2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI) | 978-1-6654-7406-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/CogMI56440.2022.00020

Collaboration, Self-Reflection, and Adaptation in Robot Communities: Using Multi-Agent Distributed Learning for Coordination Planning

Javed Mostafa
iSchool & School of Medicine
University of North Carolina
Chapel Hill, U.S.A
jm@unc.edu

Weimao Ke
College of Computing & Informatics
Drexel University
Philadelphia, U.S.A.
wk@drexel.edu

***Abstract*—Robotic communities are increasingly important in executing operations in a wide variety of industries. Before designing and deploying such robots it is important to determine and carefully plan the configuration, knowledge composition, and coordination strategies. Multi-agent simulation modeling offers a malleable and powerful way to conduct such planning and elucidate key parameters and their interactions associated with collaboration dynamics. The paper offers motivations, an adaptive learning scheme, and empirical evidence drawn from a few case studies. Among the key findings one is that complex tasks can be conducted effectively and efficiently over billions of robots without relying on a singular source of global knowledge. Another interesting finding is that through collaboration and emergent learning, robots can create communication channels among dominant players and less dominant intermediaries that are critical connectors across network overlays (representing clusters of specialists).**

***Index Terms*—Multiagent systems, robotics, distributed AI, learning, text classification, information search and retrieval**

I. Introduction

Agents, containing representation of a logical unit of knowledge and with the ability to conduct a predefined set of tasks, can offer a powerful way to model human-machine collaborations. Such models offer immense flexibility in understanding dynamics associated with achieving complex goals through knowledge sharing, communication, delegation, assessment, and adaptation over time. As AI continues to develop and trust grows in delegating more complex and large-scale operations to robots it becomes ever more challenging to determine the correct configurations, knowledge compositions, and coordination strategies needed to achieve performance at optimal levels. There could be several key configuration parameters, but the most important one is the total size of the robot population required to conduct automated task operations. The composition category dictates the amount of memory, knowledge structures, procedural routines and operational capacity (i.e., speed of processors). The coordination dimension has to do with the planning and communication algorithm deployed and used for facilitating knowledge sharing, updates, and collaboration among robots and humans[1]. In this brief paper, the focus will be primarily on configuration and composition parameters and using a multi-agent simulation platform to execute different coordination strategies to clarify and understand collaboration dynamics before robots are actually deployed and operations are run.

Distributed AI modeling, as those offered through multi-agent simulations whereby the units of operations as encapsulated in robots or packaged modules of automated operations, can offer a powerful way to to understand the coordination processes at both macro and micro levels of granularity and ultimately develop a robust plan. A macro level challenge for robot coordination include determining how the complexity of tasks and the type of data shared among a robot community influence collaboration patterns. Another major macro level challenge is to establish the learning scheme, which drives the community of agents toward common goal/s and perform the overall operation. The learning scheme facilitates ongoing and accurate information sharing among agents and it must allow human operators / collaborators to provide feedback to ensure steady progress and convergence toward expected goals. At the micro level there are also a few critical dimensions that need attention. It may be worthwhile to be able to track emergence of sub-clusters, cliques, or local overlays that emerge over time and specialize in conducting certain sub-tasks in a more optimal way. At an even finer unit of micro level it is also important to determine if a small number of robots become especially relevant in linking among key sub-tasks and become "dominant" in terms of their importance across different overlays or sub-clusters in the agent community.

The remaining parts of the paper will be organized as following. In the immediate next section, namely "Multi-agent Facets," multi-agent collaboration dynamics will be described in terms of four critical facets: 1) Learning and adaptation patterns, 2) Agent involvement, 3) Dominant player, and 4)

[1]It is assumed that in most fully- or semi-automated operational settings, humans will be performing monitoring roles, particularly focused on quality control, providing occasional feedback (reward / penalty signals) and ensuring that robots perform all the tasks effectively and efficiently.

