Udacity AIND Game Playing Agent Research Review

May 15, 2017

1 Research Review

1.1 Game Tree Searching by Min/Max Approximation by Ronald L. Rivest

> This paper was chosen, as it is very closely related to the project, and can/will be used as a blueprint, for future improvements to the game playing agent code.

555 words

1.1.1 Summary

The paper address the game playing agent that uses the min/max algorithm with alpha beta pruning. Given the combinatorial explosion of possibilities, standard AB algorithms have several flaws. First, the depth-first search, evaluates children in an arbitrary fashion, and with depth limit, grows uniform length branches. Secondly, the minimax algorithm depends only on minimum and maximum values amongst children of a given node. It is therefore not possible to evaluate the sensitivity of a root to other nodes. Instead we would like to expand those nodes, that have potentially the largest effect on the value of the root. Author proposes two main improvements over standard AB algorithm.

First, the iterative heuristics are used, which, instead of growing the whole tree to successive depths d=1,2,..., grows the search tree one step at a time. At each stop, the successors of a chosen expandable node are added to the tree and values from new nodes are "back propagated" to the ancestors.

Secondly, instead of using min/max function, generalized mean function is used, which in a limit approaches min/max. The advantage of using gmf is that it is differentiable, and derivatives can be used to calculate the impact of child on parent, and choose the node with greatest variability.

The choice of node to expand is done with a special case of penalty-based iterative search methods. These methods assign weight (penalty) to the edge between earch parent and child, that is based on the values stored in the nodes themselves. Penalties are added up when going from root to tip/leaf. The idea is to expand the node with the least total penalty.

In this case, the min/max approximation heuristic is used, where penalties are derivatives of the approximating functions. As the derivative measures sensitivity of the root value to changes in tip value, the node with greatest derivative is expanded (which should reduce the total uncertainty).

It is worth noting, that p-means function is computantionally expensive. For this reason, the traditional min and max calculations were used, the variation called "reverse approximation". The weight was calculated as difference between log value of the most favorable sibling and the node itself + constant penalty for "going deeper".

1.1.2 Key results

Connect Four was a chosen game in this research. They've tested the performance of their algorithm against the minimax with alpha-beta pruning. The evaluation function scored each set of four spaces in the frame, with different number of points, depending on the colours occupying it. Also, there were additional points depending on which player was up next. Algorithms were constrained by time and number of moves. 980 games were played in total, with different configuration of resources (time per turn from 1 to 5 seconds, and total number of moves from 1000 to 5000).

The min/max approximation method took around 7 times longer to evaluate at each step. That resulted in AB algorithm dominating when constrained by the time. However, when constrained by number of moves, the min/max approach was superior. That clearly shows the evaluation, and growing of the tree was done much more efficient compared to standard Alpha Beta pruning.

The other potential problem with min/max algorithm, as with other penalty based schemes, is that they require the tree to be explicitly stored, hence they are memory intensive.

Reference Game Tree Searching by Min/Max Approximation, R.L.Rivest, Artificial Intelligence 34 (1988) 77-96 https://people.csail.mit.edu/rivest/pubs/Riv87c.pdf