Factorization Ranking Model for Move Prediction in the Game of Go
Chenjun Xiao and Martin Müller
Department of Computing Science, University of Alberta
{chenjun, mmueller} @ualberta.ca

Fast Move Prediction Problem in Go

Problem:
- Training data includes tens of thousands of games
- Each training case consists of possible moves in one game position, with one specified move chosen by the expert
- Moves are described by a set of features
- Supervised training of a model from game records to predict expert moves as accurately as possible

Question: How to model each training case?

Two dominating fast algorithms:
- Elo Rating: consider move selection as a competition among groups of features.
- LRF: take the interaction between features within a group into account.

Our Solution:
Combine them: model interaction between features within a group, and model move selection as a group competition.

Contrast: Deep neural networks are better move predictors, but several orders of magnitude slower.

Factorization Bradley Terry Model

Objective:
Modelling competition among groups of features by taking the interaction between features into account.

Description:
- $F$: set of features, $w \in \mathbb{R}^{|F|}$: strength of features, $v \in \mathbb{R}^{|F|+k}$, interaction features.
- Strength of a group $G$ is defined by:

$$E_G = \sum_{f \in G} w_f + \frac{1}{2} \sum_{f_1 \in G, f_2 \neq f} (v_{f_1} + v_{f_2})$$

Given $N$ groups $\{G^1, \ldots, G^N\}$,

$$P(G^i \text{ wins}) = \frac{\exp(E_{G^i})}{\sum_{j=1}^{N} \exp(E_{G^j})}$$

Training:
- Model each training case with Factorization Bradley Terry (FBT) model
- Train the model using log-loss and $\ell_2$ regularization.

$$\min_{w \in \mathbb{R}^{|F|}, v \in \mathbb{R}^{|F|+k}} \frac{1}{N} \sum_{i=1}^{N} \left[ \log \left( \sum_{G \in G} \exp(E_G) \right) + \lambda \|v\|^2 \right] + \lambda \|w\|^2$$

- Optimization with Stochastic Gradient Decent Gradient for a single model parameter $\theta \in v \cup w$ is

$$\nabla \theta \theta = -h_G(\theta) + \frac{1}{\sum_{i=1}^{N} \exp(E_{G^i})} \sum_{G \in G} \sum_{i=1}^{N} \exp(E_{G^i}) h_G(\theta)$$

where if $\theta = w$, $h(\theta) = 1$, if $\theta = v_i$, $h(\theta) = 1 - \sum_{G \in G} v_G v_i$.

Accelerating Gradient Computation

- Efficient Gradient Computation
  - Maintain a value of $E_G$ for each $G$, update it according to $E_G = \theta h_G(\theta) + \theta v_G(\theta) = E_G + (\theta - \theta) h_G(\theta)$.
  - Calculate the $h$ term by incrementally updating $R_{G^i} = R_{G^i} + \frac{1}{2} (v_G^i - v_A^i)$, and applying $h_G(\theta) = R_G - R_{G^i}.$

- Approximate Gradient Create a mini-batch of groups created by sampling $M$ groups $\{G^1, \ldots, G^M\}$ from $G^i$, with probability distribution $P = \{P(G^1), \ldots, P(G^M)\}$.
  Construct the unbiased estimated gradient:

$$\nabla \theta \theta = -h_G(\theta) + \frac{1}{M} \sum_{i=1}^{M} h_G(\theta)$$

Features used in Experiments

- Common non-pattern features, always used: Pass, Capture, Extension, Self-stari, Atari, Line and Position (edge distance perpendicular to Line), Distance to previous move, Distance to second-last move, Fuego Playout Policy, Side Extension, Corner Opening Move, CFG Distance. All of these features are implemented in Fuego [1].

- Two feature sets: Small pattern feature set adds only 3 x 3 patterns. Large pattern feature set includes large patterns [2] harvested from training set. Distribution of patterns with different size harvested at least 10 times in different game phases is provided in Figure 2.

Experiments

Three data sets of increasing size, $S_1 \subset S_2 \subset S_3$, are used, which contain 1000, 10000, and 20000 master games respectively. The games are in the public domain at https://badukmovies.com/pro_games. We test interaction dimensions $k = 5, 10$ for both FBT and LFR.

Small pattern feature set
Prediction Accuracy: FBT vs LFR

<table>
<thead>
<tr>
<th>Training Set</th>
<th>FBT5</th>
<th>FBT10</th>
<th>LFR5</th>
<th>LFR10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>32.56%</td>
<td>32.82%</td>
<td>30.01%</td>
<td>30.08%</td>
</tr>
<tr>
<td>$S_2$</td>
<td>33.18%</td>
<td>33.42%</td>
<td>30.95%</td>
<td>31.63%</td>
</tr>
<tr>
<td>$S_3$</td>
<td>33.46%</td>
<td>34.01%</td>
<td>31.13%</td>
<td>31.94%</td>
</tr>
</tbody>
</table>

Large pattern feature set
Prediction Accuracy: FBT vs LFR

<table>
<thead>
<tr>
<th>Training Set</th>
<th>FBT5</th>
<th>FBT10</th>
<th>LFR5</th>
<th>LFR10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>32.56%</td>
<td>32.82%</td>
<td>30.01%</td>
<td>30.08%</td>
</tr>
<tr>
<td>$S_2$</td>
<td>33.18%</td>
<td>33.42%</td>
<td>30.95%</td>
<td>31.63%</td>
</tr>
<tr>
<td>$S_3$</td>
<td>33.46%</td>
<td>34.01%</td>
<td>31.13%</td>
<td>31.94%</td>
</tr>
</tbody>
</table>

References