978-1-6654-7406-1/22/$31.00 ©2022 IEEE
DOI 10.1109/CogMI56440.2022.00020

Influence of tasks and data sharing on collaboration dynamics. The latter facets correspond to the key dimensions that directly influence coordination planning, composition, and configuration in agent-based collaborative tasks. In the subsequent section, "A Multi-agent Simulation System", a brief overview of a system which conducts document management tasks (e.g., labelling, indexing, retrieval) called Multi-agent Collaborative Classification of Information (MACCI or pronounced "maxi") will be presented. Following the latter section, in the section titled "Agent Dynamics Experiments", some simulation results that examine key parameters associated with four critical facets, such as learning rate, number of agents, and data shared among agents, and their impact of dynamic will be analyzed using task simulation results. In the final section, "Conclusion", some of the key observations and findings will be summarized and limitation of the multi-agent simulations for human-robot dynamics analysis will be discussed.

II. Multi-Agent Facets

In a distributed environment, agents represent local entities with limited information and interact with others to gain knowledge over time. As agents explore their external settings, they learn from the interactions and develop their understanding (model) about the environment. The model thus built will enable them to collaborate more effectively with other agents to accomplish the tasks they are assigned to, e.g. for information seeking or classification. Collective intelligence may emerge as agents learn to find their partners and together contribute to a potential global knowledge.

In addition, it is important to recognize the evolution of the environment as well as individual agents as they learn. A learning model that has been built on past experiences can only achieve some local optimum at the time being. As things continue to change, agents have to relearn and adjust their models to adapt to the changes continuously. Such adaptation should be part of the agent learning model based on continuous feedback from the community and other players in it.

Agents are not created equal. While no one has the global knowledge, some agents can be more capable than others, e.g. in terms of data and computing power they possess. They may also play different roles in the agent community, especially in a heterogeneous environment. As they learn and adapt to achieve their various objectives, their influence on the community and the ways they connect and work with others will also change. Dominant players will emerge and their influences can fluctuate over time. These dominant players as gate keeper or those with "weak ties" connect diverse agent communities together and help weave a global network structure enabling efficient collaboration and dissemination of information.

Our research has studied multi-agent systems for various tasks of information processing, and examined the facets of interaction and collaboration, learning and adaptation, the roles of dominant players, the evolution of the environment, and the influence of changes on agent behavior. In the next section, we describe a specific multi-agent system we developed for text classification tasks.

III. A Multi-Agent Simulation System

In the Multi-Agent Collaboration for Classification of Information (MACCI) project, we assume no centralized (global) training data for text classification [6]. Instead, agents possess limited information (partial knowledge) in the subsets of an entire data corpus. As such, each agent only has a local view and can only build information representations and classification models based on the data subset it has. When assigned a document to be classified, an agent will likely have to consult other agents and collaborate with them to identify the best label for the document. With reinforcement learning, agents interact with others and adapt to collaborate more efficiently and effectively. We discuss these major components below.

A. Distributed Document Representation

We use the classic Vector Space Model (VSM) and Term Frequency * Inverse Document Frequency (TF*IDF) for the representation of distributed data subsets. Within an agent's data subset, each document is tokenized into single words (terms) and vectorized using TF*IDF scores as term weights. In the end, a document d is represented by a numerical vector of terms where the weight of term t is computed by:

$$W_{dt} = TF_{dt} \cdot \log \frac{N'}{n_t} \tag{1}$$

where TF_{dt} is the number of occurrence of term t in document d (term frequency), N' is the total number of documents in the data (the subset hosted by the agent), and n_t is the number of documents containing t (document frequency). TF*IDF is an effective term weighting scheme and, in this research, is computed based on a local dictionary and statistics in the agent's data collection. A class vector representation is computed based on the centroid (averaged vector) of documents belonging to that class.

B. Collaborative Document Classification

When an agent is assigned a document to be classified, it will attempt to do the work first using its local document representation. Given the vector representation of TF*IDF scores described above, the agent will also vectorize the assigned document using the local dictionary and DF statistics. The document is then compared to each and every class vector in the local data subset using the Cosine similarity coefficient. Given two vectors $X = [x_1, .., x_m]$ and $Y = [y_1, .., y_m]$, cosine coefficient is computed by:

$$Cos(X,Y) = \frac{\sum x_t \cdot y_t}{\sqrt{\sum x_t^2} \cdot \sqrt{\sum y_t^2}} \tag{2}$$

which results in a normalized score in $[0, 1]$. Cosine is 0 when the two vectors are independent (orthogonal), and is 1 when the two vectors are identical in their directions.

