Saurabh Vyas
Udacity Artificial Intelligence Nanodegree
06-Feb-18

Research Review: Game Tree Searching by Min-Max Approximation

The game playing agents have always been a topic of interest for AI research. There have been some great advancements in this area. Number of algorithms have been developed that can consistently win against opponents. Examples of such algorithms are MinMax search and alpha-beta pruning search.

The current paper proposes another method of exploring the game tree i.e. an approximation of min-max algorithm. The proposed method has two key components:

1. An approximation function that can mimic min and max functions but has continuous derivative. The paper proposes 'Generalized mean-value operator' as this approximation function and this allows selecting next expandable tip. Following is the definition of 'Generalized mean-value operator':

   $$M_p(\boldsymbol{a}) = \left(\frac{1}{n}\sum_{i=1}^{n} a_i^p\right)^{1/p}$$

   This function approximates min and max operators in the following ways:

   $$\lim_{p\to\infty} M_p(\boldsymbol{a}) = \max(a_1, \ldots, a_n),$$
   $$\lim_{p\to-\infty} M_p(\boldsymbol{a}) = \min(a_1, \ldots, a_n).$$

   The partial derivative of this function that can be used for 'sensitivity analysis' and hence in selecting the next expandable node is:

   $$\frac{\partial M_p(\boldsymbol{a})}{\partial a_i} = \frac{1}{n}\left(\frac{a_i}{M_p(\boldsymbol{a})}\right)^{p-1}.$$

2. Penalty-based iterative search method: The paper discusses the general approach of penalty-based iterative search methods and then develops a specific technique based on this method. The 'min/max approximation' defines penalties in terms of derivative of the approximation function describe above.

   $$\tilde{v}_E(c) = \begin{cases} \hat{v}(c), & \text{if } c \in T(E), \\ M_p(\tilde{v}_E(d_1), \ldots, \tilde{v}_E(d_k)), & \text{if } c \in Max\backslash T(E), \\ M_{-p}(\tilde{v}_E(d_1), \ldots, \tilde{v}_E(d_k)), & \text{if } c \in Min\backslash T(E), \end{cases}$$

   The penalties are defined as:

   $$P_s(x) = \sum_{c\in A(x)} w(c),$$ and $$w(x) = -\log(D(f(x), x)).$$

   Here c is the terminal node of partially explored tree E and d is a child of root c. $\tilde{v}_E(c)$ is an estimation of actual value v(c) using a static evaluation function.

This scheme then allows us to explore next expandable node in more efficient manner. It selects the tip that has minimum penalty.

The downside of this method is the time required to calculate generalized p-means. That's the reason why alpha beta pruning algorithm beats this method if constrained by time. On the other hand, if number of moves is a constraint then the proposed method beats alpha beta pruning consistently as shown in the experiment results by the researcher.