Computational Complexity of Fundamental Problems in Social Choice Theory

(Doctoral Consortium)

Palash Dey
Supervisors: Prof. Y. Narahari and Prof. Arnab Bhattacharyya
Department of Computer Science and Automation
Indian Institute of Science
Bangalore, India
palash@csa.iisc.ernet.in

ABSTRACT

Computational social choice (comsoc) theory is currently an important area of research in computer science and more specifically in AI. The field started with the pioneering work of Bartholdi et al. in 1989 where they explored the the possibility of using computational intractability as a barrier against manipulation. Following that, a vast amount of research explored computational complexity of various problems in the context of social choice theory. We, in this thesis, study some of the fundamental problems in this domain. Manipulation of voting rules is a well known phenomena is social choice theory. Till date, researchers have studied a plenty of ways to make manipulation either impossible or computationally intractable. Yet, there are not a single satisfactory solution to prevent manipulation. In such a scenario where prevention fails even after considerable research effort of more than four decades, a natural research direction is to explore detection of manipulation. This is precisely the goal of one of our works [2] in this thesis. Another very well studied problem in comsoc is the possible winner problem. There exist quite a large literature studying computational aspects of this problem for various commonly used voting rules. Researchers also studied parameterized complexity of this problem and a fixed parameter tractability result with parameter being the number of candidates follows very easily by reducing it to an integer program. However, the kernelization complexity of this problem is surprisingly a big open problem in comsoc. In one of our works [3], we resolve this open question for many commonly used voting rules. Arguably the most fundamental problem in comsoc is winner determination - given a set of votes and a voting rule, who wins the election? In exit polls and many other applications, people often tries to predict the winner of an upcoming election by sampling a few votes and running the election on those sampled votes. We study the sample complexity of winner prediction for many common voting rules and show upper and lower bounds on the sample complexity [1]. Moreover, the upper and lower bounds match for most of the voting rules.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity;
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Algorithms, Theory

Keywords

voting theory; algorithms; theoretical computer science

1 Introduction

In many real life situations, including multiagent systems, agents often need to agree upon a common alternative even if they have different preferences over the available alternatives. Voting is one of the most suitable tools in these scenarios. Common and classic applications of voting in artificial intelligence include collaborative filtering, planning among multiple automated agents, web ranking, rank aggregation, etc. There are many important computational problems in the context of voting theory. Our goal is to study the time, space, and sample complexity of fundamental problems in voting. The typical setting may be described as follows. An election consists of a set of voters, a set of alternatives, and a voting rule. The vote of any voter can be thought of as a ranking (more precisely, a complete order) of the set of alternatives. A voting profile is just a collection of votes of all voters. Finally, a voting rule is a function that takes a voting profile as input and outputs an alternative, which is regarded as the winner of the election. The winner is also called the outcome of the election.

2 Results

2.1 Complexity of the Detection of Possible Manipulation Problem

Following the impossibility result of the G-S theorem, there have been attempts to find workarounds to manipulation. Economists propose to restrict the domain of the voting rule as a solution. They showed that, if we restrict ourselves to the domain of single peaked preferences then the median voting rule is unanimous, non-dictatorial, and non-manipulable irrespective of the number of alternatives we have in the election. However, domain restriction is not a satisfactory solution since the social planner can never be sure about the domain of the voters' preferences.

On the other hand, computer scientists showed that for many common voting rules, the computational problem of manipulation is NP-complete. So, for many common voting rules, to manipulate an election, manipulators have to solve an intractable problem. However, recently researchers showed that any voting rule that is neutral and sufficiently far from being dictatorial (all common voting rules satisfy these properties) are easily manipulable on average by reporting a preference picked uniformly at random from the set of all preferences. This shows weakness of the intractability barrier. In a situation where prevention of manipulation looks evasive after few decades of research, we turn our attention to manipulation detection.

A subtle issue in this direction is that the "true preference" of a voter is known to her only. In particular, we cannot presume to know the real preferences of the voters. In the absence of this benchmark, we can only talk about the possibility of manipulation. We show that finding possible manipulators in an election is polynomial time solvable for many common voting rules although it is NP-complete for a few common voting rules [2]. The novelty of this work lies in initiating a new research direction namely detection of manipulation. We showed that there are voting rules, for example the Borda voting rule, for which Detecting Possible Manipulation is easy although manipulation itself is NP-complete.