During classification, the agent's local classifier computes cosine scores between the document and every class vector (again as the centroid of training documents in that class). The class with the highest cosine similarity will be chosen as the tentative label. However, if the score does not pass a predefined confidence threshold, the agent will reach out to other agents to identify potential better classes. The next section describes how this collaboration works and how agents learn to improve it over time.

C. Reinforcement Learning to Improve Collaboration

Ideally, an agent a will reach out to another agent b who has been helpful for a similar topic in the past. If agent b identifies a class label with sufficient confidence, it sends the result back to a. Otherwise, agent b will forward the request to yet another agent c, which will continue likewise. Through the chain of request-response interactions, agents collaborate with one another in the classification tasks.

Apparently, the number of hops (agents) one has to reach out to impacts the efficiency of this collaborative task. A major objective here is for the agents to collaborate more "wisely" so they can finish the tasks within a short time frame. In this research, we construct the agent with the capacity of reinforcement learning based on interaction, reward, and penalty. We implemented a classic reinforcement learning algorithm named Pursuit Learning, which we describe here [7].

Suppose an agent has a number of actions (N_a) it can perform, that is the number of neighbor agents it can contact for help on classification. A reward is given to that action (associated with that neighbor) depending on how successfully the classification has been performed.

Given an action probability vector P where a p_i score denotes the probability to choose a neighbor agent i and is initialized as $1/N_a$. Let Q be the vector of (estimated) probabilities where q_i is the estimated probability that action α_i (reaching out to neighbor i) will result in a reward, estimated as the cumulative average of rewards from past actions. The agent updates P and Q vectors after the result of each collaboration. To allow agents to explore, we include an exploration rate r, a probability with which an agent will choose a random neighbor.

At step k, assume $\alpha(k) = \alpha_j$ is the chosen action. The agent asks neighbor j for help and receive reinforcement of β_k, which is computed as the cosine similarity score between the assigned document and the class agent j helped identify. The current agent updates the number of times agent j has been contacted:

$$S_j(k+1) = S_j(k) + 1 \tag{3}$$

which is then used to update Q vector:

$$q_j(k+1) = \frac{S_j(k) \cdot q_j(k)}{S_j(k) + 1} \tag{4}$$

We update an E vector based on the estimated probabilities:

$$E_i(k) = \begin{cases} 1, & \text{if } q_i(k) = \max q_i(k) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

which is then used to update the final probabilities:

$$p_i(k+1) = p_i(k) + \lambda\big(E_i(k) - p_i(k)\big) \tag{6}$$

where λ is a super-parameter for the learning rate and convergence. With reinforcement learning, an agent will start with random exploration and gradually converge to the one that is most helpful.

D. Nearest Centroid Learning for Continuous Adaptation

The pursuit learning algorithm assumes a static view of the environment and assumes that agents helpful in the past will continue to be so in the future regardless of the content of the assigned document. We developed the Nearest Centroid Learning as a content-sensitive approach for agents to learn and adapt with the change of assigned document content. The assumption is that an agent will continue to be helpful for similar documents in the future.

In this approach, we implemented a centroid vector C_i representing each neighbor agent i and a probability vector P with each p_i corresponding to the probability of choosing neighbor i. For each successful response from a neighbor i, the centroid representation of that neighbor will include the latest classified document d, that is:

$$c_{it}(k+1) = \frac{k \cdot C_{it}(k) + d_t}{k+1} \tag{7}$$

where C_{it} is the centroid score of term t for neighbor i, $k{+}1$ is the number of successful classification tasks that neighbor i has helped accomplished, and d_t is the TF*IDF score of term t in assigned document d. When a new document d_{new} is assigned and the current agent needs to reach out for help, it will compute the P vector on the fly using the cosine similarity between the new document and neighbor centroids:

