Abstraction for Solving Large Incomplete-Information Games

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University

Abstract

Most real-world games and many recreational games are games of incomplete information. Over the last dozen years, abstraction has emerged as a key enabler for solving large incomplete-information games. First, the game is abstracted to generate a smaller, abstract game that is strategically similar to the original game. Second, an approximate equilibrium is computed in the abstract game. Third, the strategy from the abstract game is mapped back to the original game.

In this paper, I will review key developments in the field. I present reasons for abstracting games, and point out the issue of abstraction pathology. I then review the practical algorithms for information abstraction and action abstraction. I then cover recent theoretical breakthroughs that beget bounds on the quality of the strategy from the abstract game, when measured in the original game. I then discuss how to reverse map the opponent's action into the abstraction if the opponent makes a move that is not in the abstraction. Finally, I discuss other topics of current and future research.

Introduction

Most real-world games (such as auctions, negotiation settings, security games, cybersecurity games, and medical games) and many recreational games are games of incomplete information. Over the last dozen years, abstraction has emerged as a key enabler for solving large incomplete-information games. For instance, it has become an essential tool in the arsenal of the leading teams working on building programs for playing poker.

At a high level, the process works in three stages, see Figure 1. First, the game is abstracted to generate a smaller, abstract game that is strategically similar to the original game. Second, the abstract game is solved for (near-)equilibrium. Third, the strategy from the abstract game is mapped back to the original game.

Reasons to abstract games

Depending on the setting, there can be different motivations for using abstraction on games:

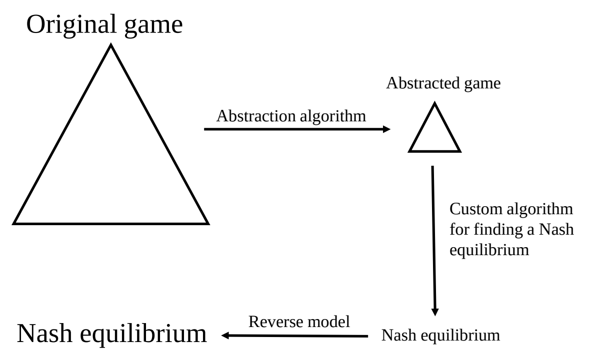

Figure 1: Framework for abstracting games.

- The original game may be too large to solve for equilibrium with today's technology. This is the case, for example, in heads-up (i.e., two-player) Texas Hold'em poker—even though the game is a two-player zero-sum game, and such games can be solved in polynomial time in the size of the game tree. The original game tree would simply be prohibitively large (Johanson 2013).
- The equilibrium solver may require a simpler game class than what the original game is. For example, it may require a game with discrete actions and/or states. Or, an equilibrium might not even be known to exist in the original game class. This has been pointed out as a motivation to abstract, for example, in computer billiards (Archibald and Shoham 2009).
- The original game is too difficult or too large to write down in full detail so abstraction is needed in modeling.
  - This is the case typically in "empirical game theory" (Wellman 2006; Wellman et al. 2005) where the game is not given explicitly but rather access to the game is by taking part in a simulation, for example a trading agents simulation.
  - All modeling of real-world settings as games can be thought of as abstraction. Thus it is important to have results that tie the abstraction coarseness to solution quality—for example, regret of the computed strategy when evaluated in the original game. Otherwise, we

could not know what our game models really have to say about reality.

What is game abstraction, really?

At its core, game abstraction amounts to assuming restrictions on the players' strategy spaces. The most common form of game abstraction is *information abstraction*, where information states are bundled together. Another common form of game abstraction is *action abstraction*, where some of the actions in the real game are assumed not to be usable. For example, if the original game has a continuum of possible actions (or a prohibitively large set of possible discrete actions) at some information sets in the game, action abstraction would create a model where typically only a small discrete set of actions is available.

Game abstraction has challenges that do not arise when using abstraction in single-agent settings

Abstraction in games is fundamentally different and more challenging than abstraction in single-agent settings (such as Markov Decision Processes). In games, unlike in single-agent settings, abstraction can be nonmonotonic, that is, pathological. Strategies computed in a fine-grained abstraction can be worse when evaluated in the original game than those computed in a strictly coarser abstraction (Waugh et al. 2009a). Essentially this can happen because we assume our opponent plays within the restricted strategy set that the abstraction affords him, but in reality he may use a strategy that falls outside our abstract model.

