Possible and Necessary Winner Problem in Social Polls (Extended Abstract)

Serge Gaspers
UNSW and NICTA
Sydney, Australia
sergeg@cse.unsw.edu.au

Victor Naroditskiy
University of Southampton
Southampton, UK
vn@ecs.soton.ac.uk

Nina Narodytska
NICTA and UNSW
Sydney, Australia
nina.narodytska@nicta.com.au

Toby Walsh
NICTA and UNSW
Sydney, Australia
toby.walsh@nicta.com.au

ABSTRACT

Social networks are increasingly being used to conduct polls. We introduce a simple model of such social polling. We suppose agents vote sequentially, but the order in which agents choose to vote is not necessarily fixed. We also suppose that an agent's vote is influenced by the votes of their friends who have already voted. Despite its simplicity, this model provides useful insights into a number of areas including social polling, sequential voting, and manipulation. We prove that the number of candidates and the network structure affect the computational complexity of computing which candidate necessarily or possibly can win in such a social poll. For social networks with bounded treewidth and a bounded number of candidates, we provide polynomial algorithms for both problems. In other cases, we prove that computing which candidates necessarily or possibly win are computationally intractable.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Economics, Theory

Keywords

Social polls, social choice, possible winner, necessary winner

1. INTRODUCTION

A fundamental issue with voting is that agents may vote strategically. Results like those of Gibbard-Satterthwaite demonstrate that, under modest assumptions, strategic voting is likely to be possible [7, 9]. However, such results do not tell us how to vote strategically. A large body of work in computational social choice considers how we compute such strategic votes [5, 4]. Typically such work starts from some strong assumptions. For example, it is typically assumed that the manipulators have complete information about the other votes. The argument given for this assumption is that computing a strategic vote will only be computationally harder with incomplete information. In practice, of course, we often only have partial or probabilistic information [10, 2]. It is also typically assumed that manipulators will vote in any way that achieves their ends. However, in practice, agents may be concerned about peer pressure and may not want to deviate too far from either their true vote or that of their peers [8]. Bikhhardani et al. [1] identified several factors that limit strategic voting by an individual agent such as sanctions on deviation, and conformity of preferences. A third strong assumption is either that all voting happens simultaneously or that the manipulators get to vote after all the other agents. Again, in practice, this is often not the case.

These issues all come to a head in *social polling*. This is a context in which voting meets social networks. Startups like Quipol and GoPollGo use social networks to track public opinions. Such polls are often not anonymous. We can see how our friends have voted and this may influence how we vote. By their very nature, such polls also happen over time. The order in which agents vote can therefore be important. The structure of social networks is also important. For example, a distinctive feature of social networks is the small world property which allows members of these communities to share information in a highly efficient and low cost manner. A rumor started in the Twitter network reaches about 90% of the network in just 8 rounds of communication [3]. In a similar way, one member of a social network can quickly create and publicize a poll among a large group of agents starting from his friends. The massive size of social networks, like Facebook, Twitter and Google+, gives statistically significant polls.

To study social polling, we set up a general model that captures several important features of voting within a social network. First, our model uses the structure of the social network. How an agent votes depends on how their friends vote. Second, our model supposes agents vote sequentially and the order in which they vote is not under their control. For example, when you vote may depend on when one of your friends chooses to invite you to vote. Third, our model supposes that agents are influenced by their friends. In fact, an agent's vote is some function of their true preferences and of the preferences revealed by the votes of their friends that have already voted. We can obtain different instances of our model by choosing different functions.

To study this model, we consider a particular instance that captures some of the features of a Doodle poll. More precisely, each agent has a set of k preferred candidates and is indifferent about other candidates. Among these k preferred candidates, one candidate is her top choice. If a particular candidate among her k pre-

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013),* Ito, Jonker, Gini, and Shehory (eds.), May 6–10, 2013, Saint Paul, Minnesota, USA.

ferred candidates has a majority amongst her friends that have already voted, then she mimics their choice. Otherwise, she votes for her top choice. Note that any computational lower bounds derived for this particular instance also hold for the general model.

Even though this instance of the model is simple and lacks some of the subtleties of social influence in practice, it nevertheless provides some valuable insights. For example, we prove that it is computationally hard to determine if a given candidate has necessarily won a social poll, irrespective of how the remaining agents vote. We also show that this intractability holds even if the social graph has a simple structure like a disjoint union of paths. Of course, in practice social influence is much more complex and subtle. In addition, social graphs often have much a richer structure than simple paths. Finally, agents in general do not know precisely how all the other agents will vote. However, all these issues will only increase the computational complexity of reasoning about a social poll.

