Plan, Activity, and Intent Recognition: Papers from the 2011 AAAI Workshop (WS-11-16)

Detecting and Identifying Coalitions

Reid Kerr and **Robin Cohen**
University of Waterloo
Waterloo, Ontario, Canada
{rckerr,rcohen}@uwaterloo.ca

Abstract

In many multiagent scenarios, groups of participants (known as *coalitions*) may attempt to cooperate, seeking to increase the benefits realized by the members. Depending on the scenario, such cooperation may be benign, or may be unwelcome or even forbidden (often called *collusion*). Coalitions can present a problem for many multiagent systems, potentially undermining the intended operation of systems. In this paper, we present a technique for detecting the presence of coalitions (malicious or otherwise), and identifying their members. Our technique employs clustering in *benefit space*, a high-dimensional feature space reflecting the benefit flowing between agents, in order to identify groups of agents who are similar in terms of the agents they are favoring. A statistical approach is then used to characterize candidate clusters, identifying as coalitions those groups that favor their own members to a much greater degree than the general population. We believe that our approach is applicable to a wide range of domains. Here, we demonstrate its effectiveness within a simulated marketplace making use of a trust and reputation system to cope with dishonest sellers. Many trust and reputation proposals readily acknowledge their ineffectiveness in the face of collusion, providing one example of the importance of the problem. While certain aspects of coalitions have received significant attention (e.g., formation, stability, etc.), relatively little research has focused on the problem of coalition identification. We believe our research represents an important step towards addressing the challenges posed by coalitions.

Introduction

The field of multiagent systems is concerned with systems where multiple, independent entities interact with one another. While multiagent systems are often classified as either cooperative or competitive, the reality is often more complex. For example, in electronic marketplace scenarios composed of agents that buy and sell goods, each agent may be a self-interested utility maximizer, but at the same time depend on other agents (its buying/selling partners) to achieve its goals. In such a scenario, 'cooperation' of a sort (in the form of behaving honestly) may be critical to success—honest agents may be more likely to find trading partners in the future. Thus, agents may display a high degree of mutually beneficial behavior, despite being independent.

Cooperation may go further, however. Agents may intentionally seek to coordinate their activities in some way, with the expectation that coordinated effort will further their goals (whether shared or independent). We refer to a group of agents engaging in coordinated activity as a *team*, or a *coalition*. Coalitions may seek to increase their scores in a game, enhance profits, improve competitive positions, provide protection from other agents, damage competitors, etc., depending on the scenario.

Coordinated activity by a team may be benign, or even desirable. A real-world example would be a 'neighborhood watch' program, where individuals cooperate for their mutual security, but no harm is done to others. In many scenarios, however, agents may cooperate despite the fact that this may be unwelcome or forbidden. For example, players may cooperate in a game such as poker (which is intended to be played individually) to increase their winnings, at the expense of other players. Such activity is often considered a form of cheating; we refer to it specifically as *collusion*.

In many situations, it can be useful to detect the presence of teams, and/or to identify team members. For example, if one can determine which players are colluding in a game, one might penalize or expel those players, act to hinder their activities, etc. (Alternatively, the power of detection might serve as a deterrent to such cooperation.)

In this paper, we present a technique for detecting the presence of coalitions in an environment, and for identifying coalition members. We believe that our approach is broadly applicable to a range of activities and scenarios. It is worth noting, however, that identifying teams is likely to be more difficult (and potentially more valuable) where such teams are unwelcome; colluding agents may wish to go undetected. For this reason, we focus on such scenarios here. We note several examples, to illustrate the scope and real-world importance of this issue:

- Trust and reputation systems are employed in multiagent systems where an agent's success depends to a large degree on the reliability or trustworthiness of the agents with whom it chooses to interact. Such systems are susceptible to collusion, as detailed in the Related Work section.
- 'Shilling' and 'astroturfing', where false opinions are given by coalition members to create the (false) impression of widespread public support (or opposition) for a

position, product, etc.

- Cheating in games, particularly forms of gambling such as poker. Such collusion is a significant problem, even reported in the mainstream media (e.g., (Katz 2005)).
- Insurgent activity in a military setting, or terrorism. Members of such a coalition attempt to 'blend in' with the population, so we cannot directly observe group membership.

