ISSN 2959-6157

Analysis of the Application of Artificial Intelligence in Mahjong

Weikai Liu

Sir Winston Churchill Secondary School, Vancouver, V6P 3P7, Canada
*Corresponding author: 2663974@learn.vsb.bc.ca

Abstract:

Over the past few years, AI-based models for Mahjong have received a substantial amount of contribution due to the advancements in machine learning and game theory. In this review, these state-of-the-art AI models are considered, including Tjong, Kanachan, Suphx, and Zhejiang University's developed model. The deployed approaches are disparately represented by reinforcement learning, Monte Carlo tree search (MCTS), and deep neural networks, all of which aim at solving the issues that come with the game's structure, such as incomplete and overburdening information. The Tjong model complements the power of the transformer architecture with hierarchical decision-making and in turn, brings strategic depth to the gameplay. Kanachan employs Q-learning and MCTS to optimize decision-making, while Suphx combines these methods with the novel ideas of Double Q-learning and Thompson Sampling to achieve higher performance than seen before. The integration of reinforcement learning and a new evaluation model imbues the model with the unique properties of both learning machines and human expertise, namely depth and intelligence. As a result of all these exploits, obstacles still exist, the most notable among them are the management of completely unknown information, handling long-distance games, and the strategic balance between offense and defense. In the future, determinants of probabilistic reasoning, the improvement of deep learning systems, and the prowess of reward shaping will result in AI improvement in Mahjong.

Keywords: Deep Q-learning (DQN); Reinforcement Learning; Convolutional Neural Network (CNN); Mahjong; Game AI Optimization.

1. Introduction

Mahjong is a typical game known for its multi-faceted nature. It is a game of complication when it involves players making choices after experiences without complete information. What clearly distinguishes Mahjong from fully visible games like chess or Go is its strategic depth, which is hidden in the incomplete information and the necessity to predict the opponent's moves. This complexity has led to interesting AI research into Mahjong, intending to design models that can imitate or outperform human reasoning and decision-making in the game.

Mahjong's integration with Game theory in AI in recent years has created a strong base for solving the above-mentioned hurdles. Game theory, which centers on the strategic interactions among rational decision-makers, is particularly valuable in environments with partial information because it helps the players make decisions. AI models may assess twists and turns, find out the best strategies for play, and determine what the competitors would likely do because they are equipped with game-theoretic principles, which can give them an edge regardless of the contingency.

The study reviews the most up-to-date developments in Mahjong AI, which were integrated with reinforcement learning and Monte Carlo Tree Search (MCTS) techniques. Those models present how AI can master Mahjong - a game with intricate strategy - with the use of advanced computational approaches and the application of game-theoretic assessments. The objective of this review is to portray the main role that the devised models play in expanding the boundaries of AI to the areas where human decision-making is fraught with uncertainties.

Note that Mahjong is a complex game without a fixed rule, which means that different regions utilize different rules for Mahjong. The models this paper is introducing are based on either the rules of the respected region or the standard international rules.

2. Tjong: A Transformer-based Mahjong AI

The Tjong model integrates a hierarchical decision-making scheme with a transformer-based architecture, as this is an effective way of dealing with the complexities of Mahjong using a sequence model and a decision process

[1].

2.1 Hierarchical Decision-Making Framework

The Tjong model organizes decision-making into three hierarchical layers:

2.1.1 Strategic layer

This layer is the strategic level of decision-making, making the right move by choosing the appropriate best hand for given circumstances. The strategic layer uses a fan-out mechanism in which these future potential hands (or "fans") are thought out and contemplated first [1]. The model finds the highest amount of the expected value *E*[*V*-future] for future hands and determines this current action a_t, which would maximize the sum value. Mathematically, the above expression can be rewritten as:

$$\max_{a_t} E[V_{future}(a_t)] \tag{1}$$

the *V* future becomes the reward to which the value function is to be increased [1].

