

Abstraction Using Analysis of Subgames

Anjon Basak

University of Texas at El Paso
 500 W University Ave, El Paso, TX 79968
 915-731-3083, abasak@miners.utep.edu

Abstract

Normal form games are one of the most familiar representations for modeling interactions among multiple agent. However, modeling many realistic interactions between agents results in games that are extremely large. In these cases computing standard solutions like Nash equilibrium may be intractable. To overcome this issue the idea of abstraction has been investigated, most prominently in research on computer Poker. Solving a game using abstraction requires using some method to simplify the game before it is analyzed. We study a new variation for solving normal form games using abstraction that is based on finding and solving suitable sub games. We compare this method with several variations of a common type of abstraction based on clustering similar strategies.

Introduction

Game theory models interactions among self-interested rational agents, where each agent tries to maximize its utility. The most basic representation of a game is the well-known normal form (NFG). One of the key challenges in computational game theory is finding effective ways of analyzing extremely large games. Many real-world situations where we would like to apply game theory naturally have very large strategy spaces or other complexities. However, finding Nash equilibria is known to be a computationally hard problem.

To analyze games that are beyond the limits of standard solution algorithms, an increasingly common approach is to apply some form of automated abstraction to simplify the game. The simplified game is then analyzed using an available solver, and the solution is somehow mapped back into the original game. If the simplified game is able to retain the key strategic features of the original game, then in principle the solution of the simpler game may be a reasonable approximation of the solution to the original game. This general approach has been very successful in developing computer poker agents, and most of the successful players in the annual competition in computer poker over the past years have used some variation of this idea (e.g. (Gilpin and Sandholm 2006)).

I study a new variation for solving normal form games¹

Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹<http://www.cs.utep.edu/kiekintveld/students/basak/paper.pdf>

. Even for this setting abstraction is not completely understood, and there has been less work in this area (with some exceptions, such as the decision-theoretic clustering methods proposed by (Bard et al. 2015)). Here I focus on the simpler setting of normal form games and particularly games with a structure called AIOS (Approximately Identical Outside Subgames). This is a similar structure to ALAGIU (All Lower Actions Give Identical Utility) (Conitzer and Sandholm 2006). The key idea is to create clusters of strategies for both players that form subgames. Within a subgame, the strategies and payoffs can vary arbitrarily. However, outside of the subgame, the strategies for each player should be as similar as possible in the payoffs for playing against any opponent strategy not in the cluster.

We are motivated by the recursive game reduction technique introduced by Conitzer et al. (Conitzer and Sandholm 2006). This idea leads to a different type of clustering abstraction where clusters form subgames that are solved separately to create the abstracted game. We also introduce a variation of this technique that can improve solution quality by iteratively modifying the subgames to account for abstraction error. In cases where there are differences outside of the subgames, simply composing the solutions of the subgames may not be an equilibrium of the original game. This is because the solution may occasionally result in play in quadrants of the game that are *not* one of the subgames solved explicitly, which results in error because the payoffs are not exactly the same for all strategies. The iterative solution technique attempts to account for this error. First, we calculate for each strategy in a cluster the expected payoff for this strategy outside of the subgame against the current opponent solution. Then, we modify the subgames using the error offset, and resolve the subgames. This results in a sequence of modified solutions that account for the differences in payoffs outside of the subgames from the previous iteration. We call this method ISASC for Iterative Subgame Abstraction and Solution Concept.

When solving an abstracted game, it is not clear that the best analysis is simply to find a Nash equilibrium of the abstracted game, since this may not be an equilibrium of the original game. This is the idea for our next contribution, Minimum Epsilon Bounded equilibrium (MEB) which represents an improved version of pure strategy Nash Equilibrium (PSNE) that considers additional information about the

abstraction in the analysis. Instead of considering deviations to clusters of actions (and the average payoff of the cluster), when considering deviations we use the maximum expected payoff for any of the actions in the original game. This allows for a better estimate of how close the outcome will be to an equilibrium in the original game, not the abstracted game.