Scenario Characteristics Certain distinguishing characteristics of this range of scenarios should be noted. First, we can identify each individual in the environment (although we might not know who is actually controlling the identity, e.g., in the case of a user account). Second, while certain actions are observable, others are not. In particular, we have no access to communications between colluding parties, nor knowledge of their sharing resources or sharing of benefits outside the system. Finally, and importantly, we assume no knowledge of the plans or tactics that may be employed; even the goals of a possible coalition may be unknown.

Related Work

Trust and reputation systems aim to help agents choose trustworthy partners and/or avoid untrustworthy ones. Typically, agents provide reviews of their experiences with other participants; when deciding whether or not to trust a potential partner, an agent can make use of the information in these reviews. While a multitude of trust and reputation systems have been proposed (e.g., (Jøsang and Ismail 2002; Kerr and Cohen 2010b; Teacy et al. 2006)), there is broad acknowledgement by researchers of the vulnerability of trust and reputation proposals to coalitions (e.g., (Dellarocas 2000; Jurca and Faltings 2007)). While efforts are made to cope with unreliable reviews, systems are often susceptible to two well-known forms of collusion (Dellarocas 2000):

- ***Ballot-stuffing***, where coalition agents give false positive reviews to their teammates, in order to inflate the reputations of the recipients.
- ***Bad-mouthing***, where coalition agents give false negative reviews to competitors, in order to damage the reputation of the competitors.

The goal of both of these attacks is to improve team members' chances of being selected by another agent.

Because of the importance of collusion to trust and reputation systems, and because they have been well-studied, we use them here as an example to demonstrate our technique.

An area with obvious topical relationship to our work is that of coalition formation and stability within the field of multiagent systems. This work is often approached from a game-theoretic perspective (e.g., (Osborne and Rubinstein 1994; Shehory and Kraus 1999)), exploring the conditions under which coalitions form, algorithms for formation, and requirements for a coalition to persist. While insight into these issues might be useful in the detection of coalitions, existing work is difficult to apply to our problem. For example, such work often makes assumptions such as the capabilities of agents being known to one another, the distribution of payouts being known, and the value earned by a coalition depending only on the actions of coalition members.

Work in multiagent plan recognition and behavior recognition (e.g., (Tambe 1996; Sukthankar and Sycara 2006; 2007)) considers scenarios where multiple agents are observed attempting to execute a joint plan. The goal is to infer the plan being executed from the observations. In such cases, teams may break into subteams to perform tasks; related to our work, these proposals may also attempt to identify membership of subteams. While sharing important concerns with our work, these proposals deal with fundamentally different scenarios. In particular, this work assumes a known plan library: pattern matching is used to identify plans (and team assignments). In contrast, we assume no knowledge of the plans in use.

The research most similar to ours appears to have come in the field of collaborative filtering. Collaborative filtering systems aid users in making selections, by making recommendations based on the opinions of others with similar tastes. Such systems are commonly encountered on the internet today, recommending books, music, movies, etc.

Recent research has targeted the problem of *shilling*—the creation of false user profiles/accounts containing ratings intended to manipulate the results of the recommendation algorithms (e.g., (Burke et al. 2006; Mehta and Nejdl 2009)). Two general types of attacks are noted: *push* attacks, intended to increase the recommendations of an item, and *nuke* attacks, intended to decrease recommendations. While these correspond roughly to ballot-stuffing and bad-mouthing, respectively, there are key differences. In a reputation system, an agent might select or weight ratings based on its relationships with the reviewers, the past accuracy of the reviewers, etc. In contrast, in a collaborative filtering system the recommendations are based on the reviews of those with similar tastes to the user (i.e., those having rendered similar opinions). Accordingly, the attack strategy is different for collaborative filtering: an attacker seeks to build shill profiles that will be as similar to (honest) users as possible, so as to strongly impact their recommendations (Mehta and Nejdl 2009). Because of this, leading approaches to detecting shills focus on the extreme consistency of the shill profiles.