2.1.2 Tactical layer

At the next layer, the AI is the one to make a quick decision once a given event occurs, such as the tile to be discarded. The strategic layer consists of a policy network, which is driven by reinforcement learning optimization [1]. The policy $\pi(a_t \mid s_t)$ is then recalibrated every time the reward R_t from the environment is updated:

$$\nabla \theta\, \pi(a_t \mid s_t) = \nabla \theta log\pi\theta(a_t \mid s_t) \cdot R_t \tag{2}$$

where θare the parameters of the policy network.

2.1.3 Operational layer

The operational layer refers to the execution of the selected action. It involves making sure the tile is discarded or a meld is declared while observing the given tactical decision. This part of the hierarchy is much easier, as it means doing what those in the upper layers have decided to do [6].

2.2 Transformer-based Architecture

The Tjong AI uses a transformer model, which is built around the Gaming World as the state (tagset) and a sequence of actions [1]. This particular functionality operates ideally in Mahjong because of the self-attention mechanism of the transformer, which determines the significance of each specific Mahjong tile in the hand or on the board. The way the model is constructed is like: 1) Input Embeddings: Each tile on the game board is represented as an embedding vector x_i, where i is the tile position [1]. 2) Self-Attention Mechanism: The self-attention mechanism computes the relevance of each tile for others using (Eq. (3)). Here, Q (query), K (key), and V (value) are linear transformations of the input embeddings, and d_k is the dimension of the key vectors [1]. 3) Output: The output of the transformer is a sequence of vectors that encode the relationship between tiles and the possible actions, feeding into the tactical and strategic layers for decision-making.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{3}$$

3. Artificial Intelligence Models based on Machine Learning

3.1 Kanachan: Reinforcement Learning Model for Mahjong

Kanachan uses Q-learning [2], a popular RL algorithm, to update its strategy. The core of Q-learning involves the update rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\,[R_t + \gamma \max_a (s_{t+1}, a) - Q(s_t, a_t)] \tag{4}$$

$Q(s_t, a_t) = R_t + ϒQ(s_{t+1})$ is the estimated reward for taking action at state *st*. α is the learning rate. ϒ is the discount factor for future rewards. R_t is the correct reward after taking action a_t. In Mahjong, the state s_t includes the tiles in hand, the visible tiles on the board, and the potential hands that could be formed. The actions are the possible moves (e.g., discarding a tile) [2].

Monte Carlo Tree Search (MCTS) is used to investigate potential future strategies and moves and how they would probably end [2]. The essential steps to be followed for MCTS are: 1) Selection: The algorithm begins at the tree root and keeps choosing child nodes through the selection policy. UCB (Upper Confidence Bound) is commonly applied (Eq. (5)), where W_i is the win rate, N_i is the visit count of node i, Np is the visit count of the parent node, and c controls the exploration-exploitation balance [2]. 2) Expansion: Once the search tree has reached a leaf node, the tree is expanded by adding the move options as child nodes to this node. 3) Simulation: From this new node onwards, a rollout is done, which goes till the end of the game, and making decisions as well, is a part of it. 4) Backpropagation: The results of the simulation are passed back upwards through the tree, updating the statistics of the nodes engaged. This process allows Kanachan to evaluate the long-term potential of different moves, balancing immediate gains with prospects [2].

$$UCB = \frac{W_i}{N_i} + c\sqrt{\frac{lnN_p}{N_i}} \quad (5)$$

By employing this method, Kanachan can create a set of strategies that would take into account the short-term gains and the long-term prospects, which will hopefully improve its winning chances [2].

3.2 Monte Carlo Model by Tokyo University

The Artificial Intelligence (AI) model focuses on integrating the principle of reinforcement learning (RL) and Monte Carlo Tree Search (MCTS) such as Kanachan, in contradistinction, the model diversified its optimization in the Mahjong game's inner structural principle.

3.2.1 Reinforcement learning framework

The model uses the derivative of Q-learning, also known as Double Q-learning [3], which alleviates the problem of the overestimation of Q-values. In Double Q-learning, two separate Q-value estimators are maintained, Q1 and Q2 [3]. The updating rule of each can be described as:

$$\begin{aligned} Q_1(s_t, a_t) &\leftarrow Q_1(s_t, a_t) + \alpha\,[R_t + \gamma Q_2(s_{t+1}, \arg\max_a Q_1(s_{t+1}, a)) - Q_1(s_t, a_t)] \\ Q_2(s_t, a_t) &\leftarrow Q_2(s_t, a_t) + \alpha\,[R_t + \gamma Q_1(s_{t+1}, \arg\max_a Q_2(s_{t+1}, a)) - Q_2(s_t, a_t)] \end{aligned} \quad (6)$$

This method adjusts both values in the light of the information, hence reducing the bias and enabling a stable learning process [3], even in Mahjong, which is very random, there is still a hidden information base that might tilt the tables.

