Winning the game Rummikub using () algorithm

MATH 818.01 Midterm Survey

Jiyoon Jeong

Abstract

This survey presents an overview of the artificial intelligence that aims to win the board game, Rummikub. We discuss the topological, linear-algebraic, and combinatorial aspects of Tverberg's theorem and its applications. The survey contains several open problems and conjectures.

1 Introduction

본문에서도 당연히 한글 사용 가능합니다..만, CJK 환경으로 지정해야 합니다. 텍 소스를 보세요.

Rummikub is a number-based board game. The players do not know

Theorem 1 (Helge Tverberg 1966 [**?**]). *Given* $(r-1)(d+1)+1$ *points in* $\mathbb{R}^d$*, there is a partition of them into* r *parts whose convex hulls intersect.*

The paper is organized as follows. In section 2 we introduce basic concepts prior to explaining the types of game in terms of complete. Next, in sections 3, 4, we describe case studies of AlphaGo and AlphaZero developed by Google Deepmind. In Section 5, we introduce integer linear programming (ILP) to see implementation of Rummikub using ILP in Section 7.

2 Complete and Incomplete Information Game

There are diverse classification of games–zero-sum and non-zero-sum game, perfect and imperfect information game, and n-person game to name a few. Especially, in terms of information In order to

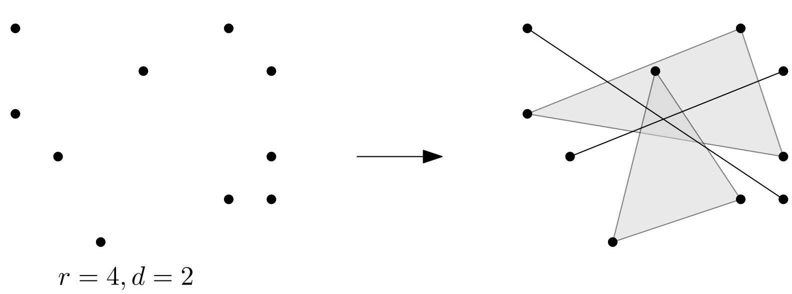


Figure 1: An example of a Tverberg partition. The partition is not unique.

make a program that wins the game Rummikub, it is meaningful to look over prior researches of similar or opposite kinds of game. First, the game Rummikub is classified into an incomplete information game which is an opposite concept of a complete information game.

Definition. Utility function Utility function represents the degree of satisfaction of an input, which consists of various alternatives. Hence it reveals the preferences of the alternatives. The value of an output is to be maximized.

Consider a utility function of a Prisoner's Dilemma.

The set of players is $\{P_1, P_2\}$.

The players are going to choose whether one will confess or deny, so there are two actions for each player, put it $A_1 = \{C, D\}, A_2 = \{C, D\}$ respectively.

If both players choose to confess, then they go to prison for 3 years. If both choose to deny, then they go to prison for 1 year each. If one confesses and the other denies, then the former goes free and the latter go to prison for 4 years. This is described on the table.

The set of outcomes is $\{(C, C), (D, D), (C, D), (D, C)\}$.

In one player's point of view, s/he do not know what the other player will choose, but knows that the other player is offered the same deal as written below. Prisoner's Dilemma is incomplete information game because the players do not know how the other player will act. A definition of incomplete information game will be introduced soon.

P1 / P2	Confess	Deny
Confess	(-3,-3)	(0,-4)
Deny	(-4,0)	(-1,-1)

The utility of the player P_i is $u_i(a_i, a_{-i})$ where a_i denotes the action of a player Pi, and a_{-i} denotes the action of the other player. Hence the result is as follows: $u_i(C, C) = -3, u_i(D, D) = -1, u_i(C, D) = 0, u_i(D, C) = -4$.

Information is common knowledge if it is known to all the players, if each player knows that all the players know it, if each player knows that all the players know that all the players know it, and so forth ad infinitum.

Complete information came is a game where each player is fully aware of the rules of the game and the utility functions of each of the players. That is, each player has the common knowledge.

If we look over the meaning of the concept 'complete' itself, we can say as the following. Complete is a nature not moving first, or her initial move is observed by every player In a game of incomplete information, Nature moves first and is unobserved by at least one of the players. Otherwise the game is one of complete information. In contrast, an incomplete information refers to situations where some of the elements of the game are not common knowledge

The concepts explained so far can also be demonstrated in mathematical language and this is dealt in many books of Game Theory, as such, reference

3 Case Study: AlphaGo

As seen in Section 2, the game of Go is a complete information game.

