Distributed Advice-Seeking on an Evolving Social Network

Golriz Rezaei, Jens Pfau and Michael Kirley
Department of Computer Science and Software Engineering
The University of Melbourne, Victoria, Australia
Email: {grezaei,jens,pfau,mkirley}@csse.unimelb.edu.au

Abstract—Within service-oriented architectures, components need to discover and interact with suitable services, which in turn can be mapped to particular subtasks. Similarly, in online social communities users may wish to discover information related to a particular interest that matches their personal preferences. These tasks are difficult when the number of available resources to select from is large, and when the actual characteristics of these resources do not become available until accessed, if they are made explicit at all. Given these requirements, we suggest that the direct exchange of “selection advice” between the users or components of the system can be beneficial. However, because individual requirements or preferences may be different, the choice from whom to accept advice is crucial. We capture this problem in an abstract agent-based model with a pool of heterogeneous resources and a population of agents with varying, but occasionally overlapping, preferences. Based on local information only, agents autonomously form connections with other agents who provide advice thereby improving resource selection. We study how this capability affects the match between agents’ preferences and the resources they access and how the underlying connection network co-evolves with advice exchange. Our results indicate that this framework promotes the formation of communities with agents of similar preferences and hence improves the overall agent utility, which is based on resource quality. This outcome is significant when the agents’ personal knowledge about the resource pool is small. These results highlight how the cooperation of components or users in concrete systems can be facilitated in order to improve system performance or user experience.

Keywords—evolving social networks; advice seeking; self-adaptation

1. INTRODUCTION

The development of distributed infrastructure technology has given rise to systems such as service-oriented techniques, web services, and online social communities. Typically, these systems are composed of a large number of providers offering different but possibly overlapping services [18]. Consider two disparate examples: specialized search engines that can analyze and identify protein streams from different perspectives [14], and online communities such as Netflix, which offer their users access to information about various entertainment products. Generally, the exact characteristics of services or items in such systems might be inaccessible to the prospective users. However, users typically seek to select resources that “best fit” their current requirements or preferences despite the fact that in some instances they might be unaware of their true preferences or may be reluctant to communicate them.

Web services are typically listed in centrally available directories that users can query in order to identify suitable services [4], [15]. Although this approach has been shown to be effective in the context of a few providers serving a few users, this centralized approach does not scale up easily to a large number of users. Furthermore, this approach requires that users be both willing and able to communicate all of their requirements to a central authority. Similarly, in recommender systems recommendations are typically issued from a central component that bears sole responsibility for the operational functionality of the system [19], [20]. In addition, the subjective user ratings used for recommendations are typically not based on specific users interests or characteristics.

In this paper, we introduce an abstract model to simulate a population of agents with heterogeneous preferences that repeatedly select among resources from a publicly available pool. The match between the preferences of an agent and the characteristics of a particular selected resource yields a certain utility for this agent. The goal of the agents is to maximize their long-term utility during a limited number of subsequent selection steps. The subjective utility of a resource is unknown to an agent until it selects this resource for the first time. Moreover, the true characteristics of resources are not revealed to the agents at all. Therefore, identifying appropriate resources among a large number of alternatives in a limited period of time is a challenge.

The rationale behind our approach, is based on the notion that it may be beneficial to allow users to seek advice from others before making a decision. However, identifying users with compatible preferences is often difficult because there might be a large number of them, their preferences might not be publicly available, and/or they may not be in a position to make their own preferences explicit. Therefore, users adopt heuristics to help identify which of the other users are able to provide suitable advice. Without a central component that hosts global information about the users in the system, users need to be able to exploit their local knowledge about others to link with those that have similar preferences.

In related work [9], [20], models have been proposed where the underlying connection graph was fixed and did not evolve. In our framework, we equip agents with the
capacity to exchange their experienced personal utility for selected resources but not their true preferences. By learning their own subjective utility for a resource based on advice received, agents can assess the accuracy of that advice and ultimately decide whether to retain the contact with the reporting agent or drop it. Furthermore, agents are able to ask their direct contacts for referrals in order to make connections with other agents.

