages wrote in a communication language. The messages transmitted contains representations of information about the intention of the message, the speaking agents (S) and the hearer agent (H), as well as the content of the message to be exchanged between the agents (Action or Statement) and the instant (t) of the conversation. Any talking agent S who sends a request type message aims at having some listening agent H perform an Action action. Any talking agent S who sends an informer-type message aims to make the listening agent H believe any statement (Statement) as in the case of the formulation of messages involving questions, all the semantics of speech acts that make up the communication language FIPA-ACL (Foundation for Intelligent Physical Agents - Agent Communication Language) can be defined in terms of information and request. Figure 2 shows a message on the target team, in which an observer captured a message from a coordinating agent to one of the target agents requesting an action.



```
(request
     :sender AgCoordinator
     :receiver AgTarget
     :content (action)
)
```

Figure 2. An example of a message sent from the coordinator to a target

The attributes used in the classification and collected by the observers when they were under the reach of the communication are the performative, target agent who speaks (S), the target agent who hears the message (H), and the content in the message (C = Action or Statement).

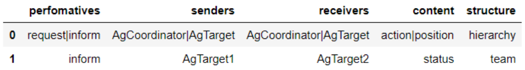
3.4. Classifiers

Our approach selected seven classifiers, which are: K-Nearest Neighbors(KNN), Support Vector Machines(SVM), Naive Bayes(NB), Decision Tree(DT), Random Forest(RF), Linear discriminant analysis(LDA) and Dummy.

This last classifier will be used as a basis of comparison. A dummy classifier is a type of classifier that does not generate any insight into the data and classifies the given data using only simple rules. The rule selected was the most frequent that an organizational structure appeared. Because suppose it is desired to determine whether a particular object has a specific property or not. If a large number of these objects have been analyzed and 90 percent has been found to contain the property, suppose that all future instances of the object have the property. There is a 90 percent probability of guessing correctly.

The values defined for the parameters of the models were in accordance with the standard values of the scikit-learn library.

The figure 3 shows an example of the attributes table of the messages exchanged during the simulation by the target agents that the observing agents captured. To simplify the table, there is no repetition of values in the attributes, that is, if a value has already been added in an attribute of the table, it will not be added again if the observer captures again during the simulation.



<table>
<tr><th></th><th>perfomatives</th><th>senders</th><th>receivers</th><th>content</th><th>structure</th></tr>
<tr><td>0</td><td>request|inform</td><td>AgCoordinator|AgTarget</td><td>AgCoordinator|AgTarget</td><td>action|position</td><td>hierarchy</td></tr>
<tr><td>1</td><td>inform</td><td>AgTarget1</td><td>AgTarget2</td><td>status</td><td>team</td></tr>
</table>

Figure 3. Attribute table example.

4. Experiments and Results

On the NetLogo platform, [Kravari and Bassiliades 2015], was designed the model. The NetLogo platform allows running parallel experiments, is easy integration with the R software [Maronna et al. 2019], and can be run on Jupyter Notebook [Randles et al. 2017]. The Jupyter Notebook ran the experiment. The parameters were based on the settings used by [Luke et al. 2005]:

- The targets and observers are in a rectangular field with dimensions $150 \ per \ 150$ units;
- 1500 interactions steps per simulation;
- The speed of the observer is 1 step by time interval;
- Target speed is 0.9 step by time interval;
- Ranger-sensor is 25 units;
- Ranger-camera is 5 units;
- Ranger-communication is 5 units;
- γ is 10;
- 12 random targets and 12 malicious targets;
- 12 observers;
- 4 cameras.

The values of the parameters of ranger-camera, ranger-communication, and target speed sought to hinder the task of observers and assess whether in an uncertain environment our approach using the seven classifiers in the communication between targets would be better than trying to classify the organizational paradigm through the how often they appeared on the stage. Thus, the lowest reach value defined by [Luke et al. 2005], i.e., 5, and the highest speed for targets, 0.9. The range-sensor value was the highest range value defined by [Luke et al. 2005], as the targets and observers have the same range; the value of this parameter will benefit both teams.

Each structure is simulated by 100 times, capturing performative, speaker (S), hearer (H), message content, and organizational structure that converts to a DataFrame. From these data, cross-validation was performed with *StratifiedKFold* with $n\_splits = 5$, standard parameter, and the *shuffle* as true.

Table 1 shows the result of the performance of classifiers.

Table 1. Cross-Validation results

Classifier	Average Accuracy	Standard Deviation
Dummy	0.11	0
KNN	0.384	0.025
SVM	0.418	0.025
LDA	0.424	0.037
Random Forest	0.424	0.036
Naive Bayes	0.424	0.030
Decision Tree	0.424	0.016

The seven classifiers were better at forecasting than the Dummy classifier. Four of them had the highest average accuracy, with the DT showing the lowest standard deviation, which means that the accuracy obtained in the cross-validation is more homogeneous than in the NB, RF, and LDA. KNN classifier obtained the lowest average accuracy value, but it was still more efficient than the Dummy. Emphasizing that the environment was modeled to hinder the task of observers, but with our approach, observers were able to classify organizational paradigms more correctly, thus, observers lose data on target communication to carry out classifications. In scenarios more favorable to observers, the hit percentage increases considerably, because observers will get more data from the communication to assist in classification.

5. Conclusion and Future Works

In this paper, we seek to classify the organizational structure of the target team through seven supervised learning classifiers using data from communications between the target agents obtained by the observing agents. This approach proved to be effective compared to the *Dummy* classifier that simulates the human logic to predict based on the frequency that the structure appeared in training simulations. Thus, if in a difficult scenario for the observer, this agent obtained exciting results, in a scenario more favorable to observers, the result should be better. Perhaps the performance of classifiers will change in an environment where they obtain more information for predictions.

The parameters used seeking to hinder the classifications of the observer team. However, there are means of communications and cameras with a higher range than human eyes (range sensor of the observer), so it is necessary to analyze the best parameters that simulate a real situation. As future work, we intend to carry out an experiment that aims to obtain the best values for the parameters and generate a dataset so that research in this area on security and organizations' types can be developed. As observers can more efficiently predict the organizational structures used by targets, this also leaves room to generate new strategies for the observer team to ensure safety in the environment.

References

Abbas, H. A., Shaheen, S. I., and Amin, M. H. (2015). Organization of multi-agent systems: an overview. *arXiv preprint arXiv:1506.09032*.

Andrade, J. P. et al. (2018). Organization/fuzzy approach to the cto problem. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 444–449. IEEE.

Aswani, R., Munnangi, S. K., and Paruchuri, P. (2017). Improving surveillance using cooperative target observation. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Costa, L. et al. (2019). Comparative study of neural networks techniques in the context of cooperative observations. In *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, pages 563–574. SBC.

da Silva, F. et al. (2019). Smart targets to avoid observation in cto problem. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1958–1960.

Kravari, K. and Bassiliades, N. (2015). A survey of agent platforms. *Journal of Artificial Societies and Social Simulation*, 18(1):11.

Luke, S., Sullivan, K., Panait, L., and Balan, G. (2005). Tunably decentralized algorithms for cooperative target observation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 911–917.

Maronna, R. A., Martin, R. D., Yohai, V. J., et al. (2019). *Robust statistics: theory and methods (with R)*. John Wiley & Sons.

Parker, L. E. (1999). Cooperative robotics for multi-target observation. *Intelligent Automation & Soft Computing*, 5(1):5–19.

Randles, B. M., Pasquetto, I. V., Golshan, M. S., and Borgman, C. L. (2017). Using the jupyter notebook as a tool for open science: An empirical study. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 1–2. IEEE.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.