Title	機械学習を用いた麻雀における相手の点数予測
Author(s)	萩原，涼太
Citation	
Issue Date	2016-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/13642
Rights	
Description	Supervisor：池田　心，情報科学研究科，修士

The Estimation of Opponent Score in Mahjong by Using Machine Learning

Ryota Hagihara (1410034)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 10, 2016

Keywords: Computer Mahjong, Machine Learning, Opponent Score Estimation.

Games have been developed as a genre of entertainment to give humans a lot of fun. Humans feel enjoyment with games in many ways, for example by competing against opponent players that have almost the same strength, by improving their skills with lessons, or by fighting against various types of tactics in the games. Most human players prefer to be matched against opponents that are almost as strong as them rather than opponents being too strong or too weak. Also, human players, especially in the case of beginners, tend to enjoy their improvement through some lessons to improve their play at the game. Nowadays, many computer players are used as teachers or enjoyable opponents instead of other human players.

Computer players as opponents or teachers for human players need to be strong to some extent. Thus, many researchers have researched about strong computer game players. Some computer players are already strong enough to win against human expert players in some perfect information games, such as Shogi or Go. Therefore, research about enjoyable computer players is now attracting more attention than before for those games.

On the other hand, there are imperfect information games in which strong computer players are still difficult to develop, for example Mahjong and Poker. One reason of this difficulty is that conventional Min-Max tree search cannot be applied naively in such games, contrary to perfect information games such as Shogi or Go. Thus, it is still a challenging problem to develop strong computer players in such imperfect information games.

In Mahjong, players should decide their strategies according to the global game situation, and they should avoid using short-sighted strategies focusing only on completing a *winning-hand* immediately. Strategies with long-term goals lead more surely to a win in the long run. For example, players sometimes give up completing a *winning-hand* if they can prevent another player from completing his *winning-hand*, leading more surely to a win in the end. For another example, they sometimes miss the chance of completing their *winning-hand* immediately on purpose in order to try to complete another form of *winning-hand* with a higher score. In such cases, the player goal is mainly to exceed their opponent scores at the end of the games.

Therefore, it is important to analyze game situations from multiple aspects and decide a strategy suitable to the whole aspects. To make such decisions, we need to approach multiple sub-problems required for the decision making, such as the problem of predicting opponents ' *winning tiles* or the problem of predicting the score that an opponent will obtain by completing his *winning hand*. After that, we can develop a computer player that make decisions considering various information obtained from those sub-problems.

We conducted experiments by applying machine learning methods for predicting the opponents winning score after the completion of their hand. We employed the same settings as Mizukami et al. for our experiments, that is, game records from an online Mahjong site " *Tenhoh* " as the learning data. We tried machine learning techniques with modified types of information as input by grouping game state features to enhance the generalization performance, while existing works applied machine learning with simpler forms of information as input. The grouping in our method is controlled by a Local Search method. Moreover, we adopt a machine learning approach with multiple layer neural networks using Deep Learning Tools on the same prediction problem, and we compared the performances.

At first, we describe the grouping of game state features with the Local Search method to achieve higher generalization performance. We applied the grouping of features to decrease the dimension of the state space and reduce the amount unnecessary information. Usually, game state features used in machine learning are selected manually before the machine learning. But sometimes it is difficult to decide the most suitable style of feature sets for the learning beforehand. For example, if we try to introduce a feature about the number of elapsed turns since the game started, we have multiple options. We can use the number as it is for the input, or we can use groups of numbers (say, " from 1 to 20 ", and " from 21 to 40 ") for the input. It is difficult to know beforehand which option is better. Furthermore, the best option might vary according to changes of the learning conditions. Therefore, we employed a method that tries many possible feature sets and suggests automatically a feature set that seems to be suitable for the learning conditions.

We did experiments about this grouping approach. We started by optimizing parameters in a linear combination model with the gradient descent method, without any grouping of the game features. As a result, we confirmed that the generalization performance improved, especially when using a larger set of training data. Next, we employed grouped game state features. The grouping is controlled by a local search method, and we obtained better generalization performance with this grouping. Additionally, we observed that the grouping of features occurred more frequently in the local search when the amount of training data is decreased. This observation implies that grouping of features is an effective approach, especially if we cannot prepare a lot of game records or if there is a too large number of game features. Moreover, we tested the performance with a combination of the features obtained after the grouping. Enhanced performance was obtained with the combination. The combination approach increases the number of feature sets used for machine learning, so this approach should be effective if we can prepare a large amount of training data. The larger number of feature sets created by the combination method do not harm the generalization performance if the amount of training data is sufficient.

Next, we explain about the attempt to enhance the generalizing performance by employing multi- layered neural networks for the machine learning. We evaluated the performance by comparing with the performance achieved by a simpler architecture.

In the experiments, firstly, we checked the generalization performance when changing the parameters of the multi-layered neural network, such as the number of training data, the number of hidden layers, or the number of hidden layer neurons. As the result, we confirmed that the performance improved with an increase of the number of hidden layers or the number of hidden layer neurons. Though, a too high number led to overfitting in the learning process. Secondly, we compared the generalization performances between the multi-layered network approach and the simple linear combination approach. The multi-layered network led to higher performance in most cases, but the linear combination worked better when the amount of training data is limited. Finally, we compared the performance obtained in this research with the performance obtained in an existing research by Mizukami et al. We confirmed that our approach achieved higher performance. Thus, we conclude that complex models are better for machine learning in the game of Mahjong rather than simpler models.