An embeddable uncertainty module for strategy game

Yueyi Zhuo

Nanjing university of science and technology, School of Computer Science and Technology

Overconfident player and their hindsight

- History strategy game players, as history enthusiast, usually have knowledge that their played historical character doesn't have.
- They make unrealistic (over-confident) decisions based on the knowledge "leaked" from history material, which is unfair for the side which has an intelligence advantage in history and leads to a bad history simulation picture.


Takagi (player): Wait, since I have known the location of the USS task force from my history textbook, why not just launch an airstrike on that point?


Fletcher (player): Always has been. (has sent bomber squadrons without any scouting).

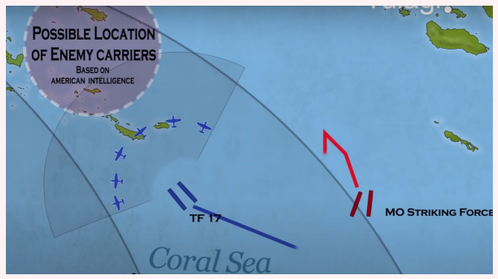
- Limiting player knowledge is impractical, but at least we can limit AI know what they should know exactly.
- Even full AI self playing, while not very "gaming", is useful to illustrate the possibility of counterfactual history and analysis for important factors.

Bad practice in recent games

- The poor understanding of the generating process leads to inconsistent mechanisms, in which many independence are wrongly assumed.
- Ex: Theater Reconnaissance in wargame Operational Art of War (TOAW), representing spies on the ground and Reconnaissance aircraft. It sets some percentage of enemy hexes to "observed" according to Theater Recon level.

Inverse problem of Bayesian Optimization

- Hidden enemy deployment in FOG = latent "trace" spatio-temporal function $f(x, y, t)$
- History commanders decide how to send recon aircraft = run a explore-only Bayesian Optimization.
- From history data we know which points are selected by the commander.
- The game gives a parameter known model, and what varys is the player's mind model. Same scenario, different playthrough.
- Here's an inverse problem - how to estimate the player's mind model which drives their history decision, or player reaction for a specific condition?
- The estimated model can be used as a module to emulate a decent player which only knows what they're expected to know and make realistic scouting decisions according to the environment.

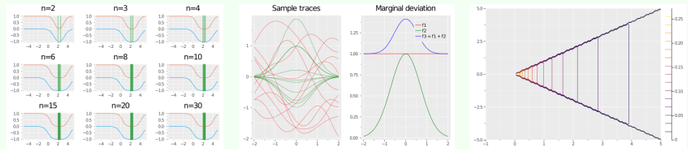


- When the enemy is in the east of him, Fletcher just sends his scouting aircrafts to north-west? 😳 While it's such a terrible mistake, we will build a model, as an unfortunate proxy of Fletcher, to make the same action in such an environment.

- Roughly, setting "observed" is i.i.d. But usually, spatial-temporal correlation matters.
- Time correlation is crucial, just think about inconsistent results given by varying turn time length. (1 hour / turn vs 10 min / turn) with i.i.d setting.
- Factors like thick cloud hindering scouting can be implicitly modeled by spatial correlation (Though TOAW itself has an explicit weather model, it performs badly.)
- If one unit failed to detect enemy from (x_1, y_1, t_1), it's very likely that the units at (x_2, y_2, t_1) or same unit in next turn (x_1, y_1, t_2) still failed to detect enemy, since the factor causing first detection failed doesn't changed that faster across spatial and time dimension.
- Ex: IJN searching on afternoon 7 May 1942.
- While systems like TOAW represent false-negative to some extent, they almost always failed to represent any false-positive.
- Ex: IJN searching on morning of 7 May 1942 recognised an oilship as a carrier task force for some hours, thus missing a decisive victory opportunity.

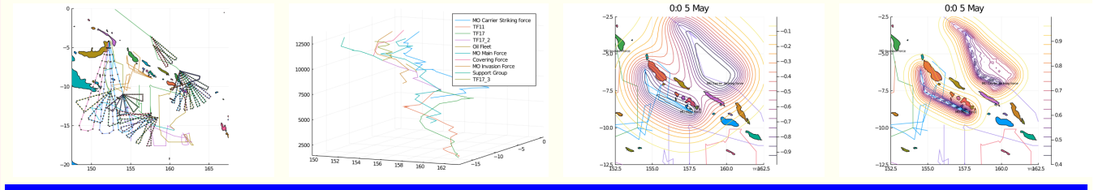
Gaussian Process as a realistic noise generator

- The correlation problem can be elegantly addressed by the Gaussian Process with a stationary kernel.
- Non-stationary kernel (ex: linear kernel) can also be used as a basis function which is able to introduce extra volatility or enemy plan hypothesis testing.



Case study: The battle of Coral Sea

- This battle shows the importance of intelligence and how bad the existing simulation is.
- If the system works on a real history battle, it's expected that it will work on similar games as well.
- `Stheno.jl`'s Gaussian process Probabilistic Programming (GPPP) suite is used for inference.



Versatile

- The method can be used in a hex based (old school wargames) or region based system (Paradox's grand strategy game) easily.

- Just choose a proper distance (Hex based systems have an old problem discussing how to measure unit range, center euclidean distance or movement cost distance?).