Analysis of Evaluation-Function Learning by Comparison of Sibling Nodes

Tomoyuki Kaneko[1] and Kunihito Hoki[2]

[1]University of Tokyo, Japan
kaneko@acm.org

[2]University of Electro-Communications

Advances in Computer Games 13

Outline

Background:

- Machine learning of evaluation functions
- Recent success in shogi

Analysis of (partial) gradient of Minmax value

- When is it differentiable?
- Is it equal to gradient of *leaf* evaluation?
  (implicitly assumed in previous work)

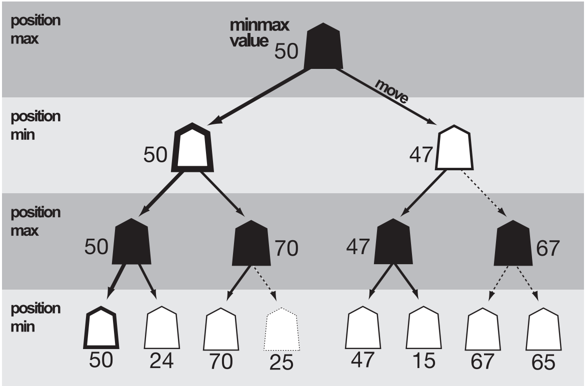
Experiments in shogi:

- How frequently is Minmax value non-differentiable?
- Upper bounds by
  - Multiple PVs
  - Different gradients in multiple PVs

Minmax search

(Tilburg photo)

Minmax search



Minmax value: result of Minmax search

- Minimum or Maximum of children (for a internal node)
- Evaluation by *evaluation function* (for a leaf node)

PV: principal variation (the left most branch)

Path from the root to a leaf, s.t. Minmax (child) = Minmax (parent)

Evaluation function

Definition $\mathbf{eval}(p, \theta)$:

- p: a game position
- $\theta \in \mathcal{R}^N$: a parameter vector

Assumption: $\mathbf{eval}(p, \theta)$ is *differentiable* w.r.t. θ

Example: $\theta = (a, b)$

- $\mathbf{eval}(p, \theta) = a \cdot \#\mathit{pawns}(p) + b \cdot \#\mathit{pieces}(p)$
- $\dfrac{\partial}{\partial a} \mathbf{eval}(p, \theta) = \#\mathit{pawns}(p)$

Motivation: machine learning

Goal of leaning evaluation functions

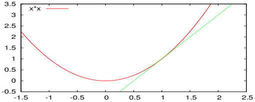
Adjustment of Minmax value via θ:

- Comparison: better Minmax value for a grandmasters' move than that of other legal moves (Nowatzyk2000, Tesauro2001, *Hoki2006*)
  - Success in shogi: outperformed all hand tuned evaluation functions
  - How it works ➔ *First talk@Session10 (tomorrow)*
- TDLeaf: similar Minmax value to that of future positions (Baxter et al. 2000)

Common problem:

How to obtain the *gradient* of Minmax value?

Partial derivative of Minmax value



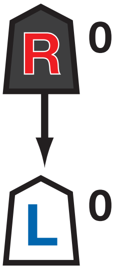
Adjustment by Gradient descent

- Goal: Adjustment of Minmax value of root (R)
- Method (ideal): update θ by $\frac{\partial}{\partial \theta_i}$ R
  - Known problem: R is *not always* partially differentiable
- Method (work around): update θ by $\frac{\partial}{\partial \theta_i}$ L
  - Work around: use $\frac{\partial}{\partial \theta_i}$ L (the leaf of PV), instead of $\frac{\partial}{\partial \theta_i}$ R
  - Observation: Equal Minmax value, R= L (by definition)
  - Expectation: Similar gradients, $\frac{\partial}{\partial \theta_i}$ R= $\frac{\partial}{\partial \theta_i}$ L

How different?: $\frac{\partial}{\partial \theta_i}$ Root (R) ↔ $\frac{\partial}{\partial \theta_i}$ PVleaf (L)

Example + informal discussion: one child

OK: $\frac{\partial}{\partial \theta_i} R = \frac{\partial}{\partial \theta_i} L$

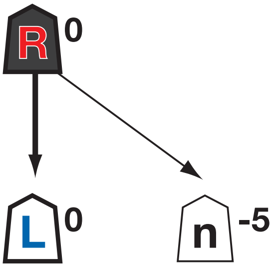


- L is always PV for any Minmax value
- Minmax value R always equals that of L

$$L + \delta = R + \delta$$

Example: two children (different leaf values)

