Poker Project - Team 15

Gong Changda, Kong Zijin, Liu Yingnan, Liu Yiyang, Yang Sihan
National University of Singapore
first,second,third,fourth@u.nus.edu

1 Introduction

This project implements a poker player AI agent that acts as one of the two players in Limit Texas Hold'em. The fundamental idea of implementation is using reinforcement learning to build a heuristic evaluation function, which measures the expected reward of each action and make the decision that maximizes the return. Hand strength, current bet and opponent's bet were the three components that make up the evaluation function. The evaluation function is a linear function that sum up the product of the numerical value of each component and their corresponding weight. To decide the weight of the three components, online training was conducted. The weight for each component reflects the expected influence that the component has on the final result of one round of game. The agent's decision and final result were feed into the online training at the end of every round, and weight for each component was adjusted. Different from current bet and opponent's bet which can be obtained directly from game state, the numerical value of hand strength was computed by evaluating hole cards and revealed community cards. A list of poker features, such as high card and pair value was selected to determine the hand strength. Similar to the evaluation function, the function that calculates hand strength is also a linear function that sums up the value of the product of weight and feature. The weight of each feature was determined from our offline training. Traditional approach of poker game agent, like listing preflop chart, is computationally expensive and space consuming. To improve the efficiency while not reducing accuracy, we choose to implement a reinforcement learning based agent. Our agent is able to learn from opponent's behavior and adjust evaluation function according to opponent's characteristic. To improve the accuracy of decision making, we trained the hand strength evaluation function of agent offline with large amount of data before playing game with others. In this report, we will explain how we implement the poker agent and discuss how the agent can make optimal decisions. In section 2, we will review a few approaches in past research and analysis their strength and weaknesses. In section 3, we will elaborate our agent's play strategy, and explain the implementation details and training methods. In section 4, we will discuss the training results and show merits and limitation of our approach. Lastly, we will go into the conclusion and future direction.

2 Past research

The core of the implementation of an AI poker player agent is to decide the next action it should take giving the information of the environment. Utility maximization and regret minimization are two approaches that have been well explored regarding next action decision. The most traditional method to optimize the utility is to list out results for all possible solutions in each round and select the action that gives highest utility. This method is precise and easy to implement but computational expansive. To reduce the computational effort, some researchers proposed heuristics that evaluate the return of the game and use decision trees to model the next actions that should be take. With sufficient and reasonable training, this method can yield good results. Regret minimization is another popular approach. Regret is defined to be the loss in utility taking this action suffers for not having selected the single best deterministic strategy and the single best deterministic strategy can only be known in hindsight. The approach aims to minimize the regret value. In 2015, Bowling and other 3 scientists announced that they weakly solved Heads-up limit Texas hold'em by using a technique called counterfactual regret minimization which is based on regret minimization. Since we discussed minimax algorithm during class and we are more familiar with the utility maximization approach, we adopted the utility maximization approach. We generate a decision tree at each step and the agent will take the action that gives the highest numeric value of the evaluation function. Evaluation function is another key element of our implementation. From past research, hand strength and pot money are two of the most widely-used factors when constructing the evaluation function. Reinforced learning is also proposed by some researchers to adjust the evaluation function based on information that cannot be got in advance, such as opponent's behavior.

3 Implementation

3.1 Agent strategy

To play Limit Taxas Hold'em Game, our poker agent evaluates the environment and estimates expected return of actions to make decisions.

1. When it's the poker agent's turn to bet, the poker agent will firstly evaluate the strength of hole cards combining

revealed community cards. The hand strength is evaluated based on existing card features in hand and current betting round. The poker agent takes 18 cards features into consideration. Hand strength evaluation function is a linear combination of those 18 card feature's value. While playing, agent uses the function settled by offline training before game to estimate the strength of cards. The function and offline training will be further elaborated in the following section.

2. After evaluating hand strength, agent will estimates the expected reward of the whole round of game if choosing each action ('call', 'raise' or 'fold'). The expected reward is estimated by evaluation function, which is a linear combination of hand strength, the difference between the current bet and the opponent's bet, the difference between the current remaining money and the money in the opponent's stack, and the difference between my gain and the opponent's gain:

3. What is the result that you achieved? (i.e. why should the reader believe you?)

3.2 Evaluation function

Generally speaking, the hand strength of the combination of hole cards and revealed community card is static. The evaluation function of hand strength is obtained by offline training and settled before the start of game. It is not affected by opponents' behavior and will not be adjusted in the process of playing game. Hand strength evaluation function is a linear combination of all features appearing in the combination of hole cards and revealed community card(s). There are 18 features that our agent use to estimate hand strength.
All the 18 card features of 4 types are shown below.

