Skill-based Matchmaking with Bayesian Inference

Nazeef Hamid

UQMSS Math Talk — 8th March 2024

Context

- Skill-based matchmaking in competitive multiplayer games

Figure: A "gamer" playing a competitive multiplayer videogame

- Matches should be fair and fun
- We need a way to **accurately** model the skill of each player

Aims

Our focus is the *TrueSkill* skill rating system, developed by Microsoft and used for games such as *Halo 3*.

1. What does a skill rating system do?
2. A brief introduction to Bayesian inference
3. *TrueSkill*'s model for a players skill
4. A remark on the 2018 sequel *TrueSkill 2*

TrueSkill is not the only skill rating system:

- Elo
- Glicko and Glicko 2

What should a Skill Rating System do?

1. Accurately model a player's skill to create fair games
2. Create incentives to play the game "properly"
   - Reward winning, penalise losing
   - Reward completing the objective, supporting teammates
   - Punish poor behaviour such as quitting
3. Be robust to manipulation
4. Be simple for a game developer to integrate
   - Easy to interpret (simple)
   - Low maintenance (independent)
   - Adapt to new games / gamemodes (flexible)
   - Low computational requirements (cheap)

Bayesian Inference

- (Statistical) Inference - predicting parameters using data e.g.

$$\mu \quad \sigma^2$$

- In Bayesian statistics parameters are considered to be random variables

- Skill can be modelled as a random variable, since players do not always perform at the same level.

- Bayesian statistics provides a tool to update our belief about a parameter when given new information

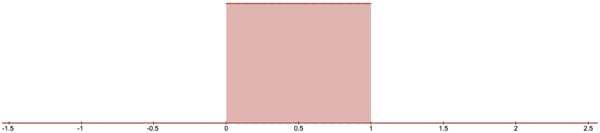
Bayesian Inference - The Idea

1. **PRIOR:** What do we currently believe about the parameter?
2. **DATA:** What is the relationship between the parameter and the data we collected?
3. **POSTERIOR:** With this data in mind, how has our belief about the parameter changed?

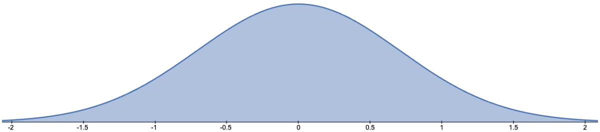
Bayesian Inference - The Belief

We want to formalise the idea of "belief". Consider the parameter y.

- "I know $y \in [0, 1]$, but nothing more than that"



- "I'm pretty sure y is around 0, but it could be any number"



Bayesian Inference - The Belief

Other distributions

- $y \sim \text{U}[0,1]$

$$f(y) = \begin{cases} 1 & \text{if } y \in [0,1] \\ 0 & \text{else} \end{cases}$$

- $y \sim \text{Bin}(n,p)$

$$f(y) = \binom{n}{y} p^y (1-p)^{n-y}$$

- $y \sim \text{N}(\mu, \sigma^2)$

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}(y-\mu)^2\right)$$

and many more!

Bayesian Inference - The Likelihood function

Specify a distribution of the data x given the parameter(s) of interest.

$$X \sim \text{Bin}(n, p) \leftrightarrow f(x|p) = \binom{n}{x} p^x (1-p)^{n-x}$$

Once data is obtained, say $x = 3$, it becomes a statement about the likelihood of the parameter:

$$f(x|p) = \binom{10}{3} p^3 (1-p)^7$$

p is the only unknown in the above equation.

Performing Bayesian Inference

Suppose we want to predict a parameter y using data $\mathbf{x}$.

1. **PRIOR:** Specify a prior distribution of y - $f(y)$
2. **DATA:** Specify a distribution for the data - $f(\mathbf{x}|y)$
3. **POSTERIOR:** Perform the update to obtain the posterior distribution using Bayes Rule:

$$f(y|\mathbf{x}) \propto \frac{f(\mathbf{x}|y)f(y)}{f(\mathbf{x})}$$

Example

We toss a coin 10 times and observe 3 heads. What is the probability p of a head.

1. **PRIOR:** Take an uninformative prior

$$f(p) = \begin{cases} 1 & \text{if } p \in [0,1] \\ 0 & \text{else} \end{cases}$$

2. **DATA:** The given data is the number of heads from coin tosses.

$$f(x|p) = \binom{10}{x} p^x (1-p)^{n-x}$$

We are given that $x = 3$ - obtain a likelihood for p

$$f(x|p) = \binom{10}{3} p^3 (1-p)^7$$

Example Continued

3. **POSTERIOR:** Perform the update using Bayes Rule:

$$
\begin{aligned}
f(p|x) &= \frac{f(x|p)f(p)}{f(x)} \\
&= \frac{\binom{10}{3} p^3(1-p)^7 \cdot 1}{f(x)} \\
&\propto p^3(1-p)^7 \\
\therefore p|x &\sim \text{Beta}(4, 8)
\end{aligned}
$$

Example Continued

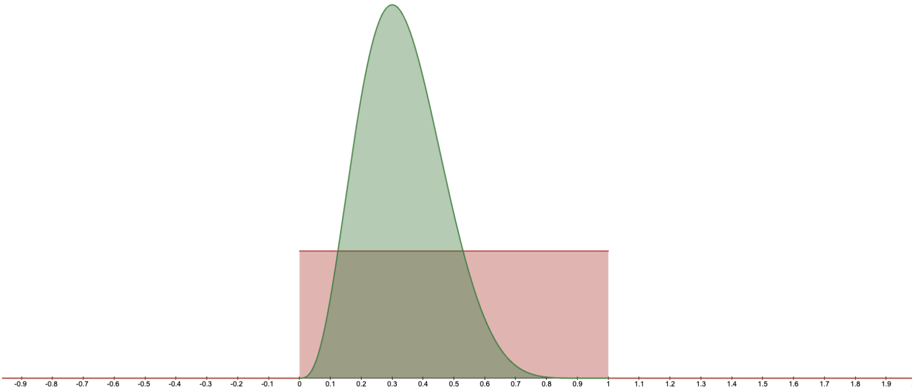


Figure: Prior and Posterior Distribution of p

Example 2

A trickier problem

1. **PRIOR:** $\lambda \sim \text{Gamma}(\alpha, \beta)$

$$f(\lambda) = \frac{\beta^{\alpha} \lambda^{\alpha-1} \exp(-\beta\lambda)}{\Gamma(\alpha)}$$

2. **DATA:** $X \sim \text{Poisson}(\lambda)$

$$f(x|\lambda) = \frac{\exp(-\lambda)\lambda^{x}}{x!}$$

3. **POSTERIOR:** Perform the update using Bayes Rule:

$$f(y|\mathbf{x}) \propto f(\mathbf{x}|y) f(y)$$

Example 2 Continued

3. **POSTERIOR:**

$$f(\lambda|x) \propto f(x|\lambda)f(\lambda)$$

$$\propto \frac{\exp(-\lambda)\lambda^x}{x!}\frac{\beta^\alpha\lambda^{\alpha-1}\exp(-\beta\lambda)}{\Gamma(\alpha)}$$

$$\propto \exp(-\lambda)\lambda^x\lambda^{\alpha-1}\exp(-\beta\lambda)$$

$$\propto \lambda^{x+\alpha-1}\exp(-(\beta+1)\lambda)$$

$$\lambda|x \sim \text{Gamma}(x+\alpha, \beta+1)$$

Example 2 Continued

See Demo

The *TrueSkill* Model

- **PRIOR**: The players' initial/prior skill is modelled by a normal distribution

$$f(\text{skill}) \sim \mathrm{N}(\mu, \sigma^2)$$

$$\sigma^2 = \underbrace{\gamma^2}_{\text{experience}} + \underbrace{\tau^2(t' - t)}_{\text{skill decay}}$$

- **DATA**: The players' performance in a match is modelled by a normal distribution

$$f(\text{perf}) \sim \mathrm{N}(\text{skill}, \beta^2)$$

  β is a tunable parameter that depends on how random games are.

- **POSTERIOR**: Using Bayes rule:

$$f(\text{skill'}|\text{perf}) = \frac{f(\text{perf}|\text{skill})f(\text{skill})}{f(\text{perf})}$$

The *TrueSkill* Model

The posterior distribution can be approximated by a normal distribution. The new skill rating μ' assigned to the player is the mean of this distribution.

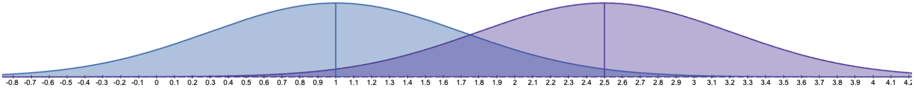


Figure: Idealised skill rating update

The *TrueSkill* Model

- More complicated in reality:

$$f(\text{skill}'|\text{perf}, \text{conditions}) = \frac{f(\text{perf}|\text{skill}, \text{conditions})f(\text{skill})}{f(\text{perf}|\text{conditions})}$$

- "conditions" refer to aspects of a match that cannot be modelled
- The problem is not solved analytically
- Calculation can be performed efficiently using an algorithm such as Expectation Propagation (EP)
- EP is an iterative method that can approximate probability distributions

The *TrueSkill* Model

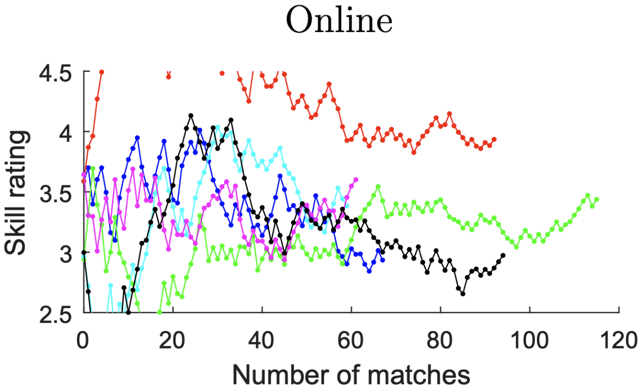


Figure: Skill ratings of several players in *Halo 5*

Tuning the *TrueSkill* model

There are a variety of parameters in the model that give it flexibility:

- μ_0, σ_0: The mean skill and skill variation of a brand new player
- β^2: Match randomness parameter
- γ: Skill increase due to experience
- τ^2: Skill decay over time parameter

These would be learned from historical match data by choosing parameters that best predict match outcomes.

Evaluating the *TrueSkill* Model

Game Mode	Number of Matches per Player
16 Players Free-For-All	3
8 Players Free-For-All	3
4 Players Free-For-All	5
2 Players Free-For-All	12
4 Teams/2 Players Per Team	10
4 Teams/4 Players Per Team	20
2 Teams/4 Players Per Team	46
2 Teams/8 Players Per Team	91

Table: Minimum matches per Player to obtain a confident skill rating

TrueSkill 2

- On historical match data from *Halo 5*, *TrueSkill* is only 52% accurate
- *TrueSkill2* is 68% accurate - substantial improvement
- Additional factors that *TrueSkill2* considers:
  - Performance includes individual statistics such as k/d ratio.
  - A quit is treated as a surrender
  - Skill is correlated with other gamemodes
  - Players in a squad are assumed to perform better
- *TrueSkill2* is planned for implementation into *League of Legends*

Conclusion

- A system for modelling the skill of players in competitive games
- Some of the basic theory behind Bayesian inference
- How the Bayesian toolkit is suited to sequential data problems
- A novel application of statistics.

References

Ralf Herbrich, Tom Minka, and Thore Graepel.
TrueSkill(TM): A Bayesian Skill Rating System.
Advances in Neural Information Processing Systems 20, 2007.

Sharon Lee.
STAT3001 Lecture Notes.
2023.

Tom Minka, Ryan Cleven, and Yordan Zaykov.
TrueSkill 2: An improved Bayesian skill rating system.
Microsoft Research, 2018.