3.2.2 Monte carlo tree search (MCTS)

This model applies the MCTS method as the key approach for option assessment; just like Kanachan MCTS, takes it one step further by introducing a more elaborate selection policy based on Thompson Sampling [3]. Thompson Sampling is an optimization strategy that selects the solution based on the probability of being optimal by making it more balanced between the two aspects – exploration and exploitation [3].

The MCTS process in this model includes: 1) Selection: The nodes are chosen based on the Thompson Sampling, a search rule that calculates the distribution of the values according to them and chooses one action based on the expected reward (Eq. (7)), Where α_a and β_a are parameters of the Beta distribution, updated based on wins and losses. 2) Expansion: Note that as in the traditional MCTS, a new node will be added due to the novelty in the API, if the AI does not have the full knowledge of the environment. 3) Simulation: The tool works with a high-level policy that combines the Q-values provided by the reinforcement learning model. This is the manner the AI can model future outcomes in more precise ways. 4) Backpropagation: Results of their simulations feed into the model, and they not only refine the Thompson Sampling parameters and the Q-values but also reinforce the AI's facts of optimal manners with time. Therefore, the implementation of Double Q-learning with Thompson Sampling into MCTS makes it possible for the systems to operate more efficiently in Mahjong, especially within areas of the game that are ill-defined or are frequently in transition [3].

$$\theta_a \sim \mathrm{Beta}(\alpha_a, \beta_a) \quad (7)$$

3.3 Suphx AI Model

This model implements a deep learning approach combined with a novel evaluation function designed specifically for Mahjong. The model emphasizes deep reinforcement learning and employs a convolutional neural network (CNN) to process the game state[4].

3.3.1 Deep reinforcement learning framework

The core of the model is a deep Q-network (DQN), which uses a CNN to approximate the Q-value function[4]. The network takes as input a multi-channel representation of the Mahjong board, which comprises the following features: 1) Current Hand: The tiles in the hand of the AI, 2) Discards: The tiles are discarded by all players. 3) Potential Melds: Tiles that permit declaring a meld to occur. The CNN processes these inputs through multiple convolutional layers, extracting spatial features that represent the relationships between tiles. The output is a Q-value for each possible action, which the AI uses to make decisions[4]. The update rule for the deep Q-network follows the standard DQN framework:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\,[R_t + \gamma \max_a(s_{t+1}, a; \theta^-) - Q(s_t, a_t; \theta)] \quad (8)$$

where θ represents the parameters of the CNN, and θ^- represents the parameters of the target network, which are periodically updated to stabilize learning [4].

3.3.2 Novel evaluation function

The introduction to this evaluation criterion makes two main distinct entities - known scoring approaches mixed with artificial intelligence technology. The evaluation function unceremoniously does the following: 1) Traditional Scoring: Fundamental Mahjong gaming concepts and scoring principles. Fans have different scoring and discarding penalties that help give the game structure and scoring. 2) Machine-Learned Component: In the framework, a deep neural network utilizes a large training dataset of experts' gameplay and calculates the likelihood of winning in the current situation and the value of future states. This enables the system to make decisions based on the perceived strength. The evaluation function is defined as:

$$V(s_t) = \lambda \cdot V_{traditional}(s_t) + (1-\lambda) \cdot V_{learned}(s_t) \quad (9)$$

where λ is a weighting factor that balances traditional scoring with the machine-learned evaluation [4].

3.3.3 Training process

The model is trained with the help of two types of training: in-house gameplay and supervised learning. As a matter of fact, at this stage, CNN plays games based on data labels, where the details of the best player's case are described. The pre-training is followed by reinforcement learning introduced by self-play, bringing constant improvement to the models by playing against one's copies [4].