3.3 Training method

Offline training

Online training

To enable our agent to learn from games that it have been played with the opponent, the weight for each feature in the evaluation function is adjusted to improve the accuracy of our evaluation. As mentioned in section 3.2, our evaluation function for each action calculate the expected reward of taking the action. Based on the result at the end of every round, the value of the weight vector, θ will be adjusted.
At the first round, pre-hardcoded base weight is used in the evaluation function. The base weight was derived from our offline training with random player (?, not sure). Monte Carlo algorithm was used.
At the end of each round, θ for each action will be updated inside

```
receive_round_result_messge()
```

function. Stochastic gradient descent approach is used to update θ.

4 Training Result and analysis

4.1 Offline training result

4.2 Online training result

4.3 discussion

Our implementation has three main advantages. Firstly, the approach we used to solve this problem is intuitive and fits the characteristic of poker game. Instead of generating game tree and running minimax algorithm to solve the problem, we use decision tree. The restriction of minimax algorithm in imperfect information game is mentioned by several researchers. Decision tree, in the other hand, requires less space and simulate well people's behavior in real life. Secondly, we took a wide range of features into consideration which makes our evaluation more complete and precise. Thirdly, by using reinforcement learning, we adjust the weight of each feature promptly.

5 Conclusion

Print manuscripts two columns to a page, in the manner in which these instructions are printed. The exact dimensions for pages are:

- left and right margins: .75″
- column width: 3.375″
- gap between columns: .25″
- top margin—first page: 1.375″
- top margin—other pages: .75″
- bottom margin: 1.25″
- column height—first page: 6.625″
- column height—other pages: 9″

All measurements assume an 8-1/2″ × 11″ page size. For A4-size paper, use the given top and left margins, column width, height, and gap, and modify the bottom and right margins as necessary.

5.1 Format of Electronic Manuscript

For the production of the electronic manuscript, you must use Adobe's *Portable Document Format* (PDF). A PDF file can be generated, for instance, on Unix systems using `ps2pdf` or on Windows systems using Adobe's Distiller. There is also a website with free software and conversion services: `http://www.ps2pdf.com/`. For reasons of uniformity, use of Adobe's *Times Roman* font is strongly suggested. In LaTeX2e, this is accomplished by putting

```
\usepackage{times}
```

in the preamble.[1]

Additionally, it is of utmost importance to specify the American **letter** format (corresponding to 8-1/2″ × 11″) when formatting the paper. When working with `dvips`, for instance, one should specify `-t letter`.

[1] You may want also to use the package `latexsym`, which defines all symbols known from the old LaTeX version.

5.2 Title and Author Information

Center the title on the entire width of the page in a 14-point bold font. The title should be capitalized using Title Case. Below it, center author name(s) in a 12-point bold font. On the following line(s) place the affiliations, each affiliation on its own line using a 12-point regular font. Matching between authors and affiliations can be done using superindices. Additionally, a comma-separated email addresses list using a 12-point regular font is also allowed. Credit to a sponsoring agency can appear on the first page as a footnote.

5.3 Text

The main body of the text immediately follows the abstract. Use 10-point type in a clear, readable font with 1-point leading (10 on 11).

Indent when starting a new paragraph, except after major headings.

5.4 Headings and Sections

When necessary, headings should be used to separate major sections of your paper. (These instructions use many headings to demonstrate their appearance; your paper should have fewer headings.). All headings should be capitalized using Title Case.

Section Headings

Print section headings in 12-point bold type in the style shown in these instructions. Leave a blank space of approximately 10 points above and 4 points below section headings. Number sections with arabic numerals.

Subsection Headings

Print subsection headings in 11-point bold type. Leave a blank space of approximately 8 points above and 3 points below subsection headings. Number subsections with the section number and the subsection number (in arabic numerals) separated by a period.

Subsubsection Headings

Print subsubsection headings in 10-point bold type. Leave a blank space of approximately 6 points above subsubsection headings. Do not number subsubsections.