We investigate how the evolving social network can improve the utility gained by the agents and how their interactions may change their social relationships. Our results clearly demonstrate that strongly connected communities of agents with similar preferences emerge. This leads to a higher utility for the agents especially during the initial period when they are still unaware about their subjectively “best” resources. However, we also find that the level of heterogeneity among agents and available resources affects the gained utility in an adverse way.

The remainder of this paper is organized as follow. In Section II, we review related work in two areas: evolving social networks and advice exchange. Then in Section III, we present the details of our model, followed by an experimental evaluation in Section IV and a discussion in Section V. We close this paper in Section VI.

II. BACKGROUND AND RELATED WORK

A. Evolving social networks

Social networks and their structures play an important role in many real-world and multi-agent systems. Typically, the objectives of studies in this domain have focused on either describing the network’s topology or attempting to understand the system’s behavior as a function of network structure [6]. For example, studies have investigated how certain types of network topologies emerge [21], [1]. Other studies have shown how certain topological properties, which couple the interactions of entities in the system, impact on the emergent behavioral dynamics [13], [11]. However, in most real-world networks there is in fact a bidirectional feedback relation between the network’s topology and the system’s behavior [6]. Consequently, the co-evolution of topology and behavior has recently gained increasing attention.

One area that has provided significant insights into the co-evolutionary dynamics described above is evolutionary game theory [16]. Here, agents are capable of changing their strategies and adapting their local network structure as a direct consequence of the games’ outcome. The network’s structure and the agents’ strategies co-evolve with repeated cycles of interactions, strategy modifications, and network adaptations. In these models, social contacts serve as a resource that needs to be managed in order to improve long-term payoff gains [6].

B. Advice Exchange and Recommender Systems

The exchange of information between interacting agents can be used to improve the overall performance in multi-agent systems [2], [17]. For example, [12] have investigated how advice-exchange mechanisms can improve the performance of a group of agents learning to solve related problems. Results of several experiments show that information exchange can improve the performance of the learning algorithms tested. However, the advantage seems to vanish for harder problems and more heterogeneous tasks.

Recommender systems provide a different domain in which users rely on suggestions of particular products (or resource or information) drawn from a large pool of products that are considered to fit a given user’s interests or preferences [10]. In the collaborative filtering approach, if a number of users have already “appreciated” a similar set of products, they are likely to share interest in other products as well [10], [3]. Users indicate their appreciation for a product by reporting a subjective product rating to the system. The utility of collaborative filtering systems clearly relies on the overlap of users’ interests and is likely to decrease with a more heterogeneous product catalog, with more diversified and less overlapping user preferences, and with a smaller number of participants. A disadvantage of these systems, is the fact that they suffer from the dependence on a central recommendation component [10].

In distributed recommender systems, recommendations are not issued by a central authority but users exchange recommendations with each other directly [5], [10]. Thus users need to find the right contacts to link to. In [20], a model of a distributed recommender system is outlined that combines the concepts of social networking and trust relationships. Agents are connected by a fixed random social network. Before considering the purchase of a product, agents first query their social contacts for recommendations and then evaluate the accuracy of these recommendations after they experience their subjective utility. Agents keep track of the accuracy of recommendations that they have received from their social contacts so far. This enables them to decide from whom to accept recommendations. In a different model, self-interested agents iteratively decide to trade recommendations or to refuse exchange [19]. The results of an experimental study show that engaging in a dyadic exchange is the rational choice as long as both agents believe that they share similar interests. Agents decide based on their past interactions when and with whom to exchange recommendations.

A significant difference between the models described in [20] and [19] and our model is that they do not make use of the implicitly underlying network structure apart from keeping track of the recommendations they have received from their social contacts. They do not exploit the existing structure to optimize their position within the social network.
to connect with those that have similar preferences.

