Chapter 8: Ranking Skill
Chapter Overview

- calculating the skill level from win statistics
- ELO-Ranking
- True Skill
- Team Skill
Models for play level

**idea:** Skill level can be deduced from past victories and defeats.

**model:** Every player $i$ has a skill level $s_i$. If $s_i > s_j$ then $s_i$ is very likely to win in a competition.

**applications:**

- **matchmaking:** choose interesting opponents with comparable skill level
- **ladders/rankings:** creating public rankings as an expression of prestige (compare Tennis, SC2, WOW arena, Halo2, …)
- **organizing tournaments:** assistance for draw, qualification, clearing disputes.
The ELO System

Introduced by Arpad Elo in 1970 and adopted by the World Chess Federation.

**Assumption:** player $i$’s performance $p_i$ is normal distributed around his skill level with variance $\beta^2$. $s_i$: $p_i = N(s_i, \beta^2)$

=> $s_i > s_j$ does not necessarily mean $i$ is losing against $j$

rather: $Pr(i$ wins against $j) > 50\%$

**task:** compute $Pr(p_i > p_j \mid s_i, s_j)$ (probability of $i$ playing better than $j$)

=> Difference of 2 normal distributed variables with the same variance $\beta^2$ is normal distributed with an anticipated value of $s_i - s_j$ and variance $\beta^2$

Let $\Phi$ be the accumulated density function of a normal distribution with anticipated value of 0 and a variance of 1, then follows:

$$P(p_1 > p_2 \mid s_1, s_2) = \Phi \left( \frac{s_1 - s_2}{\sqrt{2\beta}} \right)$$
Updating the ELO Ranking

- positions have to be adjusted as soon as new results are available.
- changes follow the zero-sum principle. \( s_{1}^{\text{new}} + s_{2}^{\text{new}} = s_{1} + s_{2} \)
- difference \( \Delta \) is supposed to increase the likelihood of the observation within the model
- match result: \( y \in \{0,-1,1\} \) (Win:1, Loss:-1, Draw:0)
- updating ELO Scores with the result \( y_{l} \):
  \[
  \Delta = \alpha \beta \sqrt{\pi} \left( \frac{y_{l} + 1}{2} - \Phi \left( \frac{s_{1} - s_{2}}{\sqrt{2} \beta} \right) \right)
  \]

  \( \alpha \) : weighing factor for a match \( 0 < \alpha < 1 \) (approx. 0.07 for chess)

- ELO scores need comparatively many matches to stabilize. (ca. 20)
- properties:
  - chronological order of updates is important: good for long intervals between measurements, but bad performance for tournaments, where a players skill presumably stays constant.
  - ELO system does not allow for conclusions about individual performance in team games.
  - restricted representation of results. No differentiated treatment of events with a ranking for result (e.g. motor racing, …).
True Skill

factor graphs
bi-partite graph with factor nodes and variable nodes.
  • variable nodes: describe distribution functions
  • factor nodes: model the interaction of variables
  • edges: description of variables interacting for a factor

example: Factor Graph for ELO System

\[
\begin{align*}
N(s_1, \beta^2) & \quad X_1 \\
N(s_2, \beta^2) & \quad X_2 \\
\Pi(p_1-p_2) & \quad p_1 \\
\Pi(d_1>0) & \quad d_1 \\
\Pi & \quad p_2 \\
\Pi & \quad X_3 \\
\Pi & \quad X_4
\end{align*}
\]

• True Skill: extension of ELO Systems used for XBOX360 Live
  (e.g. HALO2 ranking)