Special Sections

You may include an unnumbered acknowledgments section, including acknowledgments of help from colleagues.

Any appendices directly follow the text and look like sections, except that they are numbered with capital letters instead of arabic numerals.

The references section is headed "References," printed in the same style as a section heading but without a number. A sample list of references is given at the end of these instructions. Use a consistent format for references, such as that provided by BibTeX. The reference list should not include unpublished work.

5.5 Citations

Citations within the text should include the author's last name and the year of publication, for example [Gottlob, 1992]. Append lowercase letters to the year in cases of ambiguity. Treat multiple authors as in the following examples: [Abelson *et al.*, 1985] or [Baumgartner *et al.*, 2001] (for more than two authors) and [Brachman and Schmolze, 1985] (for two authors). If the author portion of a citation is obvious, omit it, e.g., Nebel [2000]. Collapse multiple citations as follows: [Gottlob *et al.*, 2002; Levesque, 1984a].

5.6 Footnotes

Place footnotes at the bottom of the page in a 9-point font. Refer to them with superscript numbers.[2] Separate them from the text by a short line.[3] Avoid footnotes as much as possible; they interrupt the flow of the text.

6 Illustrations

Place all illustrations (figures, drawings, tables, and photographs) throughout the paper at the places where they are first discussed, rather than at the end of the paper. If placed at the bottom or top of a page, illustrations may run across both columns.

Illustrations must be rendered electronically or scanned and placed directly in your document. All illustrations should be in black and white, as color illustrations may cause problems. Line weights should be 1/2-point or thicker. Avoid screens and superimposing type on patterns as these effects may not reproduce well.

Number illustrations sequentially. Use references of the following form: Figure 1, Table 2, etc. Place illustration numbers and captions under illustrations. Leave a margin of 1/4-inch around the area covered by the illustration and caption. Use 9-point type for captions, labels, and other text in illustrations.

Acknowledgments

The preparation of these instructions and the LaTeX and BibTeX files that implement them was supported by Schlumberger Palo Alto Research, AT&T Bell Laboratories, and Morgan Kaufmann Publishers. Preparation of the Microsoft Word file was supported by IJCAI. An early version of this document was created by Shirley Jowell and Peter F. Patel-Schneider. It was subsequently modified by Jennifer Ballentine and Thomas Dean, Bernhard Nebel, and Daniel Pagenstecher. These instructions are the same as the ones for IJCAI–05, prepared by Kurt Steinkraus, Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Lab.

A LaTeX and Word Style Files

The LaTeX and Word style files are available on the IJCAI–18 website, `http://www.ijcai-18.org/`. These style files implement the formatting instructions in this document.

The LaTeX files are `ijcai18.sty` and `ijcai18.tex`, and the BibTeX files are `named.bst` and `ijcai18.bib`. The LaTeX style file is for version 2e of LaTeX, and the BibTeX style file is for version 0.99c of BibTeX (*not* version 0.98i).

[2] This is how your footnotes should appear.

[3] Note the line separating these footnotes from the text.

The `ijcai18.sty` file is the same as the `ijcai07.sty` file used for IJCAI–07.

The Microsoft Word style file consists of a single file, `ijcai18.doc`. This template is the same as the one used for IJCAI–07.

These Microsoft Word and LaTeX files contain the source of the present document and may serve as a formatting sample.

References

[Abelson *et al.*, 1985] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, Massachusetts, 1985.

[Baumgartner *et al.*, 2001] Robert Baumgartner, Georg Gottlob, and Sergio Flesca. Visual information extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Databases*, pages 119–128, Rome, Italy, September 2001. Morgan Kaufmann.

[Brachman and Schmolze, 1985] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, April–June 1985.

[Gottlob *et al.*, 2002] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences*, 64(3):579–627, May 2002.

[Gottlob, 1992] Georg Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, June 1992.

[Levesque, 1984a] Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23(2):155–212, July 1984.

[Levesque, 1984b] Hector J. Levesque. A logic of implicit and explicit belief. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, pages 198–202, Austin, Texas, August 1984. American Association for Artificial Intelligence.

[Nebel, 2000] Bernhard Nebel. On the compilability and expressive power of propositional planning formalisms. *Journal of Artificial Intelligence Research*, 12:271–315, 2000.