Sichuan Mahjong AI
Research Proposal

XiangYu Wang

1.Objective

Develop a height intelligence level of reinforcement learning agent for Sichuan Mahjong.

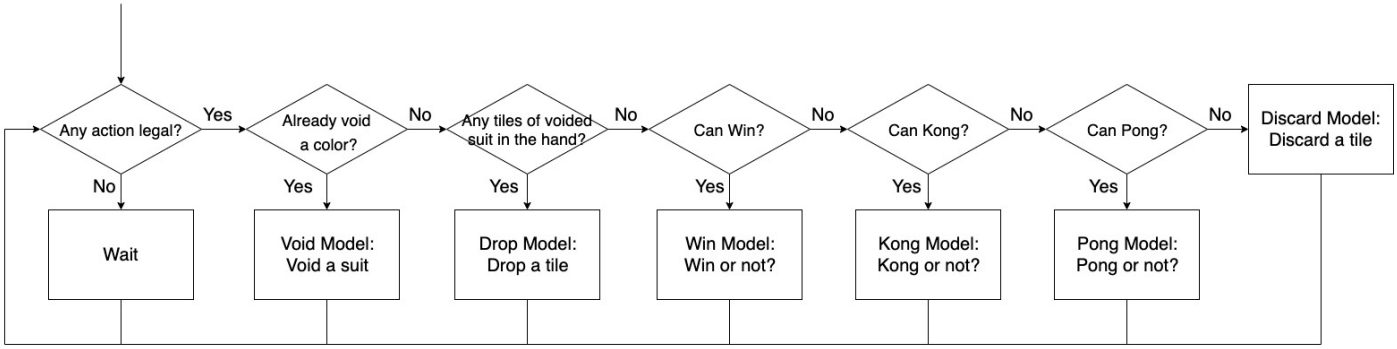
2.Background

Mahjong is more complex than other board games because a player can only have partial information about opponents, which presents unique challenges for an AI system. So far, Microsoft Suphx [1] is the most powerful AI for Japan Mahjong. Suphx is a Reinforcement Learning AI that consists of 5 policy based deep Q networks (DQN) [2], i.e., Pong/Kong/Richii/Chow/Discard Model, each of which is a ResNet with 50+ hidden layers.

Suphx gives me many useful ideas about mahjong AI. Before that, some simple attempts have been made. I have already developed a heuristic agent and use it to train a simple value based DDQN agent. As a beginner level work, the model just consists of 1 simple 4-layer FNN so that it does not give a satisfactory performance. In this project, I plan to divide the model into 5 ones, i.e., Pong/Kong/Void/Win/ Discard Model and try more complex model such as ResNet and LSTM to improve the performance.

3.Methodology

3.1.Decision Flow



3.2.Encoding

Unlike Japan Mahjong, all players in Sichuan Mahjong have to **void** a suit immediately after dealing. During the game, the player cannot discard other tiles when he/she still has the tiles of the voided suit in his/her hand. Also, the tiles of voided suit cannot be melded. Thus, the voided suit must be encoded.

How to effectively embed the voided suit to a matrix? This a problem worthy of further research.

3.3.Learning

The model is policy based, and the regularization term of entropy is used to make the learning stable:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[\pi_\theta(a_i|s_i)A^{\pi_\theta}(s_i,a_i) + \alpha H\left(\pi_\theta(s_i)\right)\right]$$

$\alpha > 0$ is a periodically updated parameter:

$$\alpha += \frac{\beta}{N}\sum_{i=1}^{N}\left[H_{target}\left(\pi_\theta(s_i)\right) - H\left(\pi_\theta(s_i)\right)\right]$$

$\beta > 0$ is a given parameter.

To accelerate the learning, some game data of human master is collected to pretrain the model (Supervised learning). Then, the model is further trained by self-playing (Reinforcement learning).

Reinforcement learning is quite time-consuming. In my previous work, it took 3 days to train the DDQN on 10000 games (batch size = 32). The model is much more complicated than previous one, so distributed machine learning algorithm must be applied. When self-playing, the slaves get the experience from the driver to update the parameters. Then, the driver periodically obtains the latest parameter to self-play.

3.4.Oracle Agent

Mahjong has rich hidden information. If a player cannot access such hidden information, it will be very hard to make a good decision. As a result, the learning is very slow. To accelerate the training, oracle guiding is introduced, which can access all the information of opponents. At beginning, the agent can see all the information of opponents to accelerate the learning. Then, we gradually mask the perfect information to make the agent become normal. Finally, the agent is trained.

3.5.Fine-tuning

In different cases of initial hand, the strategy should be adjusted. For example, if a good initial is dealt, then the player should prefer to attack, otherwise the player should prefer to defense. After the dealing, the model should be finetuned. First, the model runs *N* simulated games with the same initial hand and memorizes the trajectory set *E*. Then, the fine-tuned model θ' is:

$$\theta' = argmax_\theta \sum_{e\in E} S(e)Pr(e|\theta)$$

S(*e*): the score obtained by trajectory *e*

Finally, θ' is used to play against other opponents.

4.Evaluation

Accuracy is used to evaluate the supervised learning, i.e., What percentage of actions predicted is taken by a human master/heuristic agent. For Reinforcement learning, we evaluate the performance by actual combat.

5.Significance

Mahjong is an example of game with rich incomplete information. In the real world, many decisions are made on rich incomplete information, so some methods of Mahjong AI may give some light on many real problems.

6.Reference

[1] Li, Junjie; Koyamada, Sotetsu; Ye, Qiwei; Liu, Guoqing; Wang, Chao; Yang, Ruihan; Zhao, Li; Qin, Tao; Liu, Tie-Yan; Hon, Hsiao-Wuen(2020). “Suphx: Mastering Mahjong with Deep Reinforcement Learning”. arXiv:2003.13590

[2] van Hasselt, Hado; Guez, Arthur; Silver, David (2015). "Deep reinforcement learning with double Q-learning". AAAI Conference on Artificial Intelligence: 2094–2100.