The AI with self-playing forwarded towards the darkened room, lightened by a gentle wave of thoughtful mindset. The unprecedented evaluation function can figure out which approach should the AI take based on each situation, giving the AI source of adaptability [4].

4. The AI Model of Aliyun Developer

Aliyun Developer AI model, which is detailed in the news article of the same source, ensures an outstanding performance against normal players. The Mahjong AI uses a unique blend of formal rules, machine learning, and simulation as the basis of the decision-making process [5].

4.1 Rule-Based Logic and Machine Learning

Rule-based logic is implemented in the game as predefining patterns that the program will use in high-score situations or whenever the player needs to play defensively. These principles are expressed in terms of decision start trees or as if-then clauses. However, for more complicated cases, the machine learning algorithm that has passed through the training is employed. The algorithm is trained on streams of Mahjong game data [5].

Focusing on the machine learning aspect, probably a supervised learning approach is used in such a way that the model is trained to learn from the expert experience where expert players will provide labeled data states in which they have computed the best moves. Three types of models that provide this function are decision trees, random forests, and even neural networks. The main aim is to discover the move that results in the highest score on the current board [5].

4.2 Simulation Techniques

For quick and quality decision-making, AI simulates the future states of a game. It means imagining all possible sequences of distribution of tiles and game flow, which helps to establish the right flow and choose the most beneficial path. The key difference between the simulation process and the MCTS is that the former may call for simpler heuristics compared to the latter because MCTS faces computational restrictions [5].

The program implies a potential instant elimination hand by checking the probability of its appearance, risking to allow a favorable tile distribution among the opponents. In Mahjong, offensive & defensive balance is treated as a substantial parameter of a play and may also be more important than ground strategy [5].

4.3 Performance Metrics

Its wins are only one of the features of AI and along with them, it is important how AI minimizes its losses and adapts itself to different opponents' play styles. The AI is created with a strong foundation that is extremely solid when dealing with all different game scenarios with hardly any human interventions. The AI's success in actual games shows the hybrid strategy effectiveness, which combines rule technology and advanced simulation and machine-learning models [5].

5. Problems and Lookahead to Future Development of Mahjong AI

Building AI models for Mahjong with the view of extracting insight can be met with complex challenges, which are twofold majorly because the game is complex naturally and also because it contains incomplete information.

5.1 Challenges

5.1.1 Handling incomplete information

Unlike games that have all the information on the cards at the same time as chess or Go, Mahjong contains unseen information: eg. the tiles of opponents' hands and what tiles will be drawn next. Thus, human AIs need to assess risks and make decisions based on the odds, which is a computation challenge and harder to find the best fit.

5.1.2 High dimensionality of the game state

One would always find enormous combinations of tiles, strings, and potential hands in Mahjong, and therefore dealing with Mahjong cannot be visualized in simple one- or two-dimensional spaces. The complexity of the game state, where the high dimensions are necessary to cover the state space, cannot be navigated by traditional algorithms, as they are too slow. Progress has been achieved in the utilization of advanced machine learning techniques, particularly deep learning and Monte Carlo Tree Search (MCTS), for faster navigation of the game space.

5.1.3 Sparse rewards and long-term strategy

Mahjong games are influenced by the long string of activities following, and these activities don't always lead to immediate benefits but rather it is building a general strategy. Describing a similar case, simply discarding a tile may not look like a good way at the beginning, yet it can be essential in real victory several moves ahead. Learning in reinforcement requires a vast amount of training data; usually, the sparse nature of reward makes it hard to find the optimal strategy for the task. Moreover, reinforcement learning usually needs some reward shaping to make it efficient [6].

5.1.4 Balancing offensive and defensive play

While playing Mahjong, the player needs to constantly consider the trade-off between increasing their score (offensive play) and hindering other players from winning (defensive play). Developing an AI system that can automatically switch between these strategies, based on the game situation, is a really difficult task and requires a deep understanding of the balance between the short and long-term risks of each strategy [6].

5.2 Future Directions in Mahjong AI Research

Future research will emphasize several key issues in which AI for Mahjong can be improved.