OK: $\frac{\partial}{\partial \theta_i}$ R = $\frac{\partial}{\partial \theta_i}$ L



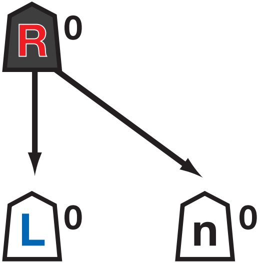
If $n = -5$ for any δ:

- L is better than n while ($L + \delta > n$, i.e., $\delta < 5$)
- L will be PV for $\delta < 5$
- Minmax value R equals that of L when $\delta < 5$,

$$L + \delta = R + \delta \quad (\delta < 5)$$

Example: two children (tie)

NG: $\frac{\partial}{\partial \theta_i}$ R (not defined)



If $n = 0$ for any δ:

- L is better than n while ($L + \delta > 0$)
- n is better than L while ($L + \delta < 0$)
- L or n will be PV for $\delta \approx 0$

L+$\delta > n$ ($\delta > 0$)
L+$\delta < n$ ($\delta < 0$)

L+ δ = R+ δ ($\delta > 0$)
L+ $\delta \neq$ R ($\delta < 0$)

Unique PV ↔ Differentiable?

True as expected:

$$\text{Unique PV} \rightarrow \left(\frac{\partial}{\partial \theta_i} \boldsymbol{R} = \frac{\partial}{\partial \theta_i} \boldsymbol{L}\right)$$

False:

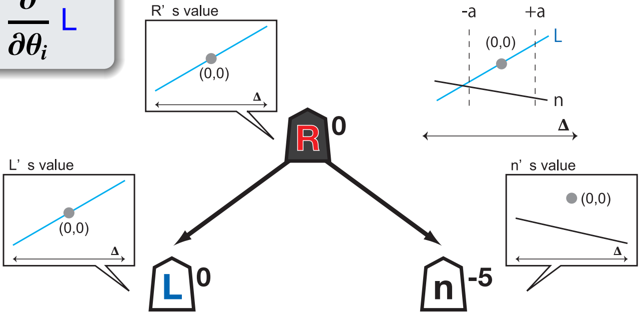
$$\frac{\partial}{\partial \theta_i}\ \text{R defined} \rightarrow \left(\text{Unique PV} \land \left(\frac{\partial}{\partial \theta_i} \boldsymbol{R} = \frac{\partial}{\partial \theta_i} \boldsymbol{L}\right)\right)$$

A counter example exists:

$$\frac{\partial}{\partial \theta_i}\ \text{R defined} \land \left(\frac{\partial}{\partial \theta_i} \boldsymbol{R} \neq \frac{\partial}{\partial \theta_i} \boldsymbol{L}\right)$$

Example: two children (different leaf values)

OK: $\frac{\partial}{\partial\theta_i} R = \frac{\partial}{\partial\theta_i} L$



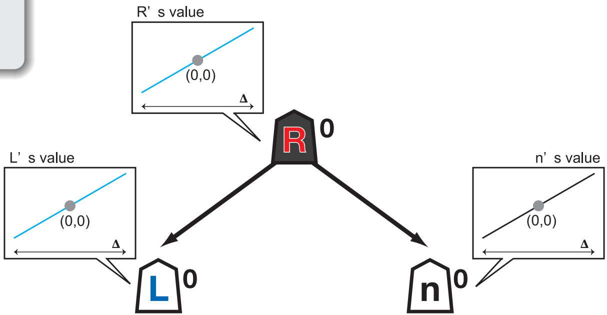
- If θ_i changed by Δ ($\theta_i \leftarrow \theta_i + \Delta$), all leaves (L and n) will change
- For any gradients of L and n, L is better than n for small $|\Delta|$

$$\left(L + \frac{\partial}{\partial\theta_i} L \cdot \Delta\right) > \left(n + \frac{\partial}{\partial\theta_i} n \cdot \Delta\right) \quad (\exists a > 0,\ |\Delta| < a),$$

$$R_{\theta_i \leftarrow \theta_i + \Delta} \approx L + \frac{\partial}{\partial\theta_i} L \cdot \Delta \quad (|\Delta| < a).$$

Example: two children (tie, same leaf gradient)

OK: $\frac{\partial}{\partial \theta_i} R = \frac{\partial}{\partial \theta_i} L$