III. THE MODEL

A. Overview

Our model represents a world consisting of a finite and fixed set of agents $\mathcal{A}$ and a larger but still finite set of resources $\mathcal{R}$. These resources represent service providers, products, or anything that agents are interested in within a specific domain. Each resource $r \in \mathcal{R}$ is characterized by particular features represented by an $n$-dimensional binary feature vector $f_r \in \{0, 1\}^n$, where an entry of 1 denotes that this resource has this feature. In the example of web services, features symbolize the specific services a provider may offer as well as other implicit characteristics such as its network bandwidth, computational performance, and overall quality of service. The agent population is heterogeneous in the sense that agents are interested in different features. The preference of an agent $a \in \mathcal{A}$ is characterized by an $n$-dimensional binary preference vector $p_a \in \{0, 1\}^n$, where an entry of 1 denotes the interest of the agent in resources with this specific feature. The description of features and preferences is similar to the one in [19].

At each time step, each agent selects a single resource and receives a real-valued utility as a result. However, agents do not learn about the true features of the selected resource. The utility formalization used here is an abstraction of the agents’ “appreciation” for a particular resource. We use this approach because we assume that not all of the features of the resource will be directly accessible in all situations. The utility an agent receives from selecting a particular resource is therefore based on the match between its preference vector and the resource’s feature vector. In other words, the utility is calculated based on the level of satisfaction with the selected resource. In the running example, the utility an agent receives from interacting with a particular service provider depends on the match between the services and characteristics required by the agent and those offered by the provider.

In general, the objective of every agent is to optimize its selection process to maximize the expected long-term utility in a limited time. However, with limited knowledge about the resources and their features, agents need to explore various resources (i.e. selecting random resources) to identify the best possible alternatives. As we discussed earlier, the number of resources might be large, the true characteristics of resources unavailable, and agents unaware about their exact preferences. In such a situation, agents are unlikely to perform efficiently and gain the highest possible utility without additional support.

To enhance the agents’ performance in this framework, agents are able to seek advice and base their resource selection on the experience of others. Agents may have heterogeneous preferences, consequently they may evaluate resources differently. Therefore, the choice of who they should query affects the quality of advice received. Effectively, agents need to form connections with others that have similar preferences. However, agents are neither assumed to be able to specify their own preferences nor to be willing to communicate them; and the exact features of resources are not assumed to be available to the agents. Therefore, the only information agents communicate are their subjective utilities for resources.

Connections between agents are described by a graph structure $G = (\mathcal{A}, \mathcal{E})$ where nodes $\mathcal{A}$ represent the agents and edges $\mathcal{E} \subseteq \mathcal{A} \times \mathcal{A}$ represent the existing connections between the agents. The graph is directed and weighted. The weight $w(a, b) \in [0, 1]$ on an edge between agents $a$ and $b$ represents the quality of the advice received so far by agent $a$ from agent $b$.

We now explain the model in detail, specifying how the agents interact with and adapt their social contacts.

B. Initialization

First, the population of agents is initialized with random preference vectors as is the pool of resources with random feature vectors. Here we have two scenarios: In the first scenario, the population consists only of random agents who are not restricted to any form of social structure and query others in the entire population randomly. In the other scenario, the population consists only of social agents who are able to self-organize themselves on a network and seek advice from social contacts. Initially, every social agent has at least one but not more than $l$ outgoing edges with default weight $w_{fl}$ (here set to 0.5). At initialization, the graph is connected without any isolated nodes. There is no network for random agent populations. The simulation then runs for a certain number of iterations, whose four distinct phases we will describe in the following.