Methodology

The main criteria we use to evaluate the contributions is how well the solution methods are able to approximate a Nash equilibrium in the original, unabstracted game, as measured by ϵ (epsilon). We generated test games based on the AIOS model, but with varying levels of δ , which specifies how much variation in the payoffs is allowed outside of the subgames. That is, we vary the maximum difference between the payoffs outside of the subgame for two strategies in the same cluster. We generated the payoffs outside the subgames in such a way that in every cluster the maximum payoff difference between the payoffs for the actions is δ for all actions of the opponent that are not part of the subgame. We compared the ISASC with cluster-based (e.g. k-means) abstraction with different solution concepts: PSNE, MEB and Quantal Response Equilibrium (QRE).

Experiments

The experiments show that ISASC improves the solution quality as the number of iterations increases, when there is noise outside the subgames. We stop iterating when the strategy does not change from one iteration to the next.

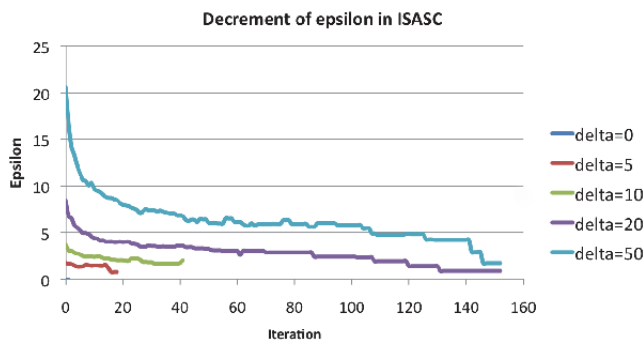


Figure 1: Improvement in solution quality with more iterations

Figure 1 shows how the solution quality (measured by the ϵ in the original game) improves with additional iterations. The next experiment compares the solution quality of ISASC with the clustering abstractions using k-means clusters and the three different solution methods applied to the subgames: PSNE, QRE, and MEB.

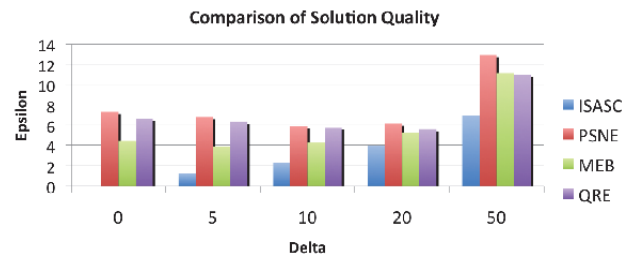


Figure 2: Graph showing the comparison of solution quality among different solution concepts

In a second sample result, Figure 2 shows the result of an experiment on the same set of games with varying levels of δ . ISASC does very well in cases with low δ , as expected. However, it continues to compare favorably to the other clustering approaches even when the values of δ are much larger. We also note that MEB generally provides better solution quality than PSNE and QRE.

Conclusion and Future Work

Solving large NFG is a fundamental problem in computational game theory, and abstraction is a promising direction for scaling up to solve the largest and most challenging games. We considered different types of abstraction. The most interesting approach we considered was based on the game reduction methods of (Conitzer and Sandholm 2006). Our experimental results showed that this approach is useful as an abstraction method, even when the structure is relaxed to a noisy approximation (which is more likely to exist in real games). We also introduced improvements to this idea including iteratively accounting for the errors outside of the subgames that further improve the solutions. Our future work involves extension of ISASC to solve games from more realistic situations. We also want to extend MEB to give a mixed strategy. We also need to analyze whether the extra computation of solving the subgames is worth instead of solving the original game.

Acknowledgement

Dr. Christopher Kiekintveld is supervising this research. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1253950, and the Army Research Office under Grant No. 26020375.

References

- Bard, N.; Nicholas, D.; Szepesvári, C.; and Bowling, M. 2015. Decision-theoretic clustering of strategies. In *AA-MAS*.
- Conitzer, V., and Sandholm, T. 2006. A technique for reducing normal-form games to compute a Nash equilibrium. In *AAMAS*, 537–544.
- Gilpin, A., and Sandholm, T. 2006. A competitive Texas hold'em poker player via automated abstraction and real-time equilibrium computation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, volume 21, 1007.