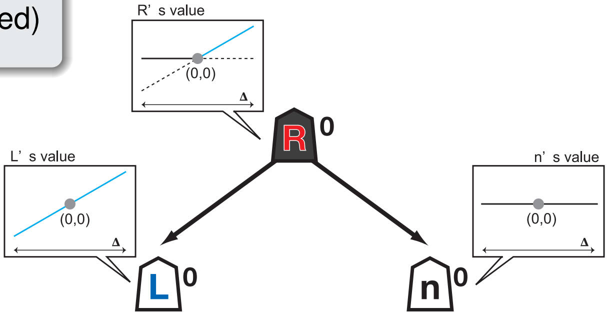


- Even if L and n has the same value,
  R is still differentiable if L and n have the same gradient.

$$R_{\theta_i \leftarrow \theta_i + \Delta} \approx L + \frac{\partial}{\partial \theta_i} L \cdot \Delta$$

Example: two children (tie, different leaf gradients)

NG: $\frac{\partial}{\partial \theta_i}$ R (not defined)



- When L and n has the same value but different gradients, change of R depends on whether $\lim_{\Delta \to +0}$ or $\lim_{\Delta \to -0}$

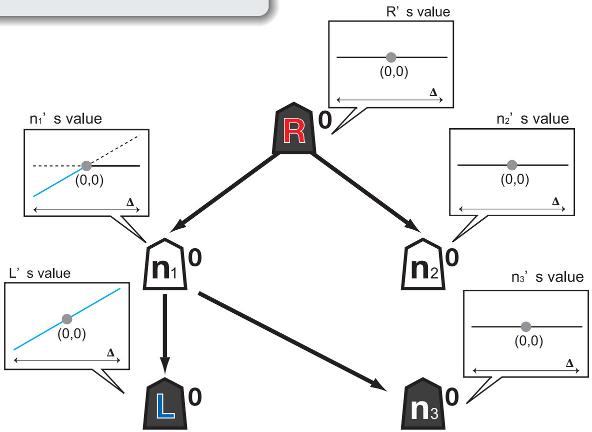
$$R_{\theta_i \leftarrow \theta_i + \Delta} \approx \begin{cases} L + \frac{\partial}{\partial \theta_i} L \cdot \Delta & (|\Delta| > 0; L \text{ is PV}) \\ n + \frac{\partial}{\partial \theta_i} n \cdot \Delta & (|\Delta| < 0; n \text{ is PV}) \end{cases}$$

Example: $\frac{\partial}{\partial \theta_i} L$ *hidden* by others

NG: $\frac{\partial}{\partial \theta_i}$ R $\neq$ $\frac{\partial}{\partial \theta_i}$ L (defined but different)

$$\frac{\partial}{\partial \theta_i} L = 1$$

$$\leftrightarrow \quad \frac{\partial}{\partial \theta_i} R = 0$$



Practical issues and experiments

(1) How frequently non-differentiable R exists?

Estimation of upper bounds in training positions by:

- Multiple PVs
- Different gradients in multiple PVs

(2) Is $\mathbf{1}$ small enough for update step $\boldsymbol{\Delta}$?

- $\boldsymbol{\Delta \geq 1}$ for integer parameters
- $\boldsymbol{\forall \epsilon > 0, \exists \delta > 0}$ for real parameters

How frequently objective function $\boldsymbol{J}$ will be improved by update along with $\boldsymbol{\frac{\partial}{\partial \theta_i} J}$, for $\boldsymbol{\Delta = 1, 2, 4,}$ and $\mathbf{8}$?

➔ please see proceedings

Experiments in shogi: evaluation functions

Practical evaluation functions:

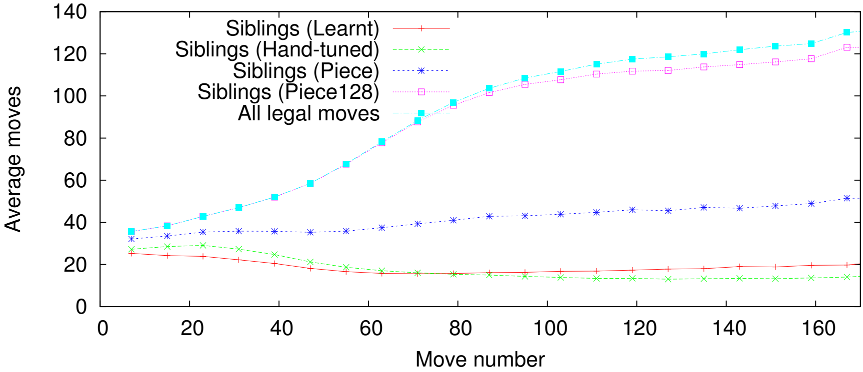
- Learnt: main evaluation function of GPSShogi revision 2590
  - Near optimal by learning
  - $\approx \mathbf{1.4}$ (8) million parameters
- Hand-tuned: old evaluation function used until 2008
  - Reasonable but far from optimal

Poor evaluation functions:

- Piece: initial values in learning
  - Same piece values as Learnt, 0 for others.
- Piece128: extreme initial values
  - 128 for piece values, 0 for others.