The nonmonotonicity has cast doubt on the entire abstraction approach for games. There has been a significant amount of experimental work on abstraction in games over the last dozen years, and experience suggests that in practice in large games such as heads-up Texas Hold'em poker, finer-grained abstractions yield programs that play better against other programs. Furthermore, it has been shown experimentally in heads-up limit Texas Hold'em poker that finer-grained abstractions yield programs that have lower exploitability—that is, they play better against their worst-case opponent (Johanson et al. 2011). Over the last two years, theoretical breakthroughs have finally been made that tie abstraction choices to solution quality in the original game (Sandholm and Singh 2012; Lanctot et al. 2012; Kroer and Sandholm 2014a; 2014b).

The state of practical game abstraction methodology

In this section I will review the state of practical abstraction methodology. In the following section I will review recent theoretical breakthroughs.

Information abstraction

Initially, game abstractions were created by hand, using domain-dependent knowledge (Shi and Littman 2002; Billings et al. 2003). The last eight years have witnessed the advent of work in game abstraction algorithms (Gilpin and Sandholm 2006; 2007b; Zinkevich et al. 2007). Throughout that period, important experimental advances have been made (Gilpin and Sandholm 2007a; Gilpin, Sandholm, and Sørensen 2007; 2008; Waugh et al. 2009b; Schnizlein, Bowling, and Szafron 2009; Johanson et al. 2013; Ganzfried and Sandholm 2013; 2014).

Early work on lossless abstraction enabled the solution of Rhode Island Hold'em, an AI challenge problem with 3.1 billion nodes in the game tree (Gilpin and Sandholm 2007b). When moving to even bigger games, lossless abstraction typically yields abstract games that are too large to solve, so lossy abstraction is needed.

In brief, the main ideas for practical lossy information abstraction in games have included the following.

- Using integer programming to optimize the abstraction (typically within one level of the game at a time) (Gilpin and Sandholm 2007a).
- *Potential-aware* abstraction, where the information sets of a player at a given level of the game tree are not bucketed based on some measure such as strength of a poker hand, but rather by the probability vector of transitions to state buckets at the next level (Gilpin, Sandholm, and Sørensen 2007).
- *Imperfect-recall* abstraction, where a player is forced to forget some detail that he knew earlier in the game so as to be able to computationally afford a more refined partition of more recent information (Waugh et al. 2009b).

The currently leading practical algorithm for information abstraction uses a combination of the latter two ideas and obviates the first (Ganzfried and Sandholm 2014). Using abstraction that divides the game disjointly across multiple blades of a supercomputer for equilibrium-finding computation, strong strategies have been constructed for heads-up no-limit Texas Hold'em (with stacks 200 big blinds deep as in the Annual Computer Poker Competition (ACPC)) (Brown, Ganzfried, and Sandholm 2014; 2015). That original game has 10^{165} nodes in the game tree (Johanson 2013).

Action abstraction

Action abstraction remained a manual endeavor for much longer than information abstraction. The last four years have witnessed the emergence of automated action abstraction to complement techniques in automated information abstraction. The first techniques iteratively adjusted the action discretization (bet sizing in no-limit poker) (Hawkin, Holte, and Szafron 2011; 2012). Recently, iterative methods have been introduced to do this in a way that guarantees convergence to the optimal discretization (assuming the problem is convex) (Brown and Sandholm 2014).

Action abstraction algorithms have also been developed based on discrete optimization: selecting a small set of prototypical actions to allow from the original discrete set of actions (Sandholm and Singh 2012). Both an integer programming algorithm and a greedy algorithm was developed. However, those algorithms are only for stochastic games and are not scalable computationally.

Recent theoretical breakthroughs

Since 2011, there have been significant theoretical breakthroughs in game abstraction. In brief, despite abstraction pathologies these results tie the "fineness" of abstraction to bounds on how good the strategy derived from the abstract game is in the original game.[1]

Basilico and Gatti (2011) give bounds for the special game class of Patrolling Security Games. Sandholm and Singh (2012) provide the first general framework for lossy game abstraction with bounds, but it is only for stochastic games. A key complication keeping the analysis from directly extending to extensive-form games is information sets. In stochastic games, once the opponent strategies are fixed, the best response analysis can be approached on a node-by-node basis. With information sets this is more complex, as strategies have to be evaluated not only according to how they perform at a given node, but also how they perform according to the distribution of nodes in the information set.