C. Exploration and Exploitation

In every round, each agent has to decide whether to select a resource based on its knowledge about the utilities of the resources it has selected so far or whether to query others for advice. In that, the agent either exploits its existing knowledge or explores other possibilities that potentially might lead to the improvement of its existing knowledge. This decision is probabilistic and depends on the richness of the agent’s acquired knowledge. Intuitively, the more resource utilities an agent knows about, the more confident the agent is to rely on its own knowledge. Therefore, the probability of an agent $a$ relying on its own knowledge is defined as the proportion of the number of resources the agent has already observed $(R_a)$ to the total number of resources in the pool $(\frac{|R_a|}{|R|})$.

If the agent is to exploit its own knowledge, it accesses the resource with the largest utility it knows so far. Otherwise, it seeks advice from others in the form of resource-utility tuples. To provide a fair opportunity for all, each agent can
query exactly $l$ other agents. Random agents query $l$ random other agents, while social agents query their $l$ social contacts. In case they do not have $l$ contacts, they query additional agents randomly to start with. Each advisor then suggests the most beneficial resources it has found so far. For a resource to be eligible for suggestion, the known utility needs to exceed a threshold $\text{thr}_{\text{dis}}$. The advice-seeking agent then filters these advices considering only resources it has not accessed so far.

D. Advice Selection

In this step, the agent follows one of the suggestions received in the previous step. The selection process is probabilistic and is carried out in two steps. First, the advice-seeking agent selects probabilistically one of the advisors based on their link’s weight and then selects probabilistically among that advisor’s suggestions based on the reported utilities. Ultimately, the selected resource is accessed and the agent receives its subjective utility. If no advice has been issued, the agent selects a random resource from the pool.

The actual utility an agent $a$ receives for resource $r$ (denoted as $u_a(r)$), is defined by a function of similarity between the preference vector of the agent, $p_a$, and the feature vector of the resource, $f_r$. The more similar the vectors are, the more utility the agent gets from that resource.

We assess similarity by a normalized Hamming distance $d(f_r, p_a)$ with values mapped to the range $[-1, 1]$:

$$u_a(r) = \frac{d(f_r, p_a)}{n} \times 2 - 1 \quad (1)$$

Positive values indicate that the utility is better than what was expected on average by a random selection while negative values indicate that the agent would have done better on average by selecting randomly.

E. Assessment

This step and the next one are network-structure related processes and thus are only applicable to social agents. The assessment step enables social agents to learn from their interactions with other agents and to adjust the weight of their links. Following a particular suggestion can be a positive or negative experience depending on the actual utility received. We consider positive interactions as those when the absolute difference between the actual and the advised utility is less than $\text{thr}_{\text{dis}}$. Multiple advisors might have reported the selected resource and the agent then adjusts the weight of the links it has with these advisors. The link weight from agent $a$ to agent $b$ is a function of the proportion of positive interactions with agent $b$ to the total number of interactions with this agent:

$$w(a, b) = \frac{e_{pos}(a, b) + 1}{e_{pos}(a, b) + e_{neg}(a, b) + 2} \quad (2)$$

where $e_{pos}(a, b)$ is the number of positive experiences and $e_{neg}(a, b)$ the number of negative ones. If an edge weight drops below a tolerance threshold $k$ (here set to 0.4), the edge is removed completely and a slot for a new link becomes available. The parameter $k$ determines how tolerable/forgiving an agent is.

F. Network adaptation

In this step, social agents have the opportunity to change their links. With probability $\epsilon_a$, an agent adds an edge with default weight $w_d$ to an agent randomly chosen from the population that is not already a contact. If this causes the agent to have more than $l$ outgoing links, the agent reviews its connections and only retains the $l$ strongest links. With probability $1 - \epsilon_a$, the agent asks its best contact to propose one of its contacts randomly as a new contact. This reflects the notion of trust propagation [19], [10], which means an agent referring another agent to a third party.

Due to the last two steps, agents have the chance to eventually make links with agents that have similar or even equal preferences. Building a community of similar-minded agents helps them to spot beneficial resources faster.

