Title	機械学習を用いた囲碁の着手の日本語表現
Author(s)	宍戸，崇音
Citation	
Issue Date	2015-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/12629
Rights	
Description	Supervisor:池田　心，情報科学研究科，修士

Japanese expression of the move of Go by machine learning

SHISHIDO Takanari (1210026)

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

February 12, 2015

Keywords: Entertainment Computing, Go, Machine Learning, Japanese Expression, Specific Term.

For many years, the main challenge of computer Go research has been to make a “strong” program. In 2012, a strong computer program Zen won against a 9-dan professional player Takemiya Masaki, with a 4-stone handicap. It means that the strength of top programs is now sufficient for most amateur players. Then, as the next step of research, “entertaining” or “educating” computer Go players are attracting attention.

Usually, novice or beginner players study how to play Go through textbooks and/or games against stronger players. The stronger players, in many cases amateur players, know and can teach which moves are bad and which moves should be preferred instead. However, it should be noted that these skillful players are not necessary skillful teachers, then frequently their coaching is not adequate, for example too offensive to entertain beginners. Because there are only few people who can coach beginners well, and sometimes such coaching is expensive, then we consider that it is valuable to make a computer player that can entertain and educate human players.

As many abilities are needed for computer players to entertain or educate human players, one important element is “speaking” to do a review after a game or to do a chat during the game. Instead of just playing, such review or chat are a valuable complement for entertaining and educating human players. For example, when a beginner succeeds to exploit a given chance, he should be praised enough; or when the beginner player fails to select a good move, it should be pointed out to him during or after the game and possible variations after the move should be shown with adequate verbal explanations. In such explanations, moves of Go should be expressed by specific terms such as Tsuke or Hane, because it is very rare between human players to express the moves by their coordinates such as "(16,12) was bad, (16,13) or (17,11) were better". Usually the moves are expressed by using specific terms such as "Keima was bad, Tsuke or Kosumi were better".

In this thesis, our goal is to make the program able to label the moves with their associated specific terms. Such goal as obtaining a term from a state and an action can be done by hand coding, if the definitions of terms are clear and simple. However, it is well known that there are many pairs which cannot be clearly divided by explicit rules, such as “Magari and Osae”, “Nobi and Hiki”, or “Tsume and Hiraki”. Then, we employ a supervised learning technique to reduce the cost of hand coding and the risk of incorrect labelling.

First, 6 strong amateur Go players are asked to record for each move of some game records the corresponding specific term, from a pre-selected list of 71 terms. In this procedure, we consider that sometimes a move can be labelled with multiple terms, such as "this move can be Hane, and also Osae", so we allowed these players to record multiple terms as their first and second choice. About 11,000 moves were recorded. Some of the 71 authorized terms appear very frequently, for example over 1000 times, while on the other hand some terms appear only less than 10 times, even though they are very popular, such as "Geta". since it is difficult to learn such rare terms automatically, this is left as one future work.

Though basically 6 players recorded the moves of different games, one game (with 117 moves, relatively short) was shared, and all players recorded for it. The recorded terms are compared to know how well the terms for the same move match each other. The averaged matching ratio between two players is only 82.2% (87.0% when considering second choices). This result shows how vague and difficult such labeling is, especially though 5 of the 6 players are from the same community, the Go club of Kanazawa University.

Two existing programs, Tencho-Igo 5 and Nomitan are evaluated by using the human inputs, as these programs have a function to output the specific term for a move. Tencho-Igo 5 did not output anything for about 30% of the moves, which is not sufficient. Nomitan returned an output for all moves by using over 500 hand-coded rules, but the matching ratio (assuming the recorded terms are correct) is only 73.7% (76.6% when considering second choices). Then we consider that it is valuable to make a better program.

The design of features is very important for the performance of supervised learning. Poor features can deliver poor accuracy, and too rich features can cause overfitting. At first, we employ the set of features used in the rules of Nomitan, such as local patterns of stones, changes of liberties or distances to the edge of the board. As a learning model, J4.8 using decision tree is employed. The matching result (with 10-folding cross validation) is 75.3% (76.8% when considering second choices). This performance is slightly better than that of initial Nomitan, but worse than that between human players.

We investigate the reason of unsufficient performance, and find that performance about local shape terms is bad compared to that of initial Nomitan. Then, we add an enhancement using the rotation and reflection of local patterns, and also add some tuning of parameters of J4.8. Finally, the matching result reaches 82.0% (85.4% when considering second choices), which is very near the matching ratio between human players.

Finally, we let a professional 6-dan player evaluate our result. For about 500 moves (5 games), the terms recorded by human players and the terms outputted by our method are compared in a blind manner. The average of total evaluation score for terms recorded by human players is 84.4, and that for terms outputted by our method is 83.6. This means that our program can output good specific terms at the level of high amateur players.