AlphaGo is a program consists of Deep Neural Network and Monte Carlo Tree Search.

List of the methods: Supervised Learning of policy network, Reinforcement Learning of policy network, Reinforcement Learning of value network, Searching with policy/value network (which is Monte Carlo Tree Search).

DNN is trained by humans and by itself to make AlphaGo a smarter player and MCTS searches the best move and choose it. Policy network and value network are improved by the training process of DNN and rollouts are not trained but built previously by human to be used in a search algorithm.

Policy network chooses the best next move. Move changes the current state by choosing some action, and turn it into the next state. Both Supervised Learning and Reinforcement Learning is used to improve policy network–that is, to make policy network smarter hence it can choose the best move. SL is done by human experts while RL is done by self-plays.

Value network calculates the probability of final win when the move is suggested by policy network.

Rollouts–or Monte Carlo rollouts for now–are Monte Carlo simulations, in which random search steps are performed without branching until a solution has been found or a maximum depth is reached.

See the figure

3.1 Deep Convolutional Neural Network

Supervised learning?

3.2 Reinforcement Learning

3.3 Monte Carlo Tree Search (MCTS)

4 Case Study: AlphaZero

alpha-beta pruning

5 Integer Linear Programming

Linear Programming (LP), or linear optimization, is an optimization problem in which the objective function is linear in the unknowns and the constraints consist of linear equalities and linear inequalities. LP is not a computer-based implementation, but only a mathematical method.

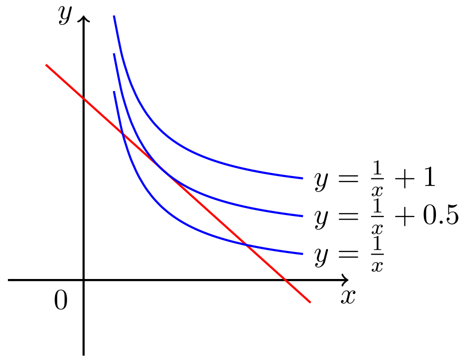
Optimization includes maximizing or minimizing a linear functional over a set of constraint polynomials. Linear equations can be reduced to a vector and matrix notation.

$\mathbf{A}$ is an $m \times n$ matrix, $\mathbf{x}, \mathbf{c} \in \mathbb{R}^{\mathbf{n}}$, $\mathbf{b} \in \mathbb{R}^{\mathbf{m}}$

$$\text{minimize } \mathbf{c^T x} \qquad \text{subject to } \mathbf{Ax} = \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}. \tag{1}$$

Integer linear programming is a linear programming whose each entry of solution vector is integer.

Maximize the value of $y - \frac{1}{x}$ on the region surrounded by $x \geq 0, y \geq 0, y \leq -0.9x + 2.4$



Put the function of purpose $k = y - \frac{1}{x}$. $y = \frac{1}{x} + k$, $y = -0.9x + 2.4$

$k = y - \frac{1}{x}$ has the form of graph as illustrated as blue graphs. The graph should be on the region, or some point on the graph should also be on the region. Combining these two conditions, we can conclude that in order to maximize k, the graph should be as high as possible about y-axis, meeting the region. Since $y = \frac{1}{x} + k$ has derivative $y' = -\frac{1}{x^2}$ on $(0, \infty)$, we can conclude that there exists some (x, y) such that $y' = -0.9$. That is, there exists k such that $y = \frac{1}{x} + k$ tangent to $y = -0.9x + 2.4$. By calculations, we can get that two graphs meet at $(\frac{\sqrt{10}}{3}, -\frac{3\sqrt{10}}{10} + 2.4)$. Then we conclude that maximum $k = \frac{19}{20} + 2.4$

We will see specific implementation of ILP in Section 7

6 Rules of the game Rummikub

There are total 106 tiles–2 sets of tiles numbered 1 to 13 in four colors: black, red, blue, orange; 2 Jokers. There are 4 racks that the players can put their tiles on, hence 2 to 4 players can join the game. meld A set is Each players has a turn and when it is one's turn, that player give out sets or A group is a set of either three or four tiles of the same number in different colors. A run is a set of three or more consecutive numbers, all in the same color.

Scoring

7 ILP on Rummikub

There are two goals of this programming:

(1) The major goal is to maximize the number or value of the tiles that can be placed onto the table.

(2) The minor goal is to minimize the movement of the existing sets on the table.
It seems like this programming is a greedy algorithm if we only look over the major goal. However, practical eff

8 Implementation