

Поиск лучшего алгоритма для игры в "дуэльного Дурака"

Лескевич Даниил
leskevich.dk@phystech.edu

Project Proposal

Создание алгоритмов для игры в "дуэльного Дурака" (партия 1 на 1) и выявление наиболее значимых эвристик.

1 Идея

Проект заключается в том, чтобы написать много различных эвристик по оценке состояния следуя которым боты будут играть в "дуэльного Дурака". После чего выявить оптимальные гиперпараметры этих эвристик.

1.1 Problem

"дуэльный Дурак" довольно интересен, как игра за счёт того, что его можно разделить на несколько этапов.

1) Он длится от начала партии и до того момента, пока в колоде ещё есть карты. Здесь игра является игрой с неполной информацией, и с каждым ходом каждый игрок получает всё больше информации, пока не наступит этап 2). При этом "информированность" (если можно так сказать, о количестве известных карт) растёт неодинаково для двоих игроков. Так если игрок В не смог отбить карту x игрока А, то игрок В узнал только то, что x больше не может быть в колоде и не получил никакой информации о руке игрока А, в то время как игрок А взял ещё одну карту y из колоды и знает, что у игрока Б есть карта x на руке.

Энтропия [1]: степень неопределённости будет характеризоваться энтропией

$$H = \sum_{i=1}^k p_i \log_2 p_i \quad (1)$$

$p_i = \frac{1}{C_{N-k}^k}$, где k , это количество карт о которых у игрока нет информации

2) После того, как карты в колоде закончились игра становится игрой с полной информацией. Поэтому на этом этапе игры можно найти последовательность ходов, которая приведёт к победе. Этот этап будет решаться полным перебором всевозможных действий игроков и выбором победного.

Запишем концептуальный вид итогового алгоритма нашей программы псевдокодом.

Создаем таблицу А

Пункт1. Формируем таблицу В из ботов, где каждый следующий имеет средний процент побед по таблице А с остальными ботами $> p$.

Пункт2. Отбираем n ботов из таблицы В, которые имеют наивысший процент против ботов из таблицы А.

Теперь говорим, что таблица А это наши n ботов, а таблицу В удаляем и возвращаемся к Пункту1

Максимизируемой метрикой является процент побед алгоритма в игре с другими. За счёт того, что боты в таблице на каждой итерации становятся сильнее всё сильнее, в итоге мы получим таблицу самых сильных, критерием остановки можно считать, то что на новой итерации таблица А не изменяется.

Algorithm 1 Турнир

```
1: Initialize evristic  $\leftarrow getRandomEvristic(n = n)$  n случайных наборов эвристик
2: Initialize tournamentTable  $\leftarrow Table$ 
3: Initialize newTournamentTable  $\leftarrow Table$ 
4: for  $i = 0, 1, \dots, n - 1$  do
5:   tournamentTable.player.append(player(evristic[i]))
6: end for
7: —
8: def objectfunction(trail, tournamentTable, newTournamentTable):
9:   Initialize evristic  $\leftarrow getAllSetsEvristic()$  всевозможный набор эвристик для оптимизации
10:  Initialize player1 = player(evristic)
11:  Initialize winRate = 0
12:  for  $i = 0, 1, \dots, len(tournamentTable)$  do
13:    win1, win2 = game(player1 = player1, player2 = tournamentTable.player[i], numGame = 100)
14:  end for
15:    winRate+ =  $\frac{win1}{100}$ 
16:  if  $\frac{winRate}{len(tournamentTable)} > p$  : then
17:    newTournamentTable.append(player1)
18:  end if
19:    metric =  $\frac{winRate}{len(tournamentTable)}$ 
20:    return metric
21: —
22: study = optuna.createStudy(direction = 'maximize')
23: for  $i = 0, 1, \dots, k$  do
24:   study.optimize(objectfunction, nTrials)
25:   tournamentTable = n ботов с наивысшим winRate из newTournamentTable
26:   Initialize newTournamentTable  $\leftarrow Table$ 
27: end for
```

2 Outcomes

Набор алгоритмов с лучшими эвристиками способных играть в "Дуэльного Дурака" и сам набор оптимальных эвристик.

3 Литературный обзор

В работе [1] есть формула степени неопределённости для каждого игрока:

$$H = \sum_{i=1}^k p_i \log_2 p_i \quad (2)$$

$p_i = \frac{1}{C_{N-k}^k}$, где k, это количество карт о которых у игрока нет информации

Ещё там приведён пример простейшей машины состояний для бота.

Так же сильно повлияла работа [2], в ней высказана идея о нескольких фазах игры и том, что цели в каждой из фаз отличаются. Это фундаментальная работа [3], где приведено более детальное описание игры с точки зрения энтропии и ключевые типы игроков (это очень важно для того, чтобы алгоритмы хорошо играли против реальных людей). Статьи [4], [5] приводят различные эвристики следуя которым можно оценить текущее положение игрока.

Тут [6], [7] приведены правила игры и идея для программной реализации

4 Метрики качества

1) Метрикой качества будет процент побед алгоритма в игре с другими алгоритмами. И если будет достаточно данных по его игре с реальными людьми, то это будет второй метрикой, которая должна подтвердить или опровергнуть правильность подбора эвристик.

5 Примерный план

- К 22.04.2022 написать игру и правила
- К 29.04.2022 выбрать и закодировать эвристики по которым будем оптимизироваться
- К 05.05.2022 просвети поиск оптимальных параметров.
- Если успею 12.05.2022 добавить графический интерфейс, чтобы человек смог удобно сыграть с лучшим алгоритмом.
- Если успею заставить к человека сыграть m партий с ботом, чтобы вычислить вторую метрику (процент побед при игре с людьми).

References

- [1] И. В.Трушин А. Ф. Ляхов. Алгоритмы и программы управления компьютером в азартных играх, созданные на основе теории нечётких множеств. 2009.
- [2] AI для «Дурака». <https://habr.com/ru/post/263259/>. 2015.
- [3] А. Ф. Ляхов. Информационный анализ азартных игр. http://m.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=mo&paperid=445&option_lang=rus, pages 32–41, 2006.
- [4] Игра (не) для дураков. Пишем AI для «Дурака» (часть 1). <https://habr.com/ru/post/437346/>. 2019.
- [5] И. В.Трушин А. Ф. Ляхов. Компьютерное моделирование поведения игрока в интеллектуальной карточной игре с помощью нейронной сети. 2013.
- [6] Карточная игра «Дурак» на двух M5Stack. <https://habr.com/ru/post/492652/>. 2020.
- [7] Дурак (карточная игра). <https://inlnk.ru/poew0v>.