A focus of social network analysis has been on discovering communities and groups within larger networks (e.g., (Girvan and Newman 2002; Newman and Girvan 2004)). Such work often uses properties such as frequency of interaction and degree of connectedness in order to identify groups of users that are related. Unfortunately, such work does not appear to be directly applicable to our problem. For example, consider a ballot-stuffing attack. Members of the coalitions may use a small number of fake positive reviews to inflate each other's reputations; this reputation may be used to earn a large number of profitable sales from outsiders. In this case, coalition members may be more connected, have a greater number of interactions, etc., with outsiders than with coalition partners.

Method

As noted, we are concerned with scenarios where no plan library is available. Without the ability to match actions

against known patterns, we must rely on fundamental properties of the observable actions themselves. Typically, a self-interested agent will be part of a coalition because it expects some net benefit from doing so. Because benefit seems to be fundamental to the existence of coalitions, we use it as the basis of our technique. Specifically, we look at the 'flow' of benefit between agents: actions of one agent that benefit another, and/or transfer of benefit from one to another.

Not all coalition members will realize observable benefit. For example, a plan may not work out as expected for every member. More to the point, it may be the case that most or all coalition members benefit, but that not all of the benefit flows are observable—some benefit may be transferred privately amongst members. In the situations with which we are concerned, however, coalitions seek to gain net benefit from or relative to other participants in the system (e.g., earn additional profits, gain extra points, etc.), and improving the net position of the coalition requires taking observable actions (e.g., making sales, attacking enemies, etc.). In such scenarios, many forms of benefit flowing from one agent to another may be observable. The actions that constitute benefit are specific to a scenario, and domain expertise is likely required to identify them.

Within a dynamic environment, coalition members may help other members; they may also harm them (e.g., by accident, or to mask their relationship). Similarly, coalition members may harm outsiders, but they may also benefit them (e.g., by making a purchase from them.) We might expect, however, that coalition members are more likely to help one another than to help outsiders, and/or more likely to harm outsiders than to harm one another. The key insight is that because coalition members favor the same set of agents (each other), there is likely similarity in terms of the agents they benefit, and harm.

Our technique, then is a two step process. First, we exploit this similarity by using clustering to identify candidate coalitions. Then, we use a statistical approach to characterize these clusters, to recognize actual coalitions.

Clustering in Benefit Space

We define the *benefit space* as a high-dimensional space reflecting the degree of benefit rendered to each agent in the system. Specifically, given N total entities in the system, the benefit space $\mathcal{B}$ is a space $\mathbb{R}^N$, where the value in each dimension β_i represents an amount of net benefit (i.e., total benefit minus total harm) to entity i. Positive values represent positive benefit, while negative values represent net harm. It is clear from this definition that benefit must be measurable in some terms; this may be as simple, however, as counting positive actions and negative actions.[1]

[1]In a particular scenario, it is possible that there are multiple, distinct aspects of benefit/harm that can't easily be composed into a single measure. For example, in a battlefield/game scenario, shooting at an agent might be an act of harm, while healing an agent would be of benefit. Combining these two into a single meaningful measure of 'net benefit' might be difficult. In such a case, each such measure would be a separate dimension, meaning multiple dimensions for each agent. In this paper, however, a single measure was used.

Each entity maps to a point in the benefit space, according to the amount of (observable) net benefit it has rendered to each entity in the system. Thus, a given agent a can be represented by the vector:

$$\mathbf{b}(a) \equiv (\beta_1(a), \beta_2(a), ...\beta_N(a)) \quad (1)$$

Members of a coalition are likely to be similar, in terms of the sets of agents that they benefit, and the sets that they harm. Thus, we would expect them to be close in this benefit space (even if they don't interact directly at all). Here, we have used a simple Euclidian distance as our dissimilarity measure, where the distance between a and b is:

$$d_{a,b} = \sqrt{(\beta_1(b) - \beta_1(a))^2 + \cdots + (\beta_N(b) - \beta_N(a))^2} \quad (2)$$

With this, a standard clustering algorithm is applied to find sets of agents that are similar in benefit space. Here, we have used simple k-means clustering. This results in a partitioning of the population P into a set of clusters $\{C_1, C_2, ..., C_n\}$. As noted above, in a dynamic environment, the interactions between any pair of agents is unpredictable (e.g., coalition partners may harm one another); this results in noise. Our results (presented later in the paper) show, however, that across all dimensions, sufficient signal can be found to isolate coalitions.

