Playing Go by Search-Embedded Pattern Recognition

Pedro Domingos[1]
Department of Information and Computer Science
University of California, Irvine
Irvine, California 92717, U.S.A.
pedrod@ics.uci.edu
http://www.ics.uci.edu/~pedrod

Go (Iwamoto, 1976) is one of the hardest known games to play by machine. Its very large branching factor (up to 361) makes search to even a moderate depth problematic, and precludes the use of massive search that has proved successful in games like checkers and chess. As a result, researchers have turned to pure pattern recognition and pattern-based reasoning approaches, which concentrate all their effort in evaluating the current board position, and make no explicit attempt to predict the evolution of the game. However, systems of this type remain far from being able to compete with even a moderately proficient human player. This extended abstract reports on GP, an artificial Go player that combines pattern recognition with limited but flexible search, and won several games against a novice human player.

The basic motivation for GP was the observation that, while humans engage in extensive pattern-based inference when playing Go, they also perform small amounts of look-ahead search. In particular, the latter type of reasoning seems to prevail in situations of immediate attack or defense of groups with few liberties, while the former is prevalent in periods of the game where the players have a greater freedom of choice for the next move. GP recognizes and potentially acts on all variations of seven canonical patterns centered around each enemy stone, but only after it has checked that there are no more immediate tasks, like closing off the few remaining liberties of an enemy group or trying to save a troubled one of its own. If no patterns fire, it plays on handicap points if any are still free, or attempts to consolidate territory bu introducing intermediate stones between separate groups; in the long run this process will lead to the formation of walls.

All these steps are embedded in a conventional minimax search with alpha-beta cutoffs, to which a maximum branching factor is imposed (i.e., not all legal moves are expanded, but only the top few). The search also has a nominal depth limit, which is 2, 4 or 6 plies depending on the playing level chosen, but this limit varies dynamically according to the difficulty of the current situation and the stability of the evaluation. Three different static evaluation heuristics have been implemented, using combinations of points accumulated, liberties and sizes of groups.

In a series of 18 games against a novice human player, on a 9×9 board with a handicap of 3 for the machine, GP on average scored more points than the human opponent with two of the heuristics, at the higher difficulty levels; the average difference was approximately equal to the handicap. Significantly, there was a large improvement from 2-ply to 4-ply maximum depth, showing that there is an advantage to using search, but there was no significant difference between 4-ply and 6-ply, indicating that deeper search is not necessarily productive in Go, even if the available computing power allows it.

[1] This abstract describes research conducted at Instituto Superior Técnico, Lisbon, Portugal.

References

Iwamoto, Kaoru (1976). *Go for Beginners*. London: Penguin.