We focus here on computing the possible and necessary winners of the social poll. A candidate is a possible winner if there exists a voting order such that this candidate is a plurality winner over the cast votes. Similarly, a candidate is a necessary winner if he is a plurality winner over the cast votes for each voting order. The possible and necessary winner problems are interesting in their own right. In addition, they provide insight into several related and interesting problems. For example, they are related to the control problem in which the chair chooses an order of participation for the agents that favors a particular outcome. In particular, the chair can control the result of the election in this way if and only if their desired candidate is a possible winner.

2. PROBLEM STATEMENT

We consider a scenario where each agent votes for exactly one candidate. We are given a social network graph $G = (V, E)$ whose n vertices are the agents $x_1, \ldots, x_n$, a set $\mathcal{C} = \{c_1, \ldots, c_m\}$ of m candidates, a distinguished candidate $c^* \in \mathcal{C}$, and a choice function h, which for every agent x_i, every subset $S \subseteq N_G(x_i)$ of its neighbors in G, and every vote of an agent in S, assigns the candidate that x_i votes for. Each agent casts exactly one vote according to the following model. For a given *voting order* $\pi = (x_{\pi(1)}, \ldots, x_{\pi(n)})$, let S_i denote the set $\{x_j : \pi^{-1}(j) < \pi^{-1}(i)\} \cap N_G(x_i)$, i.e., the neighbors of x_i that vote before x_i. Each agent x_i votes for the candidate that the choice function h assigns for the given candidate x_i, the subset S_i and the votes of the agents in S_i. The *score* of a candidate c is the number of agents that vote c in the voting order π. A candidate $c \in \mathcal{C}$ is a (co-)winner in the voting order π if no other candidate has higher score than c. A candidate is a *possible winner* if there exists a voting order where c is a winner. A candidate is a *necessary winner* if for every voting order, c is a winner.

Refined model. We introduce a particular instance of the choice function h. This is defined via two *preference functions* $p_1 : V \to \mathcal{C}$ and $P : V \to 2^{\mathcal{C}}$. Each agent $x \in V$ has a set $P(x) \subseteq \mathcal{C}$ of k preferred candidates, where $k > 1$ is a constant. Among the preferred candidates, one candidate $p_1(x) \in P(x)$ is the top preferred candidate. Let x be an agent and S be the subset of $N_G(x)$ that voted before x. If there exists a candidate $c \in P(x)$ such that more than half of the agents from S voted for c, then x votes for c. Otherwise, x votes for $p_1(x)$.

The *unweighted possible (necessary) winner* problem is to determine whether c^* is a possible (necessary) winner. The *weighted possible (necessary) winner* problems are defined similarly, except that integer weights are associated with agents and the score of a candidate is the sum of the weights of the agents that vote for him.

3. OVERVIEW OF RESULTS

We show that the computational complexity of the possible and necessary winner problem depends on the structure of the underlying social graph and the number of candidates. In particular, we prove that if the underlying social graph has bounded treewidth and the number of candidates is bounded then the unweighted possible and necessary winner problems can be solved in polynomial time. The degree of the polynomial bounding the running time of this algorithm is a function of the number of candidates and the treewidth of the social network graph. We give evidence that this cannot be avoided. For arbitrary social network graphs and a bounded number of candidates, the weighted possible winner problem is NP-complete, while the weighted necessary winner problem is polynomial. If we relax the restriction on the treewidth, all problems become computationally intractable. Finally, we investigate these problems under the assumptions that the number of candidates is unbounded and the social graph is a disjoint union of paths. We show that the unweighted possible winner problem is hard even if the length of each path is at most one. By contrast, the necessary winner problem is polynomial under the assumption that the number of candidates is unbounded and the underlying social graph has bounded treewidth. Our results also demonstrate that the possible winner problem is inherently computationally harder than the necessary winner problem.

We refer the reader to [6] for a full version of the paper.

Acknowledgments NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council. Serge Gaspers acknowledges support from the Australian Research Council (grant DE120101761).

4. REFERENCES

[1] S. Bikhchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy*, 100(5):992–1026, 1992.

[2] V. Conitzer, T. Walsh, and L. Xia. Dominating manipulations in voting wih partial information. In *Proceedings of AAAI 2011*. AAAI Press, 2011.

[3] B. Doerr, M. Fouz, and T. Friedrich. Why rumors spread so quickly in social networks. *Communications of the ACM*, 55(6):70–75, 2012.

[4] P. Faliszewski, E. Hemaspaandra, and L.A. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

[5] P. Faliszewski and A.D. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[6] S. Gaspers, V. Naroditskiy, N. Narodytska, and T. Walsh. Possible and necessary winner problem in social polls. Technical Report 1302.1669, arXiv, 2013.

[7] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–601, 1973.

[8] S. Obraztsova and E. Elkind. Optimal manipulation of voting rules. In *Proceedings of AAMAS 2012*, 2012.

[9] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–216, 1975.

[10] T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proceedings of AAAI 2007*, pages 3–8. AAAI, 2007.