TACTICAL AND STRATEGIC GAME PLAY IN DOPPELKOPF

DANIEL TEMPLETON

1. Abstract

The German card game of Doppelkopf is a complex game that involves both individual and team play and requires use of strategic and tactical reasoning, making it a challenging target for a computer solver. Building on previous work done with other related games, this paper is a survey of the viability of building a capable and efficient game solver for the game of Doppelkopf.

2. Introduction

Throughout human history, games have served an important role, allowing real life problems to be abstracted into a simplified environment where they can be explored and understood. Today, games continue to serve that role and are useful in a variety of fields of research and study, including machine learning and artificial intelligence. By researching ways to enable computers to solve the abstracted, stylized problems represented by games, researchers are creating solutions that can be applied directly to real world problems.

2.1. **Doppelkopf.** Doppelkopf is a game in the same family as Schafkopf and Skat played mostly in northern areas of Germany. The rules are officially defined by the Deutscher Doppelkopf Verband [1], but optional rules and local variants abound. The game is played with a pinochle deck, which includes two each of the nines, tens, jacks, queens, kings, and aces of all four suits, for a total of 48 cards. As in many games, like Skat, Schafkopf, Spades, Bridge, etc., the general goal is to win points by taking tricks, with each trick going to the highest card, trump or non-trump, played. As in Schafkopf, the highest trump card is determined not only by strict rank, but also by suit, e.g. the jack of clubs is a higher trump than the jack of spades. In addition to points won through taking tricks, a second tier point system rewards taking tricks in specific situations, often governed by optional rules.

The complexity of the game is further multiplied by partnering and soloing rules. In a regular (non-solo game), the four players are arranged in teams of two based on which players hold the two queens of clubs, although team membership is not revealed. Discovery

Date: 16 December 2011 12:00 PST.

of the team arrangements happens either through gameplay, when a player plays a queen of clubs, or through "calling", which amounts to accepting an increased scoring burden in order to declare team affiliation. (Special rules exist for the case when one player holds both queens of clubs, called a "Hochzeit" or wedding.) Until team membership has been definitively revealed, the game is effectively four players playing against each other. Once the team arrangements are known, proper team play commences. In some cases an in-between state may exist where only one player's team membership is known, offering its own unique gameplay dynamics. It's also worth noting that a player not holding a queen of clubs has no way to definitely signal his or her team membership other than by calling.

At the beginning of a hand, any player may opt to play a solo, meaning that the usual partnering rules are discarded, and that one player plays as a team of one against the remaining players as a team of three using a soloist-selected alternate trump arrangement. In an official tournament game, each player plays four non-solo hands and at least one solo hand.

2.2. **Related Research.** The complexity and regional popularity of Doppelkopf have presumably kept it from being a subject of mainstream machine learning research. A handful of desktop or online Doppelkopf games are available on the market that include Doppelkopf AIs. Taking the FreeDoko AI as a representative example, those AIs make use of well understood algorithms, such as decision trees and game heuristics. Related research in similar card games and large state model imperfect information games does provide a foundation for approaching this problem space from the perspective of machine learning.

Tsitsiklis and Van Roy [2] and Baird [3] lay important groundwork proving the soundness of applying reinforcement learning to problems with very large state spaces via approximation functions. Sturtevant and White [4] present an algorithm for playing Hearts that makes use of reinforcement learning with a perceptron as the approximation function, using the Max^n algorithm from Luckhardt and Irani[5]. Rather than reducing the state space through feature approximation, Sturtevant and White train against an extremely high dimensional space composed of boolean combinations of "atomic" features, with good results. Koller and Pfeffer [6] take a different approach to managing the large state space by reducing the complexity of the problem by restructuring the entire problem space around transition paths rather than states. Yet another approach, taken by Buro et al [7], is to leverage the sparse state model by translating states into indexes into a lookup table.

3. Implementation Approach

For this paper, the decision was made to apply reinforcement learning to a limited variant of Doppelkopf. In this Doppelkopf variant, solos are not allowed, and weddings result in a redeal. Other complex special cases of non-solo games, such as "Armut", or poverty, also result in a redeal. Calling is allowed only during the first trick, and its effect on scoring is ignored.

Because the state space for Doppelkopf is untenably large, as is the case with most other card games, an approach such as a vanilla Markov Decision Process, that attempts to have complete knowledge of the system could not be used. Instead, an approximation function is required to estimate the values represented by the states in the model. For this paper, the function used was simple linear regression.

As an avid Doppelkopf player, I was able to apply my understanding of game play to developing an approximate feature set that models the relevant details at any state in game. By reducing the specific cards to an approximate feature set, not only is the size of the state model reduced, but a level of game intuition is built into the learning algorithm. The final set of features included aspects of each player's hand, such as number of trump or number of aces, aspects of the cards already played in the trick, and aspects of the cards that are held by the other players.

For this paper, it was decided that the game of Doppelkopf would be approached as an imperfect information game, where each player only sees the cards in his hand and the cards played in the current and previous tricks. A side effect of this approach is that transitions in the state model become highly nondeterministic. For a game state $s \in S$ with player p set to play, the selection of a card c to be played can result in a very large number of possible subsequent states $s' \in S$.