Figure 1 shows snapshots of this process in different time steps. At the beginning (Figure 1a), agents are linked randomly (described in the initialization step). After a few iterations (Figure 1b), agents have found some similar-minded peers to connect with. In Section IV, we show how this can support the selection of resources and improve overall utility. After 100 iterations (Figure 1c), almost all agents are arranged in clear-cut communities. There are a few links between communities whose members share some parts of their preference vectors. There are only very few links between agents with completely different preferences. These inter-community links have been formed as a result of the random creation of links, which allows agents to go on exploring the possibility of linking with other agents.

IV. SIMULATIONS

To study the dynamics of the proposed model, we run Monte-Carlo simulations with various parameter settings. Specifically, we wish to compare the social and random agent scenarios. We also seek to study whether there is a relation between community structures and received utilities and how the level of heterogeneity among agents’ preferences and resources’ features affects utility gain. To answer these questions, we introduce some metrics to measure gained utility from different perspectives, thus providing a means to compare the performance of different instantiations of the model. After that, we present our experiments and results.

A. Metrics

Average Utility is the average of the utilities $u_a(\cdot)$ that all $N$ agents have gained at a particular iteration $t$:

$$\bar{u}_t = \frac{1}{|A|} \sum_{a \in A} u_a(\cdot)$$
For some experiments, the overall performance of the system over a period of time is more interesting than the performance in a given iteration. Hence, we also calculate the average accumulated utility $\bar{U}_T$ of all agents. It is the average of the utilities that all agents have gained after $T$ iterations.

In our model, there is a difference between the reported utility of an advised resource and its actual utility experienced by the advice-seeking agent due to the heterogeneity of the agents’ preferences. To measure the overall quality of received advice, we need a metric to measure these differences. Mean Absolute Error (MAE) is a common metric in evaluating the accuracy of recommender systems.

Mean Absolute Error (MAE) is a common metric in evaluating the accuracy of recommender systems in their predictions [7], [10]. We use this as a starting point.

For all agents $a \in A_r(t) \subseteq A$ whose resource selections followed the given advice at time $t$, we record the difference between the actual utility $u_a(r_a)$ and the reported utility $\hat{u}_a(r_a)$ for the selected resource $r_a$. This yields a metric for the error rate $\Delta_t$ of all advice that have lead to a selection at time $t$:

$$\Delta_t = \frac{1}{|A_r(t)|} \sum_{a \in A_r(t)} |u_a(r_a) - \hat{u}_a(r_a)|$$

The efficiency metric illustrates the utility gained by the agent population, taking into account the number of resources where utility had to be learned for that. Intuitively, a single agent is more efficient the higher its gained utility and the less resources it had to discover for that. Let $|R|$ the total number of resources in the pool, $|R_a|$ the number of resources discovered by agent $a$, $T$ the current timestep, and $U^T_a$ the accumulated utility gained by agent $a$ until timestep $T$. We define the efficiency of an agent as the ratio of its accumulated utility to its observation rate $U^T_a / |R_a| / |R| = U^T_a / |R_a| / |R_a|$. Because the maximum utility to be gained from a selection is 1 and because there are exactly $T$ selections for every agent until time $T$, the maximum efficiency is $T/|R|$, considering that at least one resource has to be discovered for this metric to be meaningful. Given this, we can calculate the normalized average efficiency of all agents at time $T$:

$$\bar{\Sigma}_T = \frac{1}{|A|} \sum_{a \in A} U^T_a / |R_a| / |R| = \frac{1}{|A|} \sum_{a \in A} U^T_a / |R_a| / |R_a|$$

This provides a fair measurement for comparing utility gain between different scenarios with a different number of available resources.