Characterizing Clusters

While we would expect members of coalition to be similar in benefit space, similarity does not necessarily imply that a set of agents is a coalition. Considering a marketplace scenario, for example, buyers who favor a particular set of sellers may be close in benefit space, but may not be colluding—instead, they may simply share similar tastes, or have found the same set of reliable sellers. Moreover, a clustering algorithm will provide clusters as output, whether or not a coalition was actually present. Thus, we consider the clusters found to be *candidate coalitions*; in our second step, candidates are characterized to detect coalitions.

Our technique to identify coalitions is again based on the notion that coalition members are more likely to benefit one another than to benefit outsiders (or more likely to harm outsiders than each other). From this principle, one approach might be to compare the amount of pairwise net benefit flowing from agent to agent within a cluster, to the amount flowing from agents inside the cluster to agents outside the cluster. One might expect that if a cluster contains a coalition, the benefit flowing within the cluster (to members) would be greater than that flowing out of the cluster (to outsiders).

We expect that this approach might work for many domains. In the trust and reputation marketplace scenario used in out experiments, however, it can be misleading. Consider a group of agents S (who both buy and sell) who are not in a coalition, but rather are just excellent sellers. Because they are good sellers, they earn strong reputations, which in turn makes them more popular. Because they are popular, the agents in S are often selected by buyers, *including the other agents in* S. Because the agents in S often buy from others in S (i.e., they are benefiting the same agents), they may wind up in the same cluster together. Because the agents

in S buy from one another in preference to sellers outside S, the benefit flow within the cluster containing S would be greater than the flow out of S. According to the policy outlined above, S may be erroneously be viewed as a coalition.

Instead, we take an approach that is related, but does not suffer from this problem. We would expect a coalition of agents to benefit *each other* more than *outsiders* would favor them. Consider, then, a coalition T. We would expect the benefit flowing from members of T to other members in T, to be greater than the flow of benefit from non-members into T. (Note the contrast with S: agents in S favor one another because of high reputation, but agents outside S also favor them for the same reason.)

We apply this to characterize candidate coalitions. First, we compute the benefit flowing from agents in a candidate cluster C, to other agents within C. Let m be the number of agents in C. In our representation, benefit is directed (i.e., $\beta_a(b)$ might not equal $\beta_b(a)$), so there are m^2 'relationships' between agents in C. The average (per relationship) benefit within C, then, is

$$\bar{\beta}_C = \frac{\sum_{i \in C} \sum_{j \in C} \beta_j(i)}{m^2} \quad (3)$$

To know whether the computed value is abnormally high, we need a benchmark to which to compare it. For this, we take random samples of agents (drawn from the entire population P, including agents both within and outside of C) of size m (the same size as C). For each sample D, we compute the benefit flowing from the agents in D, to agents in C:

$$\bar{\beta}_D = \frac{\sum_{i \in D} \sum_{j \in C} \beta_j(i)}{m^2} \quad (4)$$

Performing this computation for a large number of samples (here, we use 100), we estimate the mean and standard deviation for the amount of benefit flowing from any random selection of agents to members of C.

With this information, we can estimate the probability of obtaining a measure as high as $\bar{\beta}_C$ by chance, using the normal distribution. If this probability is too low, we conclude that members of C are benefitting each other far more than outsiders are, and that C thus contains a coalition.[2]

The threshold probability below which clusters are considered to contain coalitions (α) is a parameter: lower values reduce the risk of false positives, while increasing the risk of false negatives. In our tests, we used $\alpha = 0.001$.

When a cluster has been identified as containing a coalition, we label all agents in that cluster as coalition members.

Experimental Scenario

The TREET testbed (Kerr and Cohen 2010a) was used to validate our technique. TREET provides a rich, flexible simulated marketplace environment for experimentation with trust and reputation technologies; the reader is referred to that paper for details. Agents within TREET are buyers and sellers of products; while user accounts can be observed, the owners of the accounts (and their communications with other agents) are hidden, allowing experimentation with collusion. In TREET, each agent is assigned a set of products that they can produce. Each turn, each agent is assigned a random set of products that they need to purchase, requiring buyers to interact with a variety of sellers.