Doppelkopf is a zero sum game. At the completion of a hand, two players will win the point value of the hand, and two players will lose the point value of the hand. (In a solo, the soloist will win or lose three times the point value of the hand to preserve the zero sum quality of the game.) One approach to reinforcement rewards would be to only issue rewards in the final trick. That approach tends to dilute some basic gameplay wisdom, such as valuing winning a trick with a fox in it. Instead, the approach to state rewards taken was to award points for each trick taken with bonus points for second tier scoring options, such as capturing or losing a fox. Because partnerships may not be known until late in the game, this reward scheme cannot be zero sum. A trick that is taken by a player before that player's partner is revealed must count as positive points only for the capturing player.

To jumpstart the learning process, an anonymous online "correspondence" Doppelkopf (or "Doko") site has donated records of more than a thousand games played online by four human players. Because the Doko site allows the customization of rules, and no single combination of rules represents a clear majority, one of the leading optional rule sets was selected, all games played with alternate rule sets were ignored during training. Because the training set data is from players of varying skill levels, the data is somewhat noisy. The hope is that the data noise should be minimal and would be outweighed by the demonstration of more subtle gameplay techniques like signaling and "hunting the fox."

3.1. **Implementation Details.** The first implementation step was to ingest and process the donated training data set. Provided as a SQL dump of roughly 2000 games, the data needed to be converted into a format amenable to the reinforcement learning algorithm. A process then had to be created to replay each game, card by card, tracking the full game details so that they could be used in training the approximation function.

After a game is played to completion, the game is then played back in reverse, allowing the expected values of each state along the way to be calculated and recorded in a straightforward fashion. Because of the choice to handle Doppelkopf as an imperfect information game, the state model is constructed based only on the data visible to a single player, i.e that player's cards and the cards thus far played by all players. In effect, each game in the training data set is processed as four separate training example games, one for each player's imperfect information state model.

As each state revisited, the state features and calculated expected value are stored for use as training data for the linear regression parameters. After the entire set of donated games is processed, the linear regression parameters are trained against the entire data set. The resulting parameters are make logical sense for the most part. The features that most strongly correlate to a high value trick (from a given player's perspective) are the number of "Dullen" (the highest trump) held, and whether the player's partner is winning the trick. Both are clearly good indicators of expected success. Oddly, the next highest indicator of high value trick is the number of foxes held by the player, which is counterintuitive at first glance. Winning a trick that contains a fox played by the opposing team results in additional second tier points for the winning team. Because the training data is drawn from games played by experienced players, the risk of losing the fox may be mitigated and even turned to an advantage by smart and careful use of the card.

As the training set is very small compared to the state space, the linear regression parameters are only trained by the training data against a tiny percentage of the possible states. At this stage, the algorithm is therefore a pretty poor Doppelkopf player. To provide additional training data, four copies of the solver are set to play against each other in groups of 10,000 games, producing 480,000 training data elements for each batch of games. After each batch of games, the parameters are trained against the new data combined with the previous data. The parameters thus trained appear to better match with the expected relevance of the state features. The number of Dullen and whether the player's partner is winning the trick are still strong indicators, but the number of foxes is a negative valued feature. In a system that has not yet learned how to properly play such a card, it very logically represents more of a risk than a value.

Beyond the first batch of self-play games, the linear regression parameters change very little, indicating that the regression model has learned as much as it can from the data. Unfortunately, the value approximation of the linear regression is still quite poor, resulting in a poor Doppelkopf player. The algorithm remains only slightly better than a player that selects cards at random.

Given that the trained parameters seem to match logical expectations of relative magnitude and sign for feature relevance, my theory for the poor performance of the algorithm is that the approach is too simple for a game as complex as Doppelkopf.

4. Conclusions and Future Work

While failing to produce an efficient expert-level Doppelkopf solver is a disappointing result, it is not at all unexpected given the complexity of the problem and the limited time and resources available for the project work. As the opportunity presents itself to continue this project in subsequent coursework, the next steps will be to explore significantly different feature representations for game states and to investigate the use of other variants of reinforcement learning, such as $TD(\lambda)$. I have found this project to be exciting and challenging and look forward to the opportunity to develop it further.

5. References

[1] Deutscher Doppelkopf Verein. Turnierspielregeln. "http://www.doko-verband.de/download/turnierspielregeln.rtf", 2003.

[2] J. N. Tsitsiklis and B. Van Roy, An Analysis of Temporal-Difference Learning with Function Approximation. In *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674-690, 1997.

[3] L.C.Baird,Residual algorithms: Reinforcement learning with function approximation, in *Machine Learning: Proceedings 12th Int. Conf.*, July 912, Prieditis and Russell, Eds. San Francisco, CA: Morgan Kaufman, 1995.

[4] N.R.Sturtevant and A.M.White. "Feature Construction for Reinforcement Learning in Hearts." In *5th International Conference on Computers and Games ICCG*, 2006

[5] C.A.Luckhardt and K.B.Irani, "An Algorithmic Solution of N-Person Games." In *AAAI-86 Proceedings*, 1986

[6] D.Koller and A.Pfeffer. "Generating and solving imperfect information games." In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1185-1192, 1995

[7] M. Buro, J. R. Long, T. Furtak, and N. Sturtevant. "Improving state evaluation, inference, and search in trick-based card games." In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009