

Foundations of Artificial Intelligence Final Project

b09902137 簡尉棠

一、我使用的方法：

方法一：利用Linear Regression建模得到分數，並依此進行評估

首先我先嘗試根據從現在看到的牌去推出達到各種牌型的機率，如hole card = ['SA', 'ST']以及community card = ['S2', 'H3', 'D5']這樣的情況下，我最後能夠變成皇家同花順的機率、同花順的機率、鐵支的機率……，因共有十種牌型，故我可以得到十個機率，再分別乘上牌型的倍率，便可以得到現在這個情勢我的分數。根據此分數去進行fold, raise, 以及call的判斷。

在這樣的前提下，我會需要的有每一種牌型的分數 $v = (v_1, v_2, v_3, \dots, v_{10})$ ，再與我計算出來的機率 $p = (p_1, p_2, p_3, \dots, p_{10})$ 進行內積便可得到我的分數，機率 p 的部分可以透過程式分析現有卡牌搭配排列組合進行計算而得，但 v 無法透過簡單的數學計算而得。原本我打算使用Linear regression的方式去建模型，對於 X_{test} 所對應的 Y_{test} 目前來說是不存在的，因此我寫了一份能夠比較牌的大小的程式去對輸入的牌進行比較大小，利用random的生成牌型去比較大小，但即使這樣做仍然沒辦法得到一個好的 Y_{test} ，因此無法使用這個方法。所以我換了下面的方法。

方法二：純粹使用贏牌機率進行估算

在Preflop階段，每位玩家僅能看到2張牌，並且這兩張牌的好壞與否，我認為會很大程度的影響最終結果，因此我希望在這邊能夠藉由跑更多場所得到的勝場數去計算勝率會更準，但因為每步只能計算五秒，因此我在此階段採取建表的方式，因起手牌型不多且花色不拘，我細分為以下四項進行計算勝率：

一對(pair)

同花(flush)

有機會成為順子(但不同花)

完全不相關的兩張牌

且將最後得到的結果建成表格放在tbe.py的檔案裏面。當在Preflop階段時，能夠直接查表得知勝率約略為何。

在The Flop、The turn、The River階段，此時再用建表的方式其表格會過於巨大，因此我在這邊利用約莫3.6秒的時間讓電腦在這樣的條件下進行 $10^6 \times 10000$ 局並計算勝率後回傳。

根據回傳的勝率進行判斷，若勝率低於fold_thres的話便蓋牌，高於fold_thres但低於call_thres的話就call，否則就根據p的大小進行raise。由於賽制問題，當我測試到若接下來的回合全部蓋牌仍然不會輸的時候，我就永遠選擇蓋牌以確保勝率。

二、

fold_thres = 0.15 ; call_thres = 0.65

當對面raise高於300的時候，判斷對面牌有一定程度的實力，此時要高於call_thres才會繼續遊戲，否則蓋牌。

三、方法比較、結論分析：

由於方法一沒有得出結果，因此只能分析和baseline的對戰結果。

我認為現階段對對手的牌掌握度不高，理想情況應是1.當對手在The flop選擇call而不是raise的時候，若手牌並非太糟，能夠嘗試稍微進攻逼迫對手蓋牌。2.當對手選擇raise不高的stack，但持續地進行raise的時候我沒辦法有效的偵測出來。3.因為是使用random跑多次去計算勝率，其勝率並不是理論上的勝率，也會和隨機生成的生成方法有關，因此在產生勝率表的時候可能也需要做一定程度的修正。

四、可改進之處

我認為在偵測對手牌型的危險度的時候應該要去觀察整個牌局的歷史，不能僅從現在的牌就去估算勝率，因為對手的叫牌可能就隱含了他的牌的好壞，也是一種資訊。只是在我的方法中並沒有利用到。