Research Review

Advanced Game Playing

Game Tree Searching Using Min-Max Approximation – Ron Rivest 1987

This paper introduces a new technique for searching in game trees, based on the idea of approximating the min and max operators with generalized mean-value operators (GMV) performing better than min-max search with alpha-beta pruning.

For the purpose of algorithm, a generalized p-mean of *a*, $M_p(a)$, where $a=(a_{1,}..,a_n), a_i \in \Re^+, p \in \Re \setminus \{0\}$, is defined as:

$$M_p(a)=\left(\frac{1}{n}\sum_{i=1}^{n} a_i^p\right)^{\frac{1}{p}} \text{ , p > 0 and } M_0(a)=\lim_{p\to 0} M_p(a)=(a_{1,}..,a_n)^{\frac{1}{n}} \text{ , p = 0}$$

Observing that $\lim_{p\to\infty} M_p(a)=max(a_{1,}..,a_n)$, $\lim_{p\to-\infty} M_p(a)=min(a_{1,}..,a_n)$ and $p\le q \to M_p(a)\le M_q(a)$, we can come to a conclusion that the first derivation wrt each a_i is continuous, i.e.:

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

This property is far more useful for sensitivity analysis than computing derivatives of *max(a)*, which are always 0 or 1.

Penalty based scheme is introduced that introduces an approximate way to expand search in a given node based on GMV. Non-negative weight *w(e)* of an edge *e* in a game tree is a penalty; we want edges leading to unfavorable outcome to have larger penalty. Now for each node in the game tree we compute its penalty that is a sum of penalties on all edges from the node to the root node. We will then select nodes to expand with the lowest penalty. Next we define

$$\widetilde{v_E}(c)=\begin{cases} \widehat{v_E}(c) & c\in Terminal \\ M_p(\widetilde{v_E}(d_1)...\widetilde{v_E}(d_k)) & c\in Max\setminus Terminal \\ M_{-p}(\widetilde{v_E}(d_1)...\widetilde{v_E}(d_k)) & c\in Min\setminus Terminal \end{cases} \text{ and } D(x,y)=\frac{\partial\widetilde{v_E}(x)}{\partial\widetilde{v_E}(y)}$$

where $\widehat{v_E}(c)$ is value at a terminal node, *Terminal*, *Max* and *Min* are sets of nodes of given type. *D(s, c)* measures sensitivity of the root value $\widetilde{v_E}(s)$ to the changes of in the tip value $\widetilde{v_E}(c)$. Here comes the idea to choose tip *c* with the highest value of *D(s, c)* to expand, and formulate it as a penalty-based heuristics. If we define weight *w(x)* on the edge between node *x* and its parent *f(x)* as $w(x)=-\log(D(f(x),x))$, by chain rule we get $D(s,x)=\prod_{c\in A(x)} D(f(c),c)$, *A(x)* = ancestors of *x* then by choosing the game tree's tip *x* with the highest *D(s, x)* guarantees the lowest penalty, as tip's penalty is defined as $P_s(x)=\sum_{c\in A(x)} w(c)$.

By using a penalty-based scheme instead of alpha-beta pruning over minimax algorithm, employing aforementioned continuous approximation, the results depend on the game criteria – when time is restricted, alpha-beta pruning wins in ratio of 239:186 wins:loses but when the depth of searched moves is restricted, penalty-based scheme prevails with 249:190.

The penalty-based scheme is much more computationally intense with worse asymptotic running complexity, hence its usage might not be the most practical.