• considers:
  • skill uncertainty
  • allows conclusions for team-members in team games
    (additive performance \(t_1\))
  • result presentation as order of play results \(t_1 \geq t_2 \geq .. \geq t_m\)
Factor graph for True Skill

\[
\begin{align*}
N(\mu_1, \sigma_1^2) & \quad N(\mu_2, \sigma_2^2) & \quad N(\mu_3, \sigma_3^2) & \quad N(\mu_4, \sigma_4^2) \\
N(s_1, \beta^2) & \quad N(s_2, \beta^2) & \quad N(s_3, \beta^2) & \quad N(s_4, \beta^2) \\
N(p_1) & \quad N(p_2) & \quad N(p_3) & \quad N(p_4) \\
\Pi(t_1=p_1) & \quad \Pi(t_2=p_2+p_3) & \quad (t_3=p_4) & \quad \text{Team Distr.} \\
\Pi(d_1=t_1-t_2) & \quad \Pi(d_2=t_2-t_3) & \quad \text{Distribution of score differences} \\
\Pi(d_1>\varepsilon) & \quad \Pi(|d_2| \leq \varepsilon) \\
\end{align*}
\]

Apriori-Distr. \quad Perf. Distr.

Example: 4 Players, 3 Teams: \{(s_1), (s_2,s_3),(s_4)\}
Result: \(t_1 > \varepsilon + t_2, \ t_1 > \varepsilon + t_3, \ \varepsilon > |t_2 - t_2|\)
Factor Graph use for True Skill

- factor graph represents the distribution for $Pr(s,p,t|r,A)$
  - $r$: ranking result, $A$: team composition
  - $s$: player skill, $p$: player performance, $t$: team rating
- compute the distribution of player skill $s$ conditional to the observations $r$ and $A$:
  $$Pr(s \mid r, A) = \int \cdots \int Pr(s, p, t \mid r, A) dp \, dt$$

$s_i$ is normal distributed with mean value $\mu_i$ and standard deviation $\sigma_i$

- With the given factor graph and the current values of $\mu$ and $\sigma$ for the participating players $\Pi(d_1 > \varepsilon)$ and $\Pi(|d_2| \leq \varepsilon)$ can be estimated.
- Comparing the prediction with the actual result, one can propagate the error back to $\mu$ and $\sigma$ and adapt the model accordingly.
- Propagating probabilities and parameter updates on a factor graph are also called message-passing or belief propagation.
Training scheme for True Skill

1. **Forward propagation**: estimate the results
2. **Update of Team-performance**: redistribution of results to teams
3. **Update of a-posteriori Distributions**: propagates update-messages as far as parameters $\mu$ and $\sigma$.

**A-priori-Distr.**

**Perf. Distr.**

**Team Distr.**

**Distribution of score-differences**

\[
\begin{align*}
N(\mu_1, \sigma_1^2) & \quad N(\mu_2, \sigma_2^2) & \quad N(\mu_3, \sigma_3^2) & \quad N(\mu_4, \sigma_4^2) \\
N(s_1, \beta^2) & \quad N(s_2, \beta^2) & \quad N(s_3, \beta^2) & \quad N(s_4, \beta^2) \\
p_1 & \quad p_2 & \quad p_3 & \quad p_4 \\
\Pi(t_1=p_1) & \quad \Pi(t_2=p_2+p_3) & \quad \Pi(t_3=p_4) \\
\Pi(d_1=t_1-t_2) & \quad \Pi(d_2=t_2-t_3) \\
\Pi(d_1>\varepsilon) & \quad \Pi(|d_2| \leq \varepsilon)
\end{align*}
\]
Discussion True Skill

• Improves the ELO Systems by:
  • expansion of result representation
  • converges faster using a priori distributions for particular players
  • team Assessment

• Disadvantages of True Skill:
  • chronological order is important, even though one can assume that skill does not change between two matches. (Expansion: True Skill Trough Time 2008)
  • team skill is considered as the sum of player skills

But: In reality player synergy is much more complicated: having 5 carries in a Moba will not work
**Team Skill**

**idea:** Considering not only individual play level, but also team chemistry.

\[ \Rightarrow \text{Viewing a player’s joint performance compared to his single performance.} \]

\[ \Rightarrow \text{Some player’s performance increases when combined with specific players.} \]