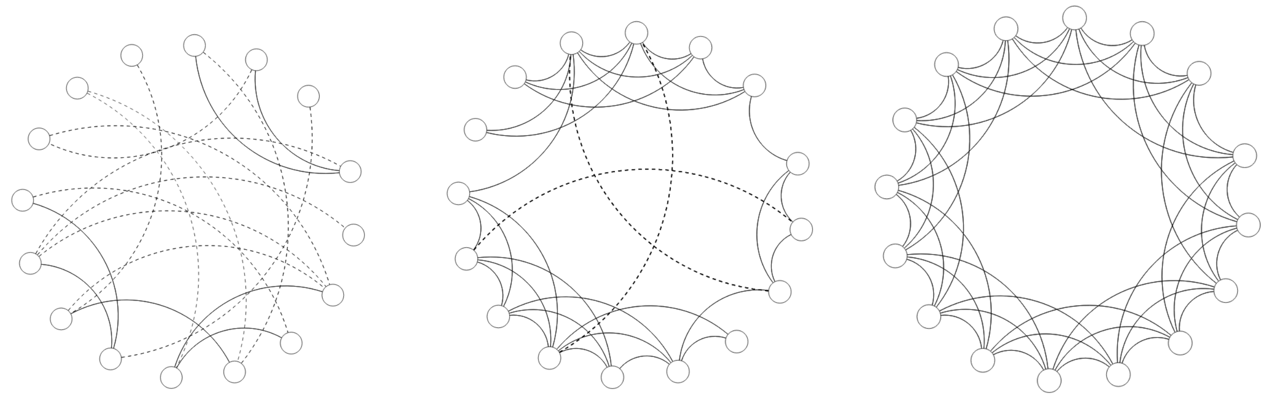
$$p_i = Cos(d_{new}, C_i) \tag{8}$$

The neighbor with the highest cosine similarity will be chosen to collaborate with on the new classification task. Over time, agents will learn about what topics (centroids) their neighbors are good at and identify collaborators selectively based on the content of the new document.

E. Experimental Results and Discussions

With the MACCI multi-agent system described above, we conduct text classification experiments on a benchmark text collection, namely the Reuters RCV-v2. Results show that agent collaboration improves classification accuracy and both learning algorithms facilitate collaboration, leading to a classification quality comparable to a centralized baseline [6]. Puisuit learning is more efficient in that it does not have to analyze the document contents and, surprisingly, is also competitive in terms of overall classification effectiveness.

Close examination of the agent collaboration shows interesting patterns and evolution of engagement over time. Learning led to increased intensity of collaboration and productivity. Key players did emerge through this learning and collaboration. The dominant agents, which became involved in many



(a) Random Network (weak) (b) Balanced Network (c) Regular Network (strong)

Fig. 1. Balancing between strong-ties and weak-ties based on a network clustering function $p_r = r^{\alpha}$. Showing the influence of the clustering exponent and a mid-level value is likely to achieve optimum collaboration among agents. (a) Weak clustering and many random connections with a small α, (b) Balanced network with a mid-level α, and (c) Strong clustering and strong ties only with a large α.

classification tasks, were associated with some of the major classes in the data [2]. As agents learn to improve, reaching out to these key players for help increases the probability of success and its associated reward.

IV. Agent Dynamics Experiments

The distributed classification research shows valuable details of agent collaboration, learning, and dynamics at a small scale (a community of 37 collaborative agents). In a related thread of research on decentralized information retrieval, where the objective was to find relevant information in a highly distributed environment, we experimented with up to $100,000$ agents hosting distributed subsets of 50 million web pages and studied agent dynamics together with issues of scalability [3].

In such a large-scale environment, it is no longer possible for each agent to maintain connections with and reach out to every other agents. With a limited capacity, each agent is confined with a small number of local connections. Nonetheless, they learn to identify better connections via self-organization and adaptation to a community structure where a search query (question) can be answered within a small number of hops (contacts).

We showed that efficient decentralized searches are possible in such a distributed network of agents only when the network structure is optimized. According to a phenomenon we refer to as the *Clustering Paradox*, there is a specific level of network clustering, i.e. grouping of neighbor agents with an ideal proportion of similar vs. dissimilar contents, where searches can be most efficient. Either over-clustering or under-clustering will lead to degraded search performances. Finding the ideal level is key to the success of agent collaboration for decentralized information seeking [4].