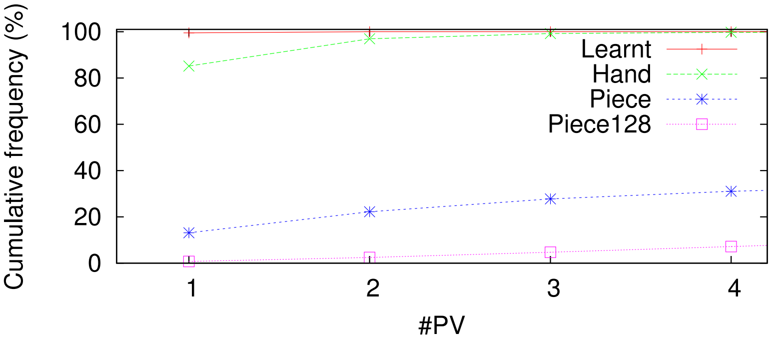
GPSShogi: open source, winner of CO 2011
`http://gps.tanaka.ecc.u-tokyo.ac.jp/gpsshogi/index.php?GPSShogiEn`

Statistics: #legal moves and #moves of similar evaluation



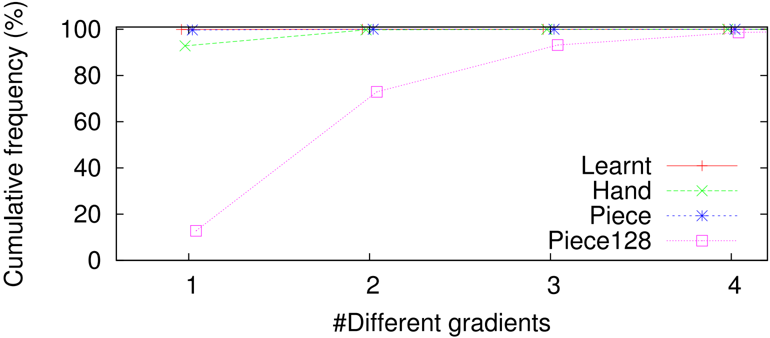
- Legal moves■: ≈ 20 (opening), ≈ 130 (endgame)
- Practical evaluation functions (Learnt+, Hand-tuned×): ≈ 20 (opening, endgame) moves in $\alpha\beta$ window of 2 pawns.
- Poor evaluation functions (Piece*, Piece128□): ≈ 40 moves or more in $\alpha\beta$ window of 2 pawns.

Frequency: number of PVs



- Practical evaluation functions: almost always unique
  - Learnt+: unique PV for almost all positions
  - Hand-tuned×: unique PV in more than 80% of positions, more than 2 PVs in less than 4% of positions.
- Poor evaluation functions: rarely unique
  - Piece*: multiple PVs for more than 86% of positions
  - Piece128□: multiple PVs for more than 99% of positions

Frequency: different gradients of pawn in multiple PVs

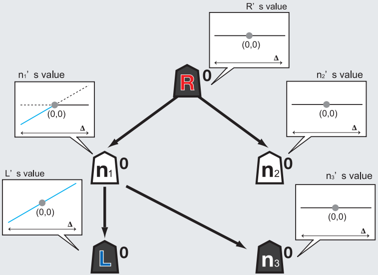


- Practical evaluation functions: almost always unique
  - Learnt+: unique gradient for almost all positions
  - Hand-tuned×: unique gradient in more than 92% of positions
- Poor evaluation functions: rarely unique for Piece128
  - Piece*: unique gradient for almost all positions
  - Piece128□: multiple gradients for more than 77% of positions

Concluding remarks

Conclusion

- Analysis on partial (sub-) gradient of Minmax value for root (R):
  - May equal that of PV leaf (L)
  - Composition of gradients of two leaves, in general
    ➔ please see proceedings for details
- Experiments in shogi:
  - Observed Multiple PVs and different gradients in multiple PVs
  - Frequent in early stage of learning



Future work

- Improved learning by composition of accurate sub-gradient of its objective function