Kroer and Sandholm (2014a) introduce a mathematical framework that can be used to give bounds on solution quality for any perfect-recall extensive-form game. The framework uses a new notion for mapping abstract strategies to the original game, and it leverages a new equilibrium refinement for analysis. Using this framework, they develop the first general lossy extensive-form game abstraction method with bounds. Experiments show that it finds a lossless abstraction when one is available and lossy abstractions when smaller abstractions are desired. Prior abstraction algorithms typically operate level by level in the game tree. This paper proves that that can be too myopic and can therefore fail to find even obvious lossless abstractions. Recently, Kroer and Sandholm (2014b) extend the results to imperfect-recall abstraction. They also present clustering-based abstraction approaches, which have promise to be more scalable than the integer programming algorithms in their earlier paper.

Lanctot et al. (2012) also provided early theoretical bounds on lossy game abstraction. They show that running the counterfactual regret minimization (CFR) algorithm on their class of (potentially imperfect-recall) abstractions leads to bounded regret in the full game. Their work focuses on abstraction via information coarsening, thus allowing neither action abstraction nor reduction of the size of the game tree in terms of nodes. It does allow for the game tree size to be reduced by bundling information sets in certain ways.

Reverse mapping

Strategies computed in an abstraction assume the opponent adheres to the action abstraction. An actual opponent may choose actions outside the abstraction. Thus an action mapping (aka *reverse mapping* aka *reverse model* aka *action translation*) algorithm is used to map his action (e.g., bet size in poker) to one of the actions in the model (Gilpin, Sandholm, and Sørensen 2008; Rubin and Watson 2012; Schnizlein, Bowling, and Szafron 2009; Ganzfried and Sandholm 2012). Action mapping is a significant weakness in the current paradigm. For example, it tends to render even the best programs for heads-up no-limit Texas Hold'em highly exploitable by clever bet sizing. Below I will refer to bet sizing in poker for concreteness, but the approach applies to essentially any game where action sizing is an issue: bid sizing in auctions, offer sizing in negotiations, allocating quantities of attack/defense resources in security, etc.

Suppose our opponent makes a bet of size $x \in [A, B]$, where A denotes the largest betting size in our abstraction less than or equal to x, and B denotes the smallest betting size in our abstraction greater than or equal to x (assume $0 \leq A < B$). The *action mapping problem* is to determine whether we should map x to A or to B (perhaps probabilistically). Thus, our goal is to find a function $f_{A,B}(x)$, which denotes the probability that we map x to A (so, $1 - f_{A,B}(x)$ denotes the probability that we map x to B); this is the *action mapping function*.

We recently developed the first axiomatic approach to action mapping (Ganzfried and Sandholm 2013). It seems clear that an action mapping should satisfy the following basic properties.

1. If the opponent takes an action that is in our abstraction, it is natural to map his action x to the corresponding action with probability 1; hence we require that $f(A) = 1$ and $f(B) = 0$.
2. As the opponent's action x moves away from A towards B, it is natural to require that the probability of his action being mapped to A decreases.
3. Scale invariance: scaling A, B, and x by some multiplicative factor $k > 0$ does not affect the action mapping.
4. Action robustness: f changes smoothly in x (rather than having abrupt changes that can beget exploitability).
5. Boundary robustness: f changes smoothly with A and B.

We proved that no prior action mapping seriously proposed in the literature (Gilpin, Sandholm, and Sørensen 2008; Rubin and Watson 2012; Schnizlein, Bowling, and Szafron 2009)—deterministic or randomized—satisfies these five desiderata (Ganzfried and Sandholm 2013; 2012). We then introduced one that does:[2,3]

$$f_{A,B}(x) = \frac{(B-x)(1+A)}{(B-A)(1+x)}$$

We also motivated this mapping directly using analysis on small games. More recently, we conducted exploitability experiments (nemesis computations) on the *clairvoyance game* (Chen and Ankenman 2006), *Kuhn poker* (Kuhn

[1] Johanson et al. (2011) provide computational methods for evaluating the quality of a given abstraction in certain kinds of game, such as heads-up limit Texas Hold'em poker, via computing a best response in the full game after an approximate equilibrium in the abstract game has been computed.