In our simulations, the marketplace was populated with honest agents who make use of the Beta Reputation System (Jøsang and Ismail 2002) in order to find trustworthy sellers and avoid unreliable ones. After each sale, the buying agent rates the seller's trustworthiness. We inserted into this population, coalitions of various sizes making use of either bad-mouthing or ballot-stuffing to improve their competitive position. (The coalition agents were otherwise honest, fulfilling all sales diligently.) The total population size in each run was 400 agents. At most, one coalition was present in any given trial.

The simulator provides us with labelled data: each agent is known to be either part of a coalition, or not. We remove these class labels before applying our technique. Afterward, we compute the accuracy of our technique by comparing our output to the actual classes of the agents.

Results

In the first set of tests, coalition members were engaged in bad-mouthing. Agents did not make additional purchases in order to bad-mouth competitors, however; instead, coalition members bought the products that they actual needed, but if they purchased from a competitor, they gave a negative review with probability 0.5. (Agents do not bad-mouth on every transaction, to avoid being obviously engaged in the tactic.) When the same product was available from both coalition members and non-members, members had no preference to buy from one or the other.

In our environment, a number of measures of benefit and harm can be identified, e.g., number of purchases, dollar value of purchase, number of positive reviews, average review score, etc. Here, we use only one of the available measures: the net sum of the review values given, weighted by the dollar value of the transaction. This captures both benefit (positive reviews) and harm (negative reviews), as well as the importance of transactions.

A variety of coalition sizes were tested, as reflected in the figures. For each coalition size, 5 trials were run; the figures reported reflect the aggregate results across trials. (There was one exception: to be very confident that the technique handles the case where there are no coalitions present, 35 trials were run with zero coalition members.)

Figure 1 depicts the clustering performance when coalitions engaged in bad-mouthing. Here, we are not necessarily concerned if coalition members are placed in a single cluster, or split amongst multiple clusters. (The same holds for non-members.) Rather, our primary concern is that coalition members and non-members are partitioned into separate clusters, so that each cluster can be characterized and labelled. Thus, we use purity, and the true class of each

[2] We only apply this technique to clusters no larger than half the size of the population. Clusters larger than this are ignored. There are two key reasons for this. First, drawing repeated random samples of a size approaching the size of the population is problematic. Second, coalitions consisting of the majority of the members in a population are likely to be poorly-kept secrets, and need no special detection methods.

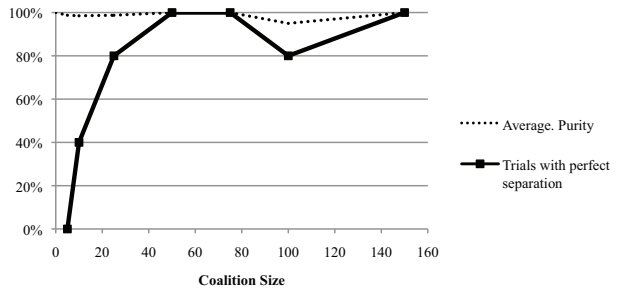

Figure 1: Clustering performance against bad-mouthing.

agent (i.e., the real identity as either coalition member or non-member) to evaluate clustering. Briefly, to calculate the purity of a cluster, we determine which class has the highest number of members in the cluster, then compute the portion of the cluster represented by that class. A purity score of 1.0 indicates that a cluster consists entirely of a single class.

In Figure 1, purity values diverge little from 100%, but this can be deceptive—trivially placing every agent into a single cluster results in a high purity for small coalitions. (E.g., for 5 colluders and 395 others in one cluster, purity $= 395/400 = 0.9875$.) In any given trial, we ideally want perfect separation (i.e., purity of 1.0). Thus, for clarity, Figure 1 also shows the percentage (of five trials run for each coalition size) that resulted in perfect separation.