B. Experimental Setup

We experiment with two scenarios: When the population consists of social agents only and when it consists of random agents only (see Section III-C). We investigate two population sizes: a small population size (100 agents) and a large population size (300 agents). We study both scenarios with different levels of environmental complexity, realized by a varying number of available resources (1000, 5000, 10000, and 50000), and with different levels of heterogeneity, realized by a varying length of preference and feature vectors (2, 3, 4, and 5). It is worth noting that with unlimited time both types of agents eventually can find the best matching resources through exhaustive exploration. However, we seek to study utility gain in a limited amount of time and thus restrict our analysis to the first 1000 iterations only. All of the following experiments are repeated for 30 independent runs.

C. Basic model behaviour

In the first set of experiments, we investigate whether social agents are able to gain higher utilities. The population size is set to 100 with a moderate level of heterogeneity...
the social agent population in every iteration is significantly of social contacts for advice-seeking in this model facilitates This result suggests that the exploitation and optimization

utility gain and outperforms unstructured advice-seeking .

number of resources is

deploy

Figure 2: Comparing the performance of a social (solid line) and random (dashed line) agent population in terms of average utility and error rate.

realized by a feature and preference vector length of 3. The number of resources is 5000.

As shown in Figure 2a, the average utility gained by the social agent population in every iteration is significantly higher than the one gained by the random agent population. This result suggests that the exploitation and optimization of social contacts for advice-seeking in this model facilitates utility gain and outperforms unstructured advice-seeking.

Comparing the error rate of both scenarios (Figure 2b), we learn that the social agent scenario has a significantly lower error rate when suggesting resources.

D. The influence of environmental complexity

To expand on the previous experiment, we investigate the efficiency of the social and random scenarios when agents face more complex environments (larger resource pool). We deploy 100 agents and a varying number of resources (1000, 5000, 10000, and 50000). Feature and preference vector length is set to 3, which corresponds to a moderate level of heterogeneity. Figure 3 shows that for all resource pool sizes social agents are more efficient in gaining utility. The difference increases with resource pool size.

When the number of resources is increased by 50 times from 1000 resources to 50000 resources, the efficiency of the random agent population drastically decreases from almost $6 \times 10^{-4}$ to nearly $0.1 \times 10^{-4}$. However, the efficiency of the social agent population only drops by half from almost $7.7 \times 10^{-4}$ to approximately $3.5 \times 10^{-4}$. Higher efficiency implies that with the same discovery effort, the social scenario generally leads to higher utility than the random scenario. This result also emphasizes that even if the resource pool is huge, the social agent population can still explore the available resources efficiently to some reasonable extent in contrast to the population of random agents.

E. The influence of heterogeneity

Considering that finding similar-minded agents in the population plays an important role in the performance of social agents, in the next set of experiments we study how heterogeneity in resources and preferences can affect the performance of social agents.

We run experiments with a feature and preference vector length of 2, 3, 4, and 5 in the social scenario with two population sizes: small (100 agents) and large (300 agents). We study the behavior of the model in terms of accumulated utility by the end of the simulation (iteration 1000). The number of available resources is fixed to 5000.

As shown in Figure 4, the accumulated utility generally decreases with heterogeneity in the population. This is quite an obvious result since in more diverse populations, agents have less chance to meet similar-minded agents and cannot establish advantageous links. With a large heterogeneity, most of the received advice is misleading and reduces the efficiency of resource selection. In other words, with increasing heterogeneity the performance of social agents is expected to approach the performance of random agents. However, as we increase the size of the population, we observe some increase in gained utility. Accumulated utility improves for feature vector lengths 2, 3, and 4 significantly.

In the largest level of heterogeneity (vector length 5), the advantage is less pronounced. Generally, these results indicate that the probability of meeting similar-minded agents in the population influences overall utility gain.
F. Analyzing the underlying network

In the following, we extend our analysis to the co-evolution of the system’s behavior and the structural properties of the network. In particular, we are interested in the modularity of the network, which specifies how well the network is partitioned into distinct communities [8].