[2] For simplicity of presentation, here we assume the pot size is 1, and that all values have been normalized accordingly.

[3] A strawman action mapping, that simply maps to the arithmetically closest action with probability proportional to the distance, also satisfies the axioms. However, we showed that it is highly exploitable, and no strong ACPC agents use it. For example, faced with a bet half way between the size of the pot and an all-in bet, it would map to each with equal probability, while such a bet should clearly be mapped to the all-in action with overwhelming probability.

1950), and *Leduc Hold'em* (Waugh et al. 2009a); it exhibited less exploitability than prior action mappings (Ganzfried and Sandholm 2013). Against programs submitted by others to the 2012 ACPC no-limit Texas Hold'em category, it performed well compared to the other action mappings—in particular, dramatically better than the prior state-of-the-art mapping, when applied to our program. So, the reduced exploitability does not come at a significant cost in performance against strong opponents that do not attempt sophisticated exploitation. Most leading ACPC teams have since adopted this mapping (e.g., (Jackson 2013)).

Selected additional topics

There are many additional promising current and future research directions in the area of game abstraction—beyond the topics reviewed above.

For instance, early approaches divided the game into sequential phases, which were then solved separately—with various techniques involved in coordinating the solutions in the phases (Billings et al. 2003; Gilpin and Sandholm 2006; 2007a). A related, newer approach is to solve the endgame that is reached anew in a finer-grained abstraction than the abstraction that was used to solve the entire game (Ganzfried and Sandholm 2014).

Another idea is to iterate between abstraction and equilibrium finding (Sandholm 2010). This has the potential that the tentative equilibrium will help guide the abstraction in which a new equilibrium is computed, and so on. The iterative action abstraction—specifically, action sizing (bet sizing when applied to poker)—technique mentioned above is an instance of this (Brown and Sandholm 2014). However, I believe this idea has significant further potential as well, both in information and action abstraction.

Another interesting phenomenon is that the equilibrium-finding algorithm used to solve the abstract game can overfit the strategy to the abstraction (Johanson et al. 2011). This can be mitigated by restricting the strategy space after the fact by post-processing the strategy to be less randomized (Ganzfried, Sandholm, and Waugh 2012). Future research involves understanding this phenomenon better and developing perhaps better strategy restrictions—to be used potentially during and after equilibrium finding.

Acknowledgments

This work was supported by the National Science Foundation under grants IIS-1320620, CCF-1101668, and IIS-0964579, as well as XSEDE computing resources provided by the Pittsburgh Supercomputing Center. I also thank Intel Corporation and IBM for their machine gifts. Noam Brown, Sam Ganzfried, and Christian Kroer provided helpful comments on a draft of this paper.

References

Archibald, C., and Shoham, Y. 2009. Modeling billiards games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Basilico, N., and Gatti, N. 2011. Automated abstractions for patrolling security games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Billings, D.; Burch, N.; Davidson, A.; Holte, R.; Schaeffer, J.; Schauenberg, T.; and Szafron, D. 2003. Approximating game-theoretic optimal strategies for full-scale poker. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*.

Brown, N., and Sandholm, T. 2014. Regret transfer and parameter optimization. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Brown, N.; Ganzfried, S.; and Sandholm, T. 2014. Hierarchical abstraction, distributed equilibrium computation, and post-processing, with application to a champion no-limit texas hold'em agent. Mimeo.

Brown, N.; Ganzfried, S.; and Sandholm, T. 2015. Tartanian7: A champion two-player no-limit texas holdem poker-playing program. In *AAAI Conference on Artificial Intelligence (AAAI)*. Demo Track publication.

Chen, B., and Ankenman, J. 2006. *The Mathematics of Poker*. ConJelCo.

Ganzfried, S., and Sandholm, T. 2012. Tartanian5: A heads-up no-limit texas hold'em poker-playing program. In *Computer Poker Symposium at the National Conference on Artificial Intelligence (AAAI)*.