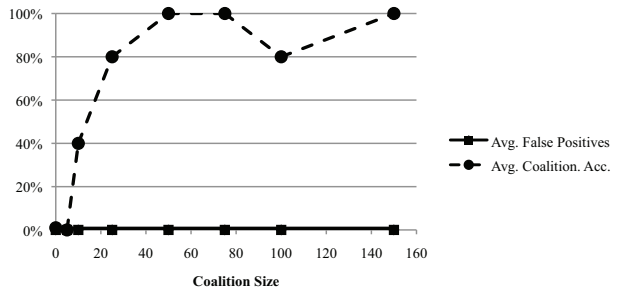

Figure 2: Detection performance against bad-mouthing.

Rather than giving overall measures of detection accuracy (which can be misleadingly high, for the same reason as purity), Figure 2 shows both the proportion of individual colluders that were accurately identified ('Avg. Coalition Acc.'), as well as the proportion of non-colluders that were wrongly labelled as colluders ('Avg. False Positives'). Several points should be noted. First, overall detection of coalition members was quite strong, except where coalition sizes were small. In fact, the characterization component of our algorithm was extremely accurate—failures to detect collusions were entirely due to failures of the clustering algorithm in partitioning coalition members.[3] Second, the algorithm is extremely resistant to false-positives: in no case was a non-member identified as a coalition member. This is particularly noteworthy where the number of colluders is zero; the technique successfully copes with the case where no coalition is present.

One might wonder whether, because all sellers were acting honestly, if bad-mouthing agents stand out because they were the only ones giving negative reviews. This is not the case. First, the benefit measure used did not reflect negative reviews, only total net benefit. More importantly, we also investigated the case where only ballot-stuffing is used, and thus no negative reviews are given. (Here, in addition to their normal purchases, coalition members engage in an extra 25% ballot-stuffing transactions.) The results, depicted in Figures 3 and 4, show similarly strong performance. One

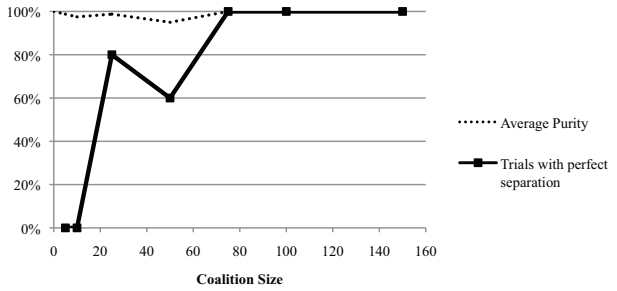

Figure 3: Clustering performance against ballot-stuffing.

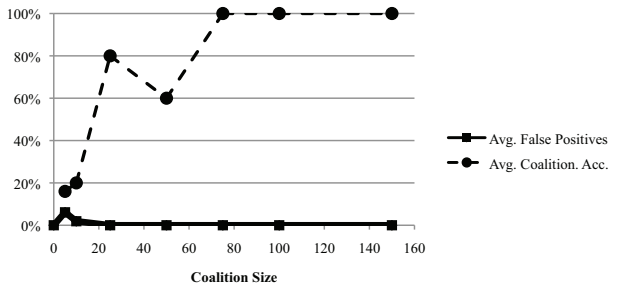

Figure 4: Detection performance against ballot-stuffing.

noteworthy difference from the bad-mouthing case is the presence of a small number of false positives in two trials. In these cases, the clustering algorithm constructed a cluster containing both the coalition members, as well as a number of outsiders. Because of the presence of the coalition members, on average the cluster members benefited each other substantially—the cluster was identified as 'containing a coalition'. As such, the entire cluster was labelled as a coalition—the non-members in the cluster were wrongly identified. This situation might be avoided by tuning the α parameter.

Discussion and Future Work

It is worth noting that our algorithm detects groups of agents that are providing more net benefit to each other than other

[3]The attentive reader may notice that detection accuracy averages all fall on multiples of 20%. This is because, as noted, a clustering failure results in none of the colluders being detected on one of the five trials.

members provide. We might call this a *de facto* coalition. It may arise because the group is an actual coalition, intentionally acting in concert. Such a situation might possibly arise, however, due to other circumstances: for example, a group of agents may have closely aligned needs and capabilities, and favor each other for this reason. We make no attempt to distinguish between these cases, and the importance of this issue is likely to be scenario-specific. We do note, however, the close correspondence between actual coalitions and detected coalitions in our results.