For the same scenarios as in the previous experiments, the development of average utility over time is reported in Figures 5a and 5b. According to these results, average utility gain is lower for higher levels of heterogeneity. These plots confirm the previously reported result that an increase in population size can compensate for this to some extent. The development of modularity over time is depicted in Figures 5c and 5d. The modularity of the network increases quickly to a high value, accompanying the increase in utility gain. This suggests that social agents in general are able to group into distinct communities of similar-minded agents quickly.

However, the results in Figure 5c indicate that community-building in the case of a small population is less pronounced for larger levels of heterogeneity. Although the same trend can be observed in the large population (Figure 5d) at the beginning of the simulation, eventually the community structure becomes similarly pronounced for all levels of heterogeneity. Consequently, larger populations are better able to group into communities of agents with similar preferences and thus achieve higher utilities on average.

Even though not reported here due to space constraints, the average path lengths of the evolving networks become less than 6 in the first 1000 iterations of all experiments with almost all parameter settings. The only exception is an agent population of size 300, a resource pool of size 1000 and a feature length of 5. However, a small path lengths implies that agents can change their position within the network drastically in a short amount of time.

V. DISCUSSION

Our results show that when agents seek advice from other agents and adapt their social contacts in response to the quality of advice received, they can optimize their resource selection more quickly than those agents who do not have this capability. There is a relation between the evolution of the community structure of the underlying network and the utility gained by the agents. In particular, the access to the resource pool is much more efficient in the case of agents that make use of social contacts, especially when the resource pool is large. However, the results also indicate that the advantage decreases with the level of heterogeneity of the available resources and agents preferences.

These results have a number of interesting implications for the development and operation of real-life systems. Let us consider again the example of an online community for movie enthusiasts that has the requirement for a decentralized organization. Our results suggest that simply by empowering users to make and adjust contacts with other users autonomously based only on local knowledge, their movie selection and thus experience with the system can be largely improved. The concrete implementation could link a new user to a few randomly selected other users initially. By exchanging advice and identifying which of these contacts have similar preferences, the new user can decide which of them to ask for referrals to other users. By this process alone, we expect the new user to eventually connect with that part of the community that has the most similar interests. Because the average path length is likely to remain small, we can expect that the optimization of the network position takes only a few link adaptations. Moreover, because our results predict that smaller communities within the whole community will emerge, the system could be made capable of recognizing these communities autonomously and catering for their specific needs. For example, multi-directional means of communication such as bulletin boards could be made available to such communities automatically. These would complement the bi-directional communication between users.

However, our results also indicate that in order for users to benefit from advice exchange, enough users need to be
active and the operators should not expand their program to other product ranges and target groups unless they expect to maintain a sufficient community size and hence overlapping interests.

Also most real large-scale systems are open and dynamic. This means that users may join or leave the system at will and in the case of service-oriented architectures, this also applies to the resources or services themselves. Users might change their interests and resources might change their characteristics. We did not consider these factors in our study in order not to distract from our main points. As a future extension, we are interested in applying the proposed framework in dynamic environments to test the robustness and usefulness of the approach. We have also made the assumption that agents, users, or system components might access the same resource over and over again to optimize their utility gain. While it is realistic in the example of service-oriented architectures, movie enthusiasts are unlikely to watch the same movie repeatedly. Yet this assumption does not negate our main point that local network adaptations in distributed advice-seeking applications can be helpful in improving system performance.

VI. CONCLUSION

This paper has introduced an abstract distributed system model, in which users or components with different interests or requirements can exchange advice about the access to a large pool of resources such as movies or service providers. In this model, agents are enabled to autonomously make links with other agents that have similar preferences. We have studied the behavior of the system under varying conditions and we have analyzed the co-evolution of the system’s behavior and of the underlying connection graph. Simply by exploiting knowledge about their direct social contacts, agents manage to optimize their links and form communities of like-minded agents that share advice effectively and thus improve the benefit of the overall system. We have discussed in detail the implications for the implementation of concrete distributed systems, suggesting that the combination of advice exchange between users and the local adaptation of connections might be helpful under certain conditions.

REFERENCES