Ganzfried, S., and Sandholm, T. 2013. Action translation in extensive-form games with large action spaces: Axioms, paradoxes, and the pseudo-harmonic mapping. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Ganzfried, S., and Sandholm, T. 2014. Potential-aware imperfect-recall abstraction with earth mover's distance in imperfect-information games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Ganzfried, S.; Sandholm, T.; and Waugh, K. 2012. Strategy purification and thresholding: Effective non-equilibrium approaches for playing large games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

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)*, 1007–1013.

Gilpin, A., and Sandholm, T. 2007a. Better automated abstraction techniques for imperfect information games, with application to Texas Hold'em poker. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 1168–1175.

Gilpin, A., and Sandholm, T. 2007b. Lossless abstraction of imperfect information games. *Journal of the ACM* 54(5). Early version 'Finding equilibria in large sequential games of imperfect information' appeared in the Proceedings of the ACM Conference on Electronic Commerce (EC), pages 160–169, 2006.

Gilpin, A.; Sandholm, T.; and Sørensen, T. B. 2007. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas Hold'em poker. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Gilpin, A.; Sandholm, T.; and Sørensen, T. B. 2008. A heads-up no-limit Texas Hold'em poker player: Discretized betting models and automatically generated equilibrium-finding programs. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Hawkin, J.; Holte, R.; and Szafron, D. 2011. Automated action abstraction of imperfect information extensive-form games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Hawkin, J.; Holte, R.; and Szafron, D. 2012. Using sliding windows to generate action abstractions in extensive-form games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Jackson, E. 2013. Slumbot NL: Solving large games with counterfactual regret minimization using sampling and distributed processing. In *AAAI Workshop on Computer Poker and Incomplete Information*.

Johanson, M.; Waugh, K.; Bowling, M.; and Zinkevich, M. 2011. Accelerating best response calculation in large extensive games. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Johanson, M.; Burch, N.; Valenzano, R.; and Bowling, M. 2013. Evaluating state-space abstractions in extensive-form games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Johanson, M. 2013. Measuring the size of large no-limit poker games. Technical report, University of Alberta.

Kroer, C., and Sandholm, T. 2014a. Extensive-form game abstraction with bounds. In *Proceedings of the ACM Conference on Economics and Computation (EC)*.

Kroer, C., and Sandholm, T. 2014b. Extensive-form game imperfect-recall abstractions with bounds. arXiv.

Kuhn, H. W. 1950. A simplified two-person poker. In Kuhn, H. W., and Tucker, A. W., eds., *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies, 24*. Princeton, New Jersey: Princeton University Press. 97–103.

Lanctot, M.; Gibson, R.; Burch, N.; Zinkevich, M.; and Bowling, M. 2012. No-regret learning in extensive-form games with imperfect recall. In *International Conference on Machine Learning (ICML)*.

Rubin, J., and Watson, I. 2012. Case-based strategies in computer poker. *AI Communications* 25(1):19–48.

Sandholm, T., and Singh, S. 2012. Lossy stochastic game abstraction with bounds. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*.

Sandholm, T. 2010. The state of solving large incomplete-information games, and application to poker. *AI Magazine* 13–32. Special issue on Algorithmic Game Theory.

Schnizlein, D.; Bowling, M.; and Szafron, D. 2009. Probabilistic state translation in extensive games with large action sets. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*.

Shi, J., and Littman, M. 2002. Abstraction methods for game theoretic poker. In *CG '00: Revised Papers from the Second International Conference on Computers and Games*, 333–345. London, UK: Springer-Verlag.

Waugh, K.; Schnizlein, D.; Bowling, M.; and Szafron, D. 2009a. Abstraction pathologies in extensive games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Waugh, K.; Zinkevich, M.; Johanson, M.; Kan, M.; Schnizlein, D.; and Bowling, M. 2009b. A practical use of imperfect recall. In *Symposium on Abstraction, Reformulation and Approximation (SARA)*.

Wellman, M. P.; Reeves, D. M.; Lochner, K. M.; Cheng, S.-F.; and Suri, R. 2005. Approximate strategic reasoning through hierarchical reduction of large symmetric games. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Wellman, M. 2006. Methods for empirical game-theoretic analysis (extended abstract). In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1552–1555.

Zinkevich, M.; Bowling, M.; Johanson, M.; and Piccione, C. 2007. Regret minimization in games with incomplete information. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*.