Conversely, an actual coalition (intentionally coordinating their efforts) might be ineffective—perhaps they act too little to benefit one another, or they have problems of coordination. Such a group would not be detected as a coalition. Is this a problem? The answer may depend on the scenario. This may be an important issue, given our results. Detection accuracy was poor for small coalitions; it may be that the total activity of these coalitions (and the benefit derived from the members) was too little to be detected.

In this first proposal for detecting coalitions, we have achieved a noteworthy degree of success, despite employing simple tools. For example, as noted early in the paper, identifying appropriate measures of benefit for a given scenario requires domain knowledge, and is non-trivial. We used only one of many measures available in our scenario; we intend to explore a variety of such measures. The use of additional features may, for example, aid in the detection of small coalitions. Similarly, we used Euclidean distance for similarity. It may be the case that the nature of interactions is more important than the quantity of interactions. To investigate this, we intend to explore the use of polar coordinates, cosine similarity, etc. Further, we used only simple k-means clustering. Given the central importance of clustering accuracy to our technique, we intend to investigate the usefulness of more sophisticated algorithms. This may be especially important when we consider cases where there may be many coalitions in the environment.

Beyond these potential improvements, we intend to explore the enhancement of this technique to handle a wider range of situations: overlapping coalitions, changing membership, etc. We also intend to apply this technique to other domains, such as those noted in the introduction.

Conclusion

In this paper, we have presented a technique that allows coalitions to be detected and their members identified. Because it is based on the concept of benefit rather than on domain-specific features, and because it requires no knowledge of the plans that may be used by coalitions, we believe it to be applicable to a wide variety of domains. The effectiveness of the technique was demonstrated using trust and reputation systems, an area where coalitions are especially problematic. The method was shown to provide strong detection performance, while at the same time being resistant to false positives, especially important where one might take corrective or punitive actions against suspected colluders.

We believe that this paper represents an important step towards addressing the challenges posed by coalitions, particularly for domains where such cooperation is problematic.

References

Burke, R.; Mobasher, B.; Williams, C.; and Bhaumik, R. 2006. Classification features for attack detection in collaborative recommender systems. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 547. ACM.

Dellarocas, C. 2000. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the 2nd ACM conference on Electronic commerce*, 150–157. ACM.

Girvan, M., and Newman, M. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99(12):7821.

Jøsang, A., and Ismail, R. 2002. The beta reputation system. 15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy.

Jurca, R., and Faltings, B. 2007. Collusion-resistant, incentive-compatible feedback payments. In *EC '07: Proceedings of the 8th ACM conference on Electronic commerce*, 200–209. New York, NY, USA: ACM.

Katz, J. M. 2005. Hold 'em, fold 'em, cheat 'em. *Slate.com* (http://www.slate.com/id/2112213).

Kerr, R., and Cohen, R. 2010a. TREET: The Trust and Reputation Experimentation and Evaluation Testbed. *Electronic Commerce Research* 10(3):271–290.

Kerr, R., and Cohen, R. 2010b. Trust as a tradable commodity: A foundation for safe electronic marketplaces. *Computational Intelligence* 26(2):160–182.

Mehta, B., and Nejdl, W. 2009. Unsupervised strategies for shilling detection and robust collaborative filtering. *User Modeling and User-Adapted Interaction* 19(1-2):65–97.

Newman, M., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical review E* 69(2):26113.

Osborne, M., and Rubinstein, A. 1994. *A course in game theory*. The MIT press.

Shehory, O., and Kraus, S. 1999. Feasible formation of coalitions among autonomous agents in nonsuperadditive environments. *Computational Intelligence* 15(3):218–251.

Sukthankar, G., and Sycara, K. 2006. Robust recognition of physical team behaviors using spatio-temporal models. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 638–645. New York, NY, USA: ACM.

Sukthankar, G., and Sycara, K. 2007. Policy recognition for multi-player tactical scenarios. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, 1–8. New York: ACM.

Tambe, M. 1996. Tracking dynamic team activity. In *Proceedings of the National Conference on Artificial Intelligence*, 80–87.

Teacy, W. T.; Patel, J.; Jennings, N. R.; and Luck, M. 2006. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems* 12(2):183–198.