DominAI

A game-playing AI for the imperfect-information game of Dominoes

Guillermo Angeris (guille.angeris@stanford.edu), Lucy Li (lucy3@stanford.edu)

Introduction

A popular game in Latin-America, dominoes is a four player, team-based, zero-sum, imperfect information game of strategy with relatively simple rules, making it it ideal for attack with modern algorithmic and approximation tools. In general, it should be noted that the game suffers heavily from combinatorial explosion throughout the crucial opening rounds. Here, we propose an algorithm to find a good lower-bound to the expectation of a given move for the computer player in order to compute an optimal move.

Imperfect information

Games of *imperfect information* are games in which some players have information that other players are unable to see, though the game's structure and payoffs are common knowledge. There has been some work on reducing the problem of imperfect information to many problem of perfect information, most famously in the Perfect Information Monte Carlo (PIMC) algorithm[1], though other algorithms have been proposed[2][3] which either perform a similar reduction or which can solve for an exact Nash equilibrium.

Imperfect minimax search

Using the usual idea of minimax and allowing possible moves to be discounted by their current probabilities, we arrive at a simple heuristic for approximating the order of some given dominos play using techniques for perfect information games. We call this approximation the Imperfect Minimax Search (IMS).

Imperfect minimax search algorithm

```
procedure ImperfectMinimaxSearch(G, p, d)
    if G is finished or d = 0 then
        return Evaluate(G, p)
    end if
    s_max ← −∞
    for m ∈ supp (P_G(·|p)) such that m is valid in G do
        q ← P_G(m|p)
        G′ ← G updated with move by m played by p
        p′ ← next player after p performs m in game G
        s_max ← max{s_max, q · ImperfectMinimaxSearch(G′, p′, d − 1)}
    end for
    return s_max
end procedure
```

Results

Using the above algorithm and a deepened search after each turn, we received the following results for a team of AI versus a team of the simplest non-trivial strategy: greedy—i.e. dump the domino with the highest value.

Wins	Losses	Ties
40	16	1
70.1%	28.1%	1.8%

We iteratively deepened the search every *N/4* plays in an exponential fashion (as the number of possible moves exponentially decreased) and left it as a tunable parameter. The depth was of the form

$$D = \alpha 2^{\beta \lfloor \frac{N}{4} \rfloor}$$

where we allowed α=6, β=1/2.

Motivation

IMS is motivated by a few notions:

1. Each move's score should be discounted by the probability of being possible—leading to a notion of 'most likely moves'
2. This reduction allows the use of alpha-beta pruning and also allows approximation techniques of complete-information games without the need to resort to expensive sampling algorithms.

Further Thoughts

The algorithm, for all of its simplicity, performs quite well against non-trivial opponents. In particular, it was highly sensitive to the parameters of the depth, leading us to believe that the evaluation function (in our case, the difference of the expectation of the sum of pips between both players) could be further improved.

Overall, we believe that PIMC might perform better given unbiased sampling. It's not obvious how to do this efficiently, though.

References

[1] Long, J. R., Sturtevant, N. R., Buro, M., & Furtak, T. (2010). Understanding the Success of Perfect Information Monte Carlo Sampling in Game Tree Search. Proc. Assoc. Adv. Artif. Intell., 134–140.

[2] Koller, D., & Pfeffer, A. (1995). Generating and Solving Imperfect Information Games. Proc. 14th Int. Joint Conf. Artif. Intell., 14, 1185–1193.

[3] Burch, N., & Bowling, M. (2013). CFR-D: Solving Imperfect Information Games Using Decomposition. arXiv Preprint arXiv:1303.4441, 1–15. Retrieved from http://arxiv.org/abs/1303.4441