**given:** A Team \( T = \{ p_1, ..., p_K \} \) with \( K \) players. Let \( t_k \) be a sub-team of \( T \) with \( k \)-elements. \( (t_k \subseteq T \land |t_k|=k) \). \( Skill(t_k) \) constitutes sub-team’s \( t_k \) skill level (for example calculated with ELO or True-Skill)

**task:** Skill level of team \( T \) considering team chemistry?

**approach:** calculating average over determined sub-team ranking
Team Skill-k

- average play level of a sub team of k size scaled to K

\[ TS_k(T) = K \cdot \frac{1}{k} \cdot \frac{1}{\binom{K}{k}} \cdot \sum_{i=1}^{\binom{K}{k}} Skill(s_{ki}) = \frac{(k-1)!(K-k)!}{(K-1)!} \cdot \sum_{i=1}^{\binom{K}{k}} Skill(s_{ki}) \]

example:

k=1 and K=5

\[ TS_1(T) = \frac{5}{1} \cdot \frac{1}{\binom{5}{1}} \cdot \sum_{i=1}^{\binom{5}{1}} Skill(s_{1i}) = \sum_{i=1}^{\binom{5}{1}} Skill(s_{1i}) \]

k=2 and K=5

\[ TS_2(T) = \frac{5}{2} \cdot \frac{1}{\binom{5}{2}} \cdot \sum_{i=1}^{\binom{5}{2}} Skill(s_{2i}) = \frac{1}{4} \sum_{i=1}^{\binom{10}{2}} Skill(s_{2i}) \]
Team Skill-ALLK-LS

Means of improvement towards Team Skill \( k \):

- determining \( k \) is hard \( \Rightarrow \) take all possible sub-teams.
- separate results do not exist for all sub-teams
  \( \Rightarrow \) only consider sub-teams with a reliable ranking.

Idea: Consider all sub-team with a reliable estimate and which are not a subset o a reliably estimated sub-team.

Approach: Determine all relevant sub-teams \( t_{k,i}^* \) whose \( \text{Skill}(t_{k,i}) \) can be determined and for which no sub-team \( t_{k+l,j} \neq t_{k,i} \) exists.

Calculate team performance as a \( k \)-multiple of average single performance.

\[
TS_{\text{ALL-LS}}(T) = \frac{K}{\sum_{m \in \{m | \exists t_m^* \neq \{} \}}^{m=1} E(t_m^*)} = \frac{K}{\sum_{m=1}^{K} \left( \frac{1}{l} \cdot \sum_{i=1}^{l} \text{Skill}(t_{m,i}^*) \right)}
\]
Example: Team Skill ALL-LS

**TS_{ALL-LS}(T) = \frac{4}{3+2} \left( Skill(t_{BCD}) + \frac{1}{2} (Skill(t_{AC}) + Skill(t_{AD})) \right)**

red: pruned area, blue: used sub-teams, grey: pruned sub-teams.
Conclusion

• method for capturing increased success of teams with good chemistry.

• team skill depends on data of as many different team compositions as possible

• approaches for improvement:
  • roles within the team are not required explicitly
  • confidence of the underlying skill estimation is not treated
  • correlation between team skill and player skill is assumed to be uniform

• Skill in Team Skill, True Skill and ELO symmetrically values win and loss.

=> in many casual games an win award more increase to player score than losses reduces the skill level (keep players motivated to play)
Alternative Approach

• rating players not by success, but by skillful behavior:
  1. collect and describe spatial-temporal behavior over the full spectrum of skill.
  2. learn a regression model.
  3. rate player, while playing, for his $k$ last actions.

• this approach is used for dynamic play level adjustment in PVE.

• very suitable if it is known what constitutes successful behavior in the game. (e.g. accuracy in FPS Games, DPS/HPS Numbers in MMORPGS)
Learning goals

• Scope of application for player ranking and matchmaking
• ELO
• True Skill
• Team Skill
Literature