5.2.1 Improved probabilistic reasoning

Working intelligence's extended domain where research is aiming for sophisticated probabilistic reasoning under uncertainty will most likely become the latest research topic. That could involve the creation of a Bayesian system or other sophisticated probabilistic models that can provide a more accurate inference of the opponents' attitudes towards the risk or prospects by considering their real moves in the game [6].

5.2.2 Advances in deep learning architectures

The continuing improvements to deep learning practice, especially the formation of weaker-like models such as transformers, happen to be at the center of this process. These models may further be fine-tuned to tackle the large-scale state space of Mahjong more efficiently by allowing attention mechanisms to generate focus on relevant information in the game.

5.2.3 Hybrid models combining RL and MCTS

Based on future development, enhanced versions of the same system such as reinforcement learning and Monte Carlo Tree Search can be formulated. For example, new procedures may focus on employing MCTS to generate potential game states while what might happen at the same time in RL that uses simulations to improve decision-making. This new AI model can tap the strengths of both the exploration and exploitation process in complex games, such as Mahjong [6].

5.2.4 Enhanced reward shaping techniques

As a solution to the challenge of breaches in the rewards system, greater complex reward-shaping techniques will be designed to grant partial information about the consequences of each action to be taken. Future research could rely on machine learning models that forecast the future values for the different states or actions; this is supposed to allow the provision of continuous and relevant info to the RL for training the agents.

5.2.5 Computational efficiency and scalability

With the increase in complexity, Mahjong AI models may demand further study on efficient and scalable computation. Future works might delve into the utilization of parallel platforms, distributed computing, or even quantum computing technologies to carry out the scale where hefty simulations and transmogrification are necessary.

6. Conclusion

By considering all types of AI models in Mahjong, one can establish the achievement of redirecting economic game traits toward the strategic surroundings. The Tjong and Kanachan models, among causes and others reviewed, show the successful incorporation of complex decision flows into high-learning processes, especially via reinforcement learning and Monte Carlo Tree Search (MCTS). The learned AI navigates the fairly fine hairline between defense and offense and can manipulate hidden information and adjust strategies in the course of the game session.

Nevertheless, there are still problems being faced as a result of these successes. The huge and varied number of opportunities Mahjong gives is one of the factors that makes the task of AI optimization quite demanding and compels one to think out a long-term plan to do this. Artificial intelligence in Mahjong must be well-trained in probabilistic reasoning, selection of strategies that come to concrete game situations, and the use of transformer neural networks.

Future AI technology will not be exclusively devoted to gaming challenges, as lessons learned in developing AI for Mahjong will be disseminated and AI will be utilized in other interesting areas where uncertainty is present.

Also, these depths will upgrade the value of AI games and will advance the understanding of strategic process decision-making under complexity, uncertainty, and unpredictability.

References

[1] Li X, Liu B, Wei Z, Wang Z, Wu L. Tjong: a transformer-based mahjong AI via hierarchical decision-making and fan backward. CAAI Transactions on Intelligence Technology, 2024, 9.

[2] Cryolite. Kanachan: a Japanese (riichi) mahjong AI framework. GitHub, 2021, github.com/Cryolite/kanachan. Accessed 26 Aug. 2024.

[3] Mizukami N, Tsuruoka Y. Building a computer mahjong player based on Monte Carlo simulation and opponent models. In: 2015 IEEE Conference on Computational Intelligence and Games (CIG). IEEE, 2015: 275-283.

[4] Li J, Zhang Y, Liu Y, Wu H, Zhang S. Suphx: mastering mahjong with deep reinforcement learning. Microsoft Research, 2020, www.microsoft.com/en-us/research/publication/suphx-mastering-mahjong-with-deep-reinforcement-learning/. Accessed 26 Aug. 2024.

[5] Designing an AI mahjong program based on deep learning - Alibaba Cloud Developer Community. Aliyun.com, 2023, developer.aliyun.com/article/1163765. Accessed 26 Aug. 2024.

[6] Zheng Y, Li S. A Review of Mahjong AI Research. In Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI '20). Association for Computing Machinery, New York, NY, USA, 2020: 345–349.