We simulate agent connectivity preferences with the following probability function:

$$p_r = r^{-\alpha} \tag{9}$$

where r is the topical distance between two agents (their data collections) and α is referred to as the clustering exponent. The α parameter controls the strength of clustering and how agents with similar contents group with each other to form communities.

Figure 1 illustrates the impact of α on the network structure and how searches can be supported in three types of networks. With a small α value, as shown in Figure 1 (a), inter-agent connections are rather random and searches will lose their orientation (structural guidance). On the other hand, with a large α value in Figure 1 (c), agents tend to group with similar agents only and form a regular network with strong ties, where searches can move *slowly* toward targets. The best network structure for decentralized searches is in between the two, when there are strong local connections to guide searches and weak (remote) connections for the searches to move from one community to another to speed up the process, as illustrated in Figure 1 (b).

Fortunately such an ideal network structure can be achieved without a global knowledge or “hard engineering” in a top-down manner. Rather, agents can interconnect locally and organically to construct this global structure from bottom up. Using a connectivity probability function that favors neighbors with similar contents but allows small chances to connect with those that are different, the network of these agent communities will emerge with the following two types of links (or ties): 1) strong ties that define local communities and serve as anchors (labels) to direct searches toward their relevant target, and 2) weak ties that connect different communities together and serve as “hubs” for searches to jump from one community to another.

When formulated properly, agents with local knowledge is capable of building such an efficient global structure collectively. They create both strong ties and weak ties in this process. While strong ties are like local paths for searches to “walk”, weak ties enable them to “fly.” Further analysis projects that efficient searches can be conducted in such a network of a billion agents [5].

V. Conclusions and Limitations

With growing interest in delegating complex and repetitive operations to robots it has become important to understand and plan such operations. The framework whereby a community of robots engaged in a joint activity are cast as a multi-agent network yields the possibility of analyzing critical dynamics and associated dimensions such as configuration, knowledge composition, and coordination. The limitations of such an approach are several, chief among them is the lack of "realism" and unexpected situations that may arise in the actual operational setting. Particularly, the environment may pose numerous challenges such as navigation, feedback noise (e.g., sub-optimal human signals), sabotage, bad actors, or accidental loss of participants. The latter factors need to be incorporated into the learning scheme and coordination planning. Past research in robotics has found that building smaller and specialized robots to conduct complex operations collaboratively is a prudent approach, however, understanding the behavior and the emergent interactions of such a robot community is an important challenges and it is worthy of further investigations [1]. This study was motivated in part to identify a method for flexible robotic coordination planning based on a multi-agent simulation approach. One critical finding discussed in this paper is that decomposing complex operations into smaller tasks/steps and supporting such operations based on a community of agents that can intelligently share responsibilities is a potentially a powerful approach. This study also showed how a large number of interacting variables in a robot community can be studied in a methodical manner and the study also unearthed an unusual coordination pattern namely the role of "weak" intermediaries in carrying out complex tasks. Finally, this paper demonstrated that highly complex operations decomposed into a robot community can be studied at scales ranging from few tens of participants to billions.

References

[1] R. Doriya, S. Mishra, and S. Gupta. A brief survey and analysis of multi-robot communication and coordination. In *International Conference on Computing, Communication & Automation*, pages 1014–1021, 2015.

[2] W. Ke and J. Mostafa. Visualizing multi-agent collaboration for classification of information. *Proceedings of the American Society for Information Science and Technology*, 45(1):1–4, 2008.

[3] W. Ke and J. Mostafa. Scalability of findability: Effective and efficient ir operations in large information networks. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, page 74–81, New York, NY, USA, 2010. Association for Computing Machinery.

[4] W. Ke and J. Mostafa. Studying the clustering paradox and scalability of search in highly distributed environments. *ACM Trans. Inf. Syst.*, 31(2), may 2013.

[5] W. Ke and J. Mostafa. Scalability analysis of distributed search in large peer-to-peer networks. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 909–914, 2016.

[6] W. Ke, J. Mostafa, and Y. Fu. Collaborative classifier agents: studying the impact of learning in distributed document classification. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '07, pages 428–437, New York, NY, USA, 2007. ACM.

[7] J. Mostafa and W. Lam. Automatic classification using supervised learning in a medical document filtering application. *Information Processing & Management*, 36(3):415–444, 2000.