2.2 Kernelization Complexity of the Possible Winner Problem

There are situations where voters do not provide complete order over the alternatives as their vote. They only provide partial orders over the alternatives. There are several situations that lead to such a scenario; for example, it may be a tedious job for the voters to provide a complete order when the number of alternatives is not small or the voters may not have enough information about the alternatives to provide the complete order. Hence, elections where the votes are partial orders over the alternatives are seen in many applications.

In this context, one pertinent computational problem is the Possible Winner problem which asks if there is a way to extend the partial votes to linear ones that make a specified alternative winner. This problem has been shown to be NP-complete for many common voting rules. Subsequently, researchers developed algorithms showing that the Possible Winner problem is fixed parameter tractable with total number of alternatives as the parameter for many common voting rules. A problem is called fixed parameter tractable with respect to a parameter k if it has an algorithm running in time $f(k) * poly(n)$ where n is the size of the input. The motivation for fixed parameter tractability is that, if the parameter value is *small*, we have a tractable algorithm for the problem. There is another notion called kernelization which goes hand in hand with fixed parameter tractability and the corresponding kernelization question for the Possible Winner problem is still open. In this work [3], we show that the Possible Winner problem does not admit a polynomial kernel when parameterized by the total number of alternatives.

Further, we studied an important and well studied special case of this problem, namely, Coalitional Manipulation. Here the partial votes are either complete orders over the alternatives or empty. The voters corresponding to empty votes are called manipulators. The Coalitional Manipulation problem is known to be NP-complete for many common voting rules even when we have constant number of manipulators. We showed that the Coalitional Manipulation problem for many common voting rules admit a polynomial kernel when parameterized by total number of alternatives.

In summary, our results show that even though both the Possible Winner and the Coalitional Manipulation problems are NP-complete, the Coalitional Manipulation problem is significantly easier to deal with than the Possible Winner problem.

2.3 Sample Complexity for Winner Prediction in Elections

Predicting the winner of an election is a favorite problem both for news media pundits and computational social choice theorists. Since it is often infeasible to elicit the preferences of all the voters in a typical prediction scenario, a common algorithm used for winner prediction is to run the election on a small sample of randomly chosen votes and output the winner as the prediction. We analyze the performance of this algorithm for many common voting rules.

More formally, we introduce the (ϵ, δ)-*winner determination problem* [1], where given an election on n voters and m candidates in which the margin of victory is at least ϵn votes, the goal is to determine the winner with probability at least $1 - \delta$. The margin of victory of an election is the smallest number of votes that need to be modified in order to change the election winner. We show interesting lower and upper bounds on the number of samples needed to solve the (ϵ, δ)-winner determination problem for many common voting rules, including scoring rules, approval, maximin, Copeland, Bucklin, plurality with runoff, and single transferable vote. Moreover, the lower and upper bounds match for many common voting rules in a wide range of practically appealing scenarios.

3 Conclusions

We studied some of the fundamental problems in comsoc in this thesis. Other than the problems discussed above, we are currently working on a couple of other interesting problems as well. One of those problems is bribery which is a very well studied problem again in comsoc. However, there are very interesting works going on currently in bribery, specially parameterized complexity of bribery and there are quite a few unresolved questions.

Acknowledgments

We acknowledge Dr. Neeldhara Misra for all the fruitful discussion and constant encouragements. We are also indebted to all the members of the Game Theory lab for providing us such an excellent research environment.

REFERENCES

[1] P. Dey and A. Bhattacharyya. Sample complexity for winner prediction in elections. In *Proceeding of the 14th International Conference on Autonomous Systems and Multiagent Systems (AAMAS-15)*, 2015.

[2] P. Dey, N. Misra, and Y. Narahari. Detecting possible manipulators in elections. In *Proceeding of the 14th International Conference on Autonomous Systems and Multiagent Systems (AAMAS-15)*, 2015.

[3] P. Dey, N. Misra, and Y. Narahari. Kernelization complexity of possible winner and coalitional manipulation problems in voting. In *Proceeding of the 14th International Conference on Autonomous Systems and Multiagent Systems (